TOWARDS HUMAN-LIKE VISUAL REASONING: THEORY AND APPLICATIONS OF OBJECT-CENTRIC LEARNING

A thesis submitted for the degree of

DOCTOR OF PHILOSOPHY (PhD)

Department of Computing
Imperial College London

AVINASH KORI

Advised by
Prof. BEN GLOCKER
Prof. FRANCESCA TONI
Prof. FRANCESCO LOCATELLO

2025

DECLARATION

I Avinash Kori, hereby declare that the work presented in this thesis is my own and has not been submitted, either in the same or different form, to this or any other university for the award of a degree.

Any references to other authors and sources have been clearly indicated and acknowledged. I understand that any violation of this declaration constitutes academic misconduct.

COPYRIGHT STATEMENT

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution 4.0 International License (CC BY). Under this license, you may copy and redistribute the material in any medium or format for both commercial and non-commercial purposes. You may also create and distribute modified versions of the work. This on the condition that you credit the author.

When reusing or sharing this work, ensure you make the license terms clear to others by naming the license and linking to the license text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this license or permitted under UK Copyright Law.

Vidyāṃ cāvidyāṃ ca yas tad vedobhayaṃ saha |
Avidyayā mṛtyuṃ tīrtvā vidyayāmṛtam aśnute ||

"One who knows both knowledge (vidyā) and ignorance (avidyā) together — by ignorance he crosses over death, and by knowledge he attains immortality."

Verse 11, Isha Upanishad

ABSTRACT

Human perception is inherently object-centric, with *objectness*, the hierarchical structuring of environments from fundamental building blocks forming the foundation of higher level cognition and systematic generalization. Understanding how *parts* constitute objects, how objects form scenes, and how scenes create an environment is crucial for reasoning and learning. Object-centric representation learning aims to build AI systems that understand and reason about the world through stable, interpretable, and disentangled object representations. However, beyond capturing geometric properties, these representations must encode the *causal and functional roles* of objects within their environment.

This thesis focuses on the *identifiability* of object-centric representations an essential property for reliable learning and causal inference. Identifiability guarantees the model convergence to a particular local minima, irrespective of starting point and optimisation strategy. Without identifiability, models risk learning *spurious correlations*, being affected by *confounding factors, selection bias, and other artefacts*. To achieve identifiable object-centric representations, this thesis introduces three novel formalisms: 1. *CoSA (Unsupervised Conditional Slot Attention)*: Grounds learned representations to understandable physical entities. 2. *PSA (Probabilistic Slot Attention)*: Focuses on achieving identifiability in object centric representations. 3. *VISA (View-Invariant Slot Attention)*: Addresses challenges of partial visibility in object representations. Additionally, this thesis bridges object-centric learning with *causal reasoning*, providing a theoretical perspective that unifies causal inference and object-centric representations with broad implications for AI research.

Beyond these theoretical contributions, in this thesis we also explore practical applications of object-centric learning, particularly in *explainable AI*. We introduce *Free Argumentative eXchanges (FAX)* a novel methodology that frames model reasoning as *contestation between agents*. Initially, FAX is applied to a *discrete representation framework* before being extended to object-

centric representations. In summary, this thesis advances the field of *object-centric learning and explainability* by integrating insights from *disentangled representation learning* and contributing both *foundational theory* and *practical applications* for *human-aligned decision-making*.

This page is intentionally left blank.

ACKNOWLEDGEMENTS

Though this thesis stands as a culmination of academic work, many people have supported me along the way, and I want to thank these remarkable individuals from the bottom of my heart.

To Ben and Francesca—thank you for granting me the freedom to explore, and for placing your trust in me to chart my own course. Your steady mentorship, unwavering support, and constant encouragement kept me grounded throughout this journey. I am especially grateful for the ways in which you've helped me grow—not just as a researcher, but as a thinker.

To Francesco, thank you for your mentorship, for being open to discussions, and for always making time. I've learned a lot from working with you.

To my amazing mates, in alphabetical order—Aakash, Adam, Anish, Ainky, Ashwin, Ben, Charlie, Carles, Denis, Fabio, Fabrizio, Henrik, Matt, Mel, Mercy, Michael, Nairouz, Nandi, Rajat, Rhys, Sarah, Shahin, Tian, Vivek, Weitong, Xiang, and everyone else—it's hard to imagine this PhD experience without your company. You made it not only endurable but genuinely enjoyable.

To my start-up crew—Roshan, Nirav, Virenn, Manish, and Pavan—thank you for your incredible trust and support, allowing me to juggle this venture alongside my startup journey. Your belief in me meant the world.

I'm also truly grateful to the STAI CDT for funding this research, and to both Biomedical Image Analysis (BioMedIA) and Computational Argumentation (CLArgs) groups for providing such a vibrant and supportive research environment.

Finally, to my family—thank you for being my foundation. To my parents and my brother, your unconditional love and unwavering support have been my greatest strength. I owe everything to you.

This page is intentionally left blank.

CONTENTS

Abstract 4
Acknowledgements 7
1 Introduction 30
1.1 Challenges . . . 31
1.2 Thesis Objective and Contributions . . . 33
1.3 Thesis Structure and Bibliographic Notes . . . 34
2 Background 39
2.1 Human Perception . . . 39
2.1.1 Active Inference . . . 40
2.1.2 Free Energy Minimization . . . 40
2.2 Representation Learning . . . 42
2.2.1 Identifiable Representation Learning . . . 42
2.2.2 Causal Representation Learning . . . 43
2.2.3 Quantised Representation Learning. . . . 44
2.2.4 Unsupervised Object Discovery. . . . 45
2.2.5 Multiview nonlinear ICA. . . . 46
2.2.6 Multi-view Object-centric learning. . . . 46
2.2.7 Temporal Object-centric learning. . . . 47
2.2.8 Slot-attention. . . . 47
2.2.9 Compositionality . . . 48
2.3 Computational Argumentation Background . . . 49
2.3.1 (Quantified) Bi-polar Argumentation Frameworks . . . 49
2.3.2 Assumption-Based Argumentation (ABA) . . . 51

Part-1: Foundations of Object-Centric Learning

3 Grounding in Object-Centric Learning 56
3.1 Motivation . . . 56
3.2 Unsupervised Conditional Slot Attention: Formalism . . . 59
3.3 Experiments . . . 63
3.4 Conclusion & Discussion . . . 67
4 Identifiability in Object-Centric Learning 69

4.1 Motivation . . . 70
4.2 Probabilistic Slot Attention . . . 73
4.3 PSA: Theoretical Analysis . . . 81
4.4 PSA: Experiments . . . 85
4.5 View Invariant Slot Attention . . . 88
4.6 VISA: Theoretical Analysis . . . 93
4.7 VISA: Empirical Evaluation . . . 98
4.8 Conclusion & Discussion . . . 101

Part-2: Foundations of Human-like Visual Reasoning

5 Compositional Reasoning 105
5.1 Motivation . . . 105
5.2 Compositional OCL: A Unifying View . . . 110
5.2.1 Data Generating Process . . . 110
5.2.2 Causal Abstraction for Composition . . . 112
5.3 Putting Things in Context . . . 121
5.3.1 Relation to Spatial Mixtures . . . 121
5.3.2 Relation to Invariance Theory . . . 122
5.4 Why does it matter? . . . 123
5.5 Conclusion & Discussion . . . 125
6 Argumentative Visual Reasoning 128
6.1 Motivation . . . 128
6.2 FAX:Set-up . . . 131
6.3 FAX: Agents and Explanations . . . 131
6.4 FAX Explanations for Image Classification . . . 136
6.5 FAX: Implementation . . . 140
6.5.1 Class-specific discrete features . . . 140
6.5.2 Class-specific agents . . . 142
6.6 FAX: Evaluation Metrics . . . 144
6.7 FAX: Results . . . 145
6.8 Object-centric Argumentative Learning . . . 148
6.8.1 NAL: Architecture . . . 149
6.8.2 NAL: Symbolic Component . . . 150
6.8.3 NAL: Inference . . . 151
6.8.4 NAL: Experimental evaluation . . . 152

6.9 CoSA: End-to-End Visual Reasoning . . . 156
6.10 Conclusions & Future Work . . . 158

7 Conclusion 160

Part-3: Appendix

A CoSA-Appendix 188
A.1 Assumptions . . . 188
A.2 Proofs . . . 189
A.2.1 Proposition 1: *(Object Discovery - ELBO formulation)*: . . . 189
A.2.2 Proposition 2: *(ELBO formulation for reasoning task)*: . . . 190
A.3 Datasets . . . 191
A.3.1 CLEVR (Johnson et al., 2017) . . . 191
A.3.2 CLEVR-Hans3 (Stammer et al., 2021) . . . 192
A.3.3 CLEVR-Hans7 (Stammer et al., 2021) . . . 193
A.3.4 Tetrominoes (Kabra et al., 2019) . . . 193
A.3.5 Objects-Room (Kabra et al., 2019) . . . 193
A.3.6 Bitmoji (Mozafari, 2020) . . . 194
A.3.7 FFHQ (Karras et al., 2020) . . . 194
A.3.8 COCO (Lin et al., 2014) . . . 194
A.3.9 FloatingMNIST . . . 194
A.4 Metrics . . . 195
A.4.1 Mean squared error (MSE) . . . 196
A.4.2 Overlapping index (OPI) . . . 196
A.4.3 Codebook divergence (CBD) . . . 196
A.4.4 Codebook perplexity (CBP) . . . 197
A.4.5 Frechet inception distance (FID) . . . 197
A.4.6 Slot average Frechet inception distance (SFID) . . . 198
A.4.7 Hungarian Matching Coefficient (HMC) . . . 198
A.5 Algorithm & Forward Pass . . . 198
A.6 Quantization . . . 199
A.7 Codebook Collapse . . . 200
A.8 Object Discovery . . . 200

A.8.1 Quantitative Analysis . . . 200
A.8.2 Qualitative Analysis . . . 201
A.8.3 DINOSAUR Adaptation . . . 210
A.8.4 Codebook Size Analysis . . . 211
A.8.5 Multiple Object Instances . . . 214
A.9 Abstraction Function . . . 216
A.10 Object Composition . . . 216
A.10.1 Slot Dictionary Analysis . . . 217
A.10.2 Scene Generation . . . 217
A.11 Discriminative/Reasoning Tasks . . . 219
A.12 Training and Computational Details . . . 220
A.13 Limitations and Future work . . . 222
B PSA-Appendix 225
B.1 List of Assumptions . . . 225
B.2 Aggregate Posterior Proofs . . . 226
B.3 Injective Decoders: Sufficient Conditions . . . 232
B.4 Slot Identifiability Proof . . . 233
B.5 Metrics . . . 236
B.6 Automatic Relevance Determination of Slots . . . 238
B.7 Compositionality Results . . . 239
C VISA-Appendix 243
C.1 Notations . . . 243
C.2 Datasets . . . 245
C.2.1 Illustrative dataset . . . 245
C.2.2 Proposed dataset . . . 246
C.3 Mask Generation . . . 246
C.3.1 Additivity Implications . . . 247
C.4 Proofs . . . 247
C.5 Experiments . . . 264
C.5.1 Toy Setting . . . 264
C.5.2 Synthetic dataset results . . . 264
C.5.3 Influence of Number of Views . . . 265
C.5.4 MVMOVI Results . . . 265
C.5.5 View warm-up . . . 268
C.5.6 Hyperparameters . . . 269

c.5.7 Computational Resources 270

This page is intentionally left blank.

LIST OF FIGURES

Figure 2.1 Illustration of the Hollow-Face Illusion 41
Figure 3.1 Motivation for CoSA 57
Figure 3.2 Overview of CoSA 59
Figure 3.3 Binding illustration on BITMOJI dataset 64
Figure 3.4 CoSA object discovery results 66
Figure 3.5 CoSA object composition and discovery results . . . 66
Figure 4.1 Illustration of spatial ambiguities 71
Figure 4.2 Graphical models of probabilistic slot attention 74
Figure 4.3 Overview of Probabilistic slot attention and the identifiable aggregate slot posterior 74
Figure 4.4 Aggregate Gaussian Mixture Density 81
Figure 4.5 Aggregate posterior identifiability 84
Figure 4.6 Comparision of slot-identifiability scores 84
Figure 4.7 Concatenated Slot Gaussian Mixture 87
Figure 4.8 Overview of VISA 89
Figure 4.9 Graphical model illustration of VISA 89
Figure 4.10 Identifiability of $q(\mathbf{c})$ 97
Figure 4.11 Viewpoint invariance of $q(\mathbf{c})$ 99
Figure 5.1 Causal abstraction for composition. 106
Figure 5.2 Hierarchical progression for learning and generation. 107
Figure 5.3 Graphical model for causal abstraction. 111
Figure 5.4 Instantaneous causal and compositional graph. 112
Figure 5.5 Instantaneous compositional graph. 114
Figure 5.6 Jacobian based invariant causal mechanisms. 117
Figure 5.7 The figure demonstrates the generalisation of latent space as a result of optimising compositional consistency. 120
Figure 5.8 Issues with current foundation models. 123
Figure 6.1 Illustration of FAX 129

Figure 6.2 Overview of our implementation (argument dependencies are temporal). 140
Figure 6.3 FAX arguments on FFHQ and AFHQ 146
Figure 6.4 Neuro-Argumentative Framework. 148
Figure 6.5 ASP rules used to generate the 6 classes for the SHAPES dataset. We generated 3K images for each rule half of which was the negative instance of the rules. We then took 500 positive and negative splits for each rule as testing data. In total, we had 12K images for training and 6K for testing. 152
Figure 6.6 **(Left)** Standard machine learning evaluation metrics of attribute classification on datasets SHAPES and CLEVR **(Center)** Mean Adjusted Rand Index on datasets **(Right)** Average Precision on datasets. . . . 153
Figure 6.7 Rules in the ABA framework generated by our NAL architecture for SHAPES (class s1). Here, alpha_2 is an assumption, with $\overline{\texttt{a}alpha\_2(A,B)}$=c_alpha_2(A,B). . 155
Figure 6.8 Rules in the ABA framework generated by our NAL architecture for SHAPES (class s5). Here, alpha_2 is an assumption, with $\overline{\texttt{a}alpha\_2(A,B)}$=c_alpha_2(A,B). . 155
Figure 6.9 Rules in the ABA framework generated by our NAL architecture for CLEVR, differentiating class c3 from c1 and c2. Here, alpha_2 is an assumption, with $\overline{\texttt{a}alpha\_2(A,B)}$=c_alpha_2(A,B). 155
Figure 6.10 Standard evaluation metrics for the performance of NAL, ResNet, and NS-CL on the CLEVR-Hans3 task. 156
Figure 6.11 Confusion Matrix of NAL on the CLEVR-Hans3 task. 156
Figure A.1 Dataset illustrations 192
Figure A.2 Vanilla SA and CoSA-Cosine object discovery results on tetrominoes dataset. 202
Figure A.3 CoSA-Euclidian and CoSA-Gumbel object discovery results on tetrominoes dataset. 203
Figure A.4 Vanilla SA and CoSA-Cosine object discovery results on objects-room dataset. 204

Figure A.5 CoSA-Euclidian and CoSA-Gumbel object discovery results on objects-room dataset. 205
Figure A.6 Vanilla SA and CoSA-Cosine object discovery results on CLEVR dataset. 206
Figure A.7 CoSA-Euclidian and CoSA-Gumbel object discovery results on CLEVR dataset. 207
Figure A.8 Vanilla SA and CoSA-Cosine object discovery results on bitmoji dataset. 208
Figure A.9 CoSA-Euclidian and CoSA-Gumbel object discovery results on bitmoji dataset. 209
Figure A.10 Vanilla SA and CoSA-Cosine object discovery results on ffhq dataset. 209
Figure A.11 CoSA-Euclidian and CoSA-Gumbel object discovery results on ffhq dataset. 210
Figure A.12 Results on COCO dataset. 211
Figure A.13 Results on COCO dataset. 212
Figure A.14 Demonstration of object binding across instances in a given image. The title of each figure corresponds to the sampled element index of GSD $\mathfrak{S}$. 214
Figure A.15 Demonstration of object binding across frames in a video sequence. The title of each figure corresponds to the sampled element index of GSD $\mathfrak{S}$. 215
Figure A.16 Gumbel dictionary embedding to observation mapping for CLEVR dataset. 218
Figure A.17 Cosine dictionary embedding to observation mapping for Objects-room dataset. 218
Figure A.18 Gumbel dictionary embedding to observation mapping for FFHQ dataset. 218
Figure A.19 Image composition with random slot prompt on bitmoji dataset. 219
Figure A.20 Image composition with random slot prompt on CLEVR dataset. 219
Figure A.21 Image composition with random slot prompt on Tetrominoes dataset. 220
Figure A.22 Few shot task adaptability accuracy for all the methods. 221

Figure A.23 Few shot adaptability on FloatingMNIST2-diff to FloatingMNIST2-add dataset. 221
Figure A.24 Few shot adaptability on CLEVR-hans3 to CLEVR-hans7 dataset. 221
Figure B.1 Concatenated Slot Gaussian Mixture 227
Figure B.2 Identifiability scores throughout training 237
Figure B.3 Synthetic random samples 238
Figure B.4 Demonstration of ARD on CLEVR dataset. 239
Figure B.5 ARD illustration on ObjectsRoom 240
Figure B.6 Aggregate posterior sampling for image composition on CLEVR dataset. 241
Figure B.7 Aggregate posterior sampling for image composition on ObjectsRoom dataset. 241
Figure B.8 PSA results on MLP decoder 242
Figure B.9 PSA results on AR transformers 242
Figure C.1 Multi-view Data generating process 245
Figure C.2 Identifiability of $q(\mathbf{c})$ and $q(\mathbf{s})$ 264
Figure C.3 Viewpoint invariance for $q(\mathbf{c})$ 265
Figure C.4 Visual illustrations of benchmark results on CLEVR-aug dataset. 266
Figure C.5 Visual illustrations of benchmark results on CLEVR-mv dataset. 267
Figure C.6 Influence of Number of viewpoints on identifiability for synthetic datasets. 268
Figure C.7 Visual illustrations of benchmark results on mvMoVi-C dataset with 2 views. 268
Figure C.8 Visual illustrations of benchmark results on mvMoVi-C dataset with 3 views. 269

LIST OF TABLES

Table 3.1 Quantitative results if CoSA (F-ARI) 64
Table 3.2 CoSA compositionality benchmarks 66
Table 4.1 Assumption list of OCL methods 70
Table 4.2 Pascal VOC2012 benchmark results using probabilistic slot attention 86
Table 4.3 Comparing slot identifiability scores (SMCC and slot averaged R2) with existing object-centric learning methods. 87
Table 4.4 Comparing identifiability of $q(\mathbf{s})$, $q(\mathbf{c})$, and $p(\mathbf{v})$ scores wrt existing OCL methods. 98
Table 4.5 Identifiability and generalisability analysis on MVMoViC dataset. 99
Table 6.1 Accuracies of the trained classifiers 145
Table 6.2 Faithfulness properties of all considered methods on the DenseNet121 classifiers. 145
Table 6.3 Argumentative metrics for DenseNet-121 classifiers. 147
Table 6.4 All metrics for ResNet-18 classifiers. 147
Table 6.5 Standard evaluation metrics denoting how well NAL can distinguish between positive and negative instances of rules present in SHAPES images 155
Table 6.6 Reasoning benchmarks w.r.t accuracy and HMC. . . 158
Table A.1 Comparison between methods and their implicit assumptions and tasks achieved. 189
Table A.2 Maximum number of objects (including background) observed in a single image per dataset. 195
Table A.3 Sensitivity analysis of CoSA on object discovery. . . . 201

Table A.4 ARI on CLEVR6, Tetrominoes, ObjectsRoom, and COCO datasets, for SA baseline model and CoSA-cosine variant. In the case of COCO we use DINOSAUR variant of SA Seitzer et al., 2022 (ViT-S16) as a baseline and use DINO feature extractor (ViT-S16) for CoSA. . . . 201
Table A.5 Codebook properties across different sampling methods across datasets. . . . 210
Table A.6 Slot properties. . . . 210
Table A.7 Codebook size ablations with Cosine Sampling 213
Table A.8 Codebook size ablations with Euclidian Sampling . . . 213
Table A.9 Codebook size ablations with Gumbel Sampling . . . 213
Table A.10 Abstraction functions 216
Table A.11 Compositional FID and SFID scores. . . . 219
Table A.12 Quantitative evaluation of visual reasoning tasks on floatingMNIST2 dataset. . . . 222
Table A.13 Quantitative evaluation of visual reasoning tasks on CLEVR-Hans dataset. . . . 222
Table A.14 Quantitative evaluation of complex visual reasoning tasks on floatingMNIST2-mixed dataset. . . . 223
Table A.15 Quantitative evaluation of complex visual reasoning tasks on floatingMNIST3-mixed dataset. . . . 223
Table A.16 Quantitative evaluation of visual reasoning tasks on floatingMNIST3 dataset. . . . 224
Table A.17 Iteration per second details for all datasets on object discovery task. . . . 224
Table A.18 Iteration per second details for all datasets on reasoning task 224
Table B.1 Quality of compositional image generation with FID measure 238
Table C.1 Notation and definitions used in the model. 243
Table C.2 Comparing identifiability of $q(\mathbf{s})$, $q(\mathbf{c})$, and $p(\mathbf{v})$ scores wrt existing OCL methods on CLEVR-AUG dataset. . . . 265

Table C.3 Identifiability and generalisability analysis on MV-MoViD dataset. 270
Table C.4 Experimental details w.r.t datasets 271

This page is intentionally left blank.

LIST OF DEFINITIONS

2.1 Definition (Compositional Contrast) 48

3.1 Definition (Grounded Slot Dictionary (GSD)) 61
3.2 Definition (GSD Sampling) 61

4.1 Definition (Identifiability.) 82
4.2 Definition ($\sim_s$-equivalence) 82

5.1 Definition (Causal Graph: $\mathcal{G}$) 112
5.2 Definition (Compositional Graph: $\mathfrak{G}$) 113
5.3 Definition (Additive composition function) 115
5.4 Definition (Latent compositional models (LCM)) 115
5.5 Definition (Mechanism consistency) 117

6.1 Definition (Agent Explanandum) 131
6.2 Definition (FAX) 132
6.3 Definition (Lingua Franca) 134
6.4 Definition (Stance) 134
6.5 Definition (Resolution) 135
6.6 Definition (Strictly Interleaved FAX) 136
6.7 Definition (Image Classification BAF) 137
6.8 Definition (Monotonicity) 138
6.9 Definition (Reward) 143

B.1 Definition (Piece-wise Decoders, Kivva et al., 2022) 232
B.2 Definition (Slot Identifiability Brady et al., 2023) 233
B.3 Definition (Generic Point) 234

C.1 Definition (Additive models) 247

This page is intentionally left blank.

LIST OF THEOREMS

3.1 Proposition (ELBO for Object Discovery) 63

4.1 Lemma (Aggregate Posterior Mixture) 82
4.1 Theorem (Mixture Distribution of Concatenated Slots) 83
4.2 Theorem ($\sim_s$-Identifiable Slot Representations) 83
4.1 Corollary (Individual Slot Identifiability) 84
4.2 Lemma (Optimal Prior) 95
4.3 Theorem (Affine Equivalence) 95
4.4 Theorem (Invariance of aggregate content) 96
4.5 Theorem (Approximate representational equivariance) 96

5.1 Lemma (Acyclicity of composition) 114
5.1 Conjecture (Compositional consistency helps in generalisation) 119
5.2 Conjecture (Identifiability of LCM) 119
5.3 Conjecture ($\boldsymbol{J}_{g_z^{-1}} \sim_{DAG} (\boldsymbol{I} - \boldsymbol{C})$) 121

6.1 Corollary (Conflict) 138
6.1 Proposition (Dialectically Monotonicity) 139
6.2 Proposition (ELBO for Reasoning Tasks) 157

B.1 Corollary (Multiview Aggregate Posterior) 230
B.2 Corollary (Injectivity of mixing function) 232
B.3 Corollary (Inverse of mixing function) 233
B.1 Theorem (Generic Functions Kivva et al., 2022) 234
B.2 Theorem (Extension of Generic Point, Kivva et al., 2022) . . 234
B.3 Theorem (Equivalence of GMM, Kivva et al., 2022) 234

C.1 Lemma (ELBO) 247
C.2 Lemma (Mean GMM) 248
C.3 Lemma (Convex Combination of GMMs) 250
C.1 Theorem (Mixture of Concatenated Slots) 256

C.2 Theorem (Invertibility Kivva et al., 2022) 257

LIST OF ASSUMPTIONS

4.1 Assumption (View-point sufficiency) 93

5.1 Assumption (*Weak* Injectivity) 115

A.1 Assumption (Representation separation) 188
A.2 Assumption (Representation overcrowding) 188
A.3 Assumption (Object level sufficiency) 188
A.4 Assumption (Sufficient dictionary components) 188
A.5 Assumption (Injectivity Assumption) 188

B.1 Assumption ($\mathcal{B}$- Disentanglement, Lachapelle et al., 2024b) . 225
B.2 Assumption (Additive Mixing Function) 225
B.3 Assumption (Irreducibility) 225
B.4 Assumption (Compositionality) 225
B.5 Assumption ($\mathbf{u}$ task) 226
B.6 Assumption (Object Sufficiency) 226
B.7 Assumption (Decoder Injectivity) 226

This page is intentionally left blank.


HEY AI, WHAT'S THAT OBJECT?
Hey, ever wonder how humans just "know" what an object is?
Like this coffee mug?
We see a "mug," "book," or chair. But AI? it sees raw pixels.
So, it's not just shape, but also "what it means"?
To understand the world, AI needs to learn objectness – the idea that the world is mde up of separate, meaningful things.
Ah, like giving names and boundaries to stuff?
That's where Object-Centric Learning comesin. It teaches AI to spot an seprate. objects, just like. we do.
Sounds like teaching common sense. to a machine
With objectness, AI can gene-ralizc better – understand new sce-nes, adapt to change, even reasong
CHALLENGES
Occlusion
Grounding
Goometric
Abstract
This thesis tackles exactly th. at – how to make AI systems learn meaningful grounded object representat-tions that can think and argue!
Cool! So, you're teaching AI to'ee and "think" more like us?
Objectnees
That's the dream — one object at a time.
OBJECTNESS
OBJECTNESS
OBJECTNESS
So what's next? Emotions?
Let's not get ahead of oursselves...

1

INTRODUCTION

"The mind is everything. What you think you become."

– Rephrased, Gautama Buddha

Since the dawn of time, humans have wrestled with life's deepest questions—What is consciousness? Why are we here? What even is happiness? And let's be honest... we still have no clue. So instead, we turn to something slightly more manageable: aligning AI systems with how humans see the world. A critical part of this challenge is instilling the concept of *objectness* in AI systems, akin to how humans perceive the world (Lake et al., 2017; Schölkopf and Kügelgen, 2022). It has been suggested that humans subconsciously organize perceptual data into object entities, allowing them to better understand and interact with their environment (Rock, 1973; Hinton, 1979; Kulkarni et al., 2015; Behrens et al., 2018). Objectness is a multifaceted concept that can be understood in physical, abstract, semantic, geometric, or boundary-based terms (Yuille and Kersten, 2006; Epstein et al., 2017). The primary goal of object-centric representation learning (OCL) is to develop systems that maintain a stable, *grounded*, and environment-invariant notion of objectness, so that the learned representations remain disentangled and meaningful (Bengio et al., 2013a).

Objectness involves understanding an environment in terms of its building blocks, which can be broken down into the composition of objects and their parts (Hinton, 1979; Hinton, 2022). This capability of decomposition has been hypothesised to be one of the reason why humans can generalize with few examples (Tenenbaum et al., 2011). Recent progress in OCL has shown promise in segmenting objects in observed scenes (Locatello et al., 2020b; Löwe et al., 2024). OCL methods aim to enable agents to learn object representations within their environments, as opposed to relying on global representations as used in traditional generative models like variational autoencoders (Kingma and Welling, 2013). OCL approaches facilitate spa-

tially disentangled representations, which are essential for compositional scene generation (Bengio et al., 2013a; Lake et al., 2017; Battaglia et al., 2018; Greff et al., 2020) and for understanding causal interactions among objects (Marcus, 2003; Gopnik et al., 2004; Gerstenberg et al., 2021).

1.1 CHALLENGES

The core challenge in achieving human-like situational awareness in AI systems lies in formalizing object representations in a way that allows the systems to not only identify and segment physical objects but also to understand their abstract and semantic roles. These representations must extend beyond mere geometric properties—they must also encompass the causal and functional roles objects play within their environments. Learning such representations is deeply connected to the concept of identifiability, which refers to the models capability to converge to a particular local minima for a considered objective irrespective of model initialisation and optimisation strategy. Identifiability is essential for both reliable representation learning and causal inference. Additionally, with extra assumptions on data/environment, identifiability enables inferring unseen aspects of the environment, while maintaining grounded and accurate object representations in the case of incomplete information setting. In summary, several challenges must be addressed to fully realize object-centric reasoning in AI systems:

1. **The binding problem**: This issue involves how individual objects are combined into an unified context (Revonsuo and Newman, 1999). In human cognition, this involves linking sensory inputs, such as visual and auditory stimuli, to the correct mental representations of objects. In any AI system, this challenge is significant because it involves creating systems that can correctly identify and link abstract features (like object parts or properties) to their corresponding entities in the environment.

2. **The grounding problem**: This challenge refers to the process of linking abstract object representations to their real-world counterparts, encompassing their function and meaning (Harnad, 1990). Grounding

can be seen as the challenge of learning abstract, *canonical* representations of objects[1].

3. **Identifiability of representations**: This refers to the model's ability to converge to the same solution regardless of the optimization process or initial starting conditions in the limit of infinite data. Identifiability is crucial for ensuring that the model's predictions are not the result of spurious correlations, bias, or confounding factors (Hyvärinen et al., 2023).

4. **Ambiguities and partial visibility**: In dynamic environments, where occlusion, partial visibility, and view ambiguities due to rapid changes are common, AI systems must be able to reason with incomplete data and infer missing information while maintaining grounded object representations, particularly in safety-critical contexts like medical imaging and autonomous driving (Biederman, 1987). Handling view ambiguities involves learning a common representation in case of multiple views, addressing the transformation of object representations due to unobserved views (Zhang et al., 2018).

5. **Temporal reasoning**: Humans naturally engage in temporal reasoning, using past experiences to understand how situations evolve and make decisions accordingly. Currently, many AI systems struggle with forecasting long-term outcomes and adjusting plans in response to changing circumstances. Developing AI that can reason over time, recognize long-term dependencies, and make decisions based on future states is crucial for achieving human-like intelligence (Friston et al., 2017).

6. **Object-part composition**: In human perception, we naturally segment objects from their surroundings and understand how smaller parts form a whole. This segregation process is crucial for reasoning and decision-making. For AI to reason effectively, it must be able to segment complex scenes into distinct objects and understand how those objects are composed of parts. Recent advancements in object-centric

1 Neural network-based representations are grounded in input data, unlike classical *symbols*, whose meanings are often subject to human interpretation.

representation learning aim to address this, but AI still struggles with setting up a hierarchy of parts and objects (Hinton, 2022).

7. **Cross-modal reasoning**: Humans integrate information from multiple sensory modalities—vision, hearing, touch, etc.—to form a unified understanding of the environment. In contrast, AI systems often operate in isolated modalities, leading to challenges in multimodal reasoning. For human-like reasoning, AI systems must be capable of combining data from multiple sources, such as visual, auditory, and textual information, to form coherent representations and make decisions.

1.2 THESIS OBJECTIVE AND CONTRIBUTIONS

The primary goal of this thesis is to develop methods that directly address several of the challenges mentioned above and demonstrate the practical application of these methods to real-world data. The research objectives tackled in this thesis are as follows:

- **Objective 1 - The notion of binding and grounding:** A central challenge in achieving human-like reasoning or in OCL is to disentangle the object representations and association of these estimated objects or building blocks of a scene to physical entities.
- **Objective 2 - Identifiability of object representations:** A significant challenge in developing generalizable models is ensuring the existence of local minima and demonstrating that the model will consistently converge to the same local minima under optimal training.
- **Objective 3 - Identifiability under spatial ambiguities:** Similar to the previous objective, another key focus is demonstrating the model's convergence to local minima in the presence of spatial ambiguities, such as partial visibility.
- **Objective 4 - Causal reasoning perspective:** By linking the previous objectives through a causal reasoning lens, we explore a unifying view of scene graph modelling by considering OCL from a causal representation learning (CRL) perspective.

- **Objective 5 - Argumentative reasoning perspective:** A natural form of human reasoning involves argumentation, whether in internal monologues or conversations with others. This thesis replicates this behaviour with AI agents, enabling agent-agent argumentation before reaching a desired conclusion.

This thesis primarily addresses **objective 1** by learning a dictionary of canonical object representations, which are used as building blocks for composition and reasoning tasks. From the perspective of **objective 2**, we developed a probabilistic interpretation of the OCL method, thereby providing identifiability guarantees up to a defined equivalence. To tackle **objective 3**, we adopt a multi-view approach, considering the scene from multiple viewpoints to learn invariant and unique object representations, while providing identifiability guarantees. For **objective 4**, we approach the problem from a world modeling perspective, unifying CRL with OCL, and demonstrate how these methods are related and can be explored to enable causal reasoning in AI systems. Finally, for **objective 5**, we build upon various methods in computational argumentation (Atkinson et al., 2017), extending their applications to the imaging domain, and illustrate their applicability in the context of end-to-end training for classification tasks.

1.3 Thesis structure and bibliographic notes

The thesis is structured to discuss the contributions and impact on OCL first followed by its applications in achieving human like reasoning. The thesis is split into two parts: (i) Foundations of object-centric learning, and (ii) Foundations of human-like reasoning. In total the thesis consists of seven chapters, building on multiple publications that are listed bellow:

- Chapter 3 builds on:
    1. Kori, Avinash and Locatello, Francesco and Ribeiro, Fabio De Sousa and Toni, Francesca and Glocker, Ben. Grounded object centric learning. In Proceedings of The Twelfth International Conference on Learning Representations (ICLR 2024) (Kori et al., 2024b).

- Chapter 4 builds on:
    1. Kori, Avinash and Locatello, Francesco and Toni, Francesca and Glocker, Ben and Ribeiro, Fabio De Sousa. Identifiable object centric representations via probabilistic slot attention. In Proceedings of the Part of Advances in Neural Information Processing Systems 37 (NeurIPS 2024) (Kori et al., 2024c)
    2. Kori, Avinash and Toni, Francesca and Ben Glocker. Identifiable Object Representations under Spatial Ambiguities. In Proceedings of the Part of International Conference of Machine Learning 42 (ICML 2025) (Kori et al., 2025d).
- Chapter 5 builds on:
    1. Kori, Avinash and and Toni, Francesca and Glocker, Ben and Schölkopf, Bernhard and Locatello, Francesco. Unifying Causal and Object-centric Representation Learning allows Causal Composition. Published as a part of ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy (In review NeuIPS 2025) (Kori et al., 2025b).
- Chapter 6 builds on:
    1. Kori, Avinash and Rago, Antonio and Toni, Francesca. Free Argumentative Exchanges for Explaining Image Classifiers. In Proceedings of The 24th International Conference on Autonomous Agents and Multi-Agent Systems (AMAAS 2025) (Kori et al., 2025c).
    2. Jacob, Abdul Rahman and Kori, Avinash and Angelis, Emanuele De and Glocker, Ben and Proietti, Maurizio and Toni, Francesca. Object-Centric Neuro-Argumentative Learning. In Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning (NeSy 2025) (Jacob et al., 2023).
    3. Kori, Avinash and Locatello, Francesco and Ribeiro, Fabio De Sousa and Toni, Francesca and Glocker, Ben. Grounded object centric learning. In Proceedings of The Twelfth International Conference on Learning Representations (ICLR 2024) (Kori et al., 2024b).

In addition, I directly or indirectly contributed during my Ph.D to the following publications (relevant to this thesis but not part of it):

1. Kori, Avinash and Toni, Francesca and Glocker, Ben. GLANCE: Global to Local Architecture-Neutral Concept-based Explanations.Published as a part of XAI workshop, at Advances in Neural Information Processing Systems 36 (NeurIPS 2023) (Kori et al., 2023a)

2. Kori, Avinash and Glocker, Ben and Toni, Francesca. Explaining Image Classifiers with Visual Debates. In Proceedings of 27th International Conference of Discovery Science (DS 2024) (Kori et al., 2024a).

3. Kori, Avinash and Balsells-Rodas, Carles and Glocker, Ben and Li, Yingzhen and Locatello, Francesco. Causal Representation Learning and Inference via Mixture-Based Priors. Published as a part of ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy (Delta ICLR25) (Kori et al., 2025a).

4. Kori, Avinash and Sanchez, Pedro and Vilouras, Konstantinos and Glocker, Ben and Tsaftaris, Sotirios. A Causal Ordering Prior for Unsupervised Representation Learning. Published as a part of CRL workshop, at Advances in Neural Information Processing Systems 36 (NeurIPS 2023) (Kori et al., 2023b).

5. Santhirasekaram, Ainkaran and Kori, Avinash and Winkler, Matthew and Rockall, Andrea and Glocker, Ben. Vector Quantisation for Robust Segmentation. In proceedings of 25th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2022) (Santhirasekaram et al., 2022).

6. Santhirasekaram, Ainkaran and Kori, Avinash and Winkler, Matthew and Rockall, Andrea and Glocker, Ben. Robust Hierarchical Symbolic Explanations in Hyperbolic Space for Image Classification. Published as a part of topology workshop at IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPRw 2023) (Santhirasekaram et al., 2023).

7. Rasal, Rajat and Kori, Avinash and Glocker, Ben. Causal Representation Learning with Observational Grouping for CXR Classification. In

review at 28th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2025).

8. Rasal, Rajat and Kori, Avinash and Ribeiro, Fabio De Sousa and Xia, Tian and Glocker, Ben. Diffusion Counterfactual Generation with Semantic Abduction. In Proceedings of the Part of International Conference of Machine Learning 42 (ICML 2025).

9. Dhir, Anish and Sedgwick, Ruby and Kori, Avinash and Glocker, Ben, and Wilk, Mark van der. Continuous Bayesian Model Selection for Multivariate Causal Discovery. In Proceedings of the Part of International Conference of Machine Learning 42 (ICML 2025) (Dhir et al., 2024).

10. Thomas, Devin and Jiang, Junqi and Kori, Aviansh and Russo, Aaron, Winkler, Steffen, Sale, Stuart, McMillan, Joseph, Belardinelli, Francesco, and Rago, Antonio. Explainable Reinforcement Learning for Formula One Race Strategy. In proceedings at 40th ACM/SIGAPP Symposium On Applied Computing (SAC 2025) (Thomas et al., 2025).

11. Ribeiro, Fabio De Sousa and Todd, Omar and Jones, Charles and Kori, Avinash and Mehta, Raghav and Glocker, Ben. Flow Stochastic Segmentation Networks. In review at International Conference on Computer Vision (ICCV 2025).

This page is intentionally left blank.

2

BACKGROUND

"To know what you know and what you do not know, that is true knowledge."

– Confucius

2.1 HUMAN PERCEPTION

Human perception is an active process where the brain interprets sensory input to construct a stable experience of the world. Central to this process is the philosophy of objectness, which explores how we perceive distinct objects despite variations in viewpoint, illumination, or occlusion Biederman, 1987. Traditional theories, such as structuralism, proposed that perception results from assembling fundamental sensory features, while Gestalt emphasized that perception arises from holistic patterns (Wertheimer, 1923). More recent computational models suggest that perception is fundamentally inferential, meaning the brain forms hypotheses about the environment and updates them based on sensory feedback (Gregory, 1980). This shift towards predictive processing aligns with the Bayesian brain hypothesis, which states that perception results from a probabilistic model that combines prior beliefs with sensory evidence (Knill and Pouget, 2004).

Pragmatic view on objectness

*Objectness is a multifaceted property that can be characterised as physical, abstract, semantic, geometric, or via spaces and boundaries (Yuille and Kersten, 2006; Epstein et al., 2017). Biederman (1987) proposed the notion of "*Geons*", which basically considers canonical list of 3D geometric icons like:* sphere, cylinders, cube, etc. *to represent any complex object composition or scenes involved. However modelling an environment as a group of geons is quite challenging. In this work, we consider a pragmatic view on objectness with a workable definition of 'objects' as in Kosiorek (2019), which basically defines an object in an image as a set of pixels characterised by certain properties and that objects in a single image or video coexist within a shared context, or remain temporally consistent in the case of videos.*

2.1.1 *Active Inference*

The inferential perspective is central to *Active Inference*, a theory suggesting that perception is not a passive recording of sensory stimuli but an active process that minimizes uncertainty. The brain conforms to reality with its preconceived prior to a larger extent, minimising uncertainty or free energy (refer Figure 2.1). Instead of merely reacting to sensory inputs, the brain predicts the sensory signal and adjusts its internal model based on mismatches (prediction errors). The predictive coding model suggests that the brain is organized hierarchically, where higher-level predictions guide lower-level sensory processing, and mismatches (errors) propagate upward to update models (Clark, 2013).

Active inference reframes perception as a control problem, where the brain selects actions that optimize its internal states to match external reality. Instead of passively receiving information, organisms actively seek sensory inputs that align with their internal models. For instance, eye movements, head tilts, and exploratory hand movements are all mechanisms through which the brain updates its perceptual models (Parr and Friston, 2018).

2.1.2 *Free Energy Minimization*

At a deeper level, this process is governed by *Free Energy Minimization*, a principle proposed by Friston, 2010, which states that biological systems strive to reduce the discrepancy between expected and actual sensory inputs. Mathematically, *free energy* is an upper bound on surprise (or uncertainty), meaning that perception and action work together to reduce uncertainty about the world. The idea that perception is an active, inference-driven process rather than a passive one has significant implications for neuroscience and AI. The free energy principle provides a unifying framework that ties together perception, action, and cognition, explaining how biological systems self-organize and learn efficiently. Notably, this principle aligns closely with modern approaches in *unsupervised learning* methods, including variational autoencoders (VAE)(Kingma and Welling, 2013).



Figure 2.1: Illustration of the Hollow-Face Illusion: (1) A fully convex face, perceived as normal, (2) A side-view of the mask, showing depth, (3) A fully concave mask, which may still be perceived as convex due to strong top-down priors in human perception (the image is generated with gpt-4o).

Free-energy – ELBO

Let $\mathbf{x}, \mathbf{z}$ *be the sensory input and the hidden causes (latent variables), let* $p(\mathbf{x}, \mathbf{z})$ *be the joint generative model. The posterior belief over hidden causes is defined as:*

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}}, \tag{2.1}$$

with brain perception model, which can be considered as a posterior $q(\mathbf{z})$, $\mathcal{F} = D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))$, *where* $\mathcal{F}$ *is the free energy, acting as a bound on surprise. The brain minimizes this bound by adjusting* $q(\mathbf{z})$ *to match the true posterior. The joint objective which the brain minimises is:* $\mathcal{F} = \textit{Prediction Error} + D_{KL}(q(\mathbf{z})||p(\mathbf{z}))$. *This explains illusory perceptions, such as the hollow mask illustration, where prior expectations override sensory evidence. Free energy minimisation suggests that conscious awareness arises when high-level generative models align with sensory input. When a perceptual conflict exists, the brain settles it based on a stable interpretation.*

$$\mathcal{F} = \textit{(Perceptual Stability)} - D_{KL}(\textit{Expectation}||\textit{Sensory Evidence}). \tag{2.2}$$

Early advances in unsupervised representation learning were built upon the VAE framework (Kingma and Welling, 2013). Approaches like β-VAE (Higgins et al., 2017) and its extensions (Eastwood and Williams, 2018; Kim and Mnih, 2018a; Mathieu et al., 2019) aim to learn disentangled representations by promoting independence between latent variables (Higgins et al., 2022; Liu et al., 2022). While these methods have achieved some success, it has been proven that learning disentangled representations from i.i.d. data in a completely unsupervised manner is theoretically impossible (Hyvärinen and Pajunen, 1999; Locatello et al., 2019). Recent approaches have shown that restricting the class of decoder functions to conformal maps (Buchholz et al., 2022) or volume-preserving transformations (Yang et al., 2022) can lead to identifiable models.

2.2.1 *Identifiable Representation Learning*

The concept of identifiability in representation learning originates from early work on Independent Component Analysis (ICA) (Hyvärinen and Pajunen, 1999; Hyvärinen and Oja, 2000), and has recently seen a resurgence (Hyvarinen and Morioka, 2016; Hyvarinen et al., 2019b; Locatello et al., 2019; Khemakhem et al., 2020a; Von Kügelgen et al., 2021; Lachapelle et al., 2024a; Yao et al., 2024b). To address the identifiability challenge, several strategies have been developed: (i) restricting the class of mixing functions (decoding functions that maps latent features with observations); (ii) leveraging non-i.i.d., interventional, or counterfactual data; and (iii) imposing structure on the latent space through distributional assumptions. For (i), restricting mixing functions to conformal maps (Buchholz et al., 2022) or volume-preserving transformations (Yang et al., 2022) has been found to lead to identifiable models. For (ii), previous works (Locatello et al., 2020a; Von Kügelgen et al., 2021; Zimmermann et al., 2021; Ahuja et al., 2022; Brehmer et al., 2022) utilize contrastive pairs of observations $(\mathbf{x}, \tilde{\mathbf{x}})$, obtained through data augmentation, interventions, or approximate counterfactual inference. Regarding (iii), latent space structure can be enforced either by introducing auxiliary variables to make the latent variables conditionally independent

(Hyvarinen et al., 2019a; Khemakhem et al., 2020a; Khemakhem et al., 2020b) or by applying distributional assumptions, such as a mixture prior in a VAE (Dilokthanakul et al., 2016; Willetts and Paige, 2021; Kivva et al., 2022).

A model is considered identifiable when different training iterations yield consistent latent distributions, thereby resulting in identical model parameters (Khemakhem et al., 2020a; Khemakhem et al., 2020c). In the context of a parameter space Θ and a family of mixing functions $\mathcal{F}$, identifiability of the model on the dataset $\mathcal{X}$ is established if, for any $\theta_1, \theta_2 \sim \Theta$ and $f_{\theta_1}, f_{\theta_2} \sim \mathcal{F}$, the condition $p(f_{\theta_1}^{-1}(\mathbf{x})) = p(f_{\theta_2}^{-1}(\mathbf{x}))$ holds for all $\mathbf{x} \in \mathcal{X}$, implying $\theta_1 = \theta_2$. However, in practical scenarios, exact equality or "strong" identifiability is often unnecessary, as establishing relationships to transformations, which can be manually recovered (transformations like affine, polynomial maps), proves equally effective, resulting in a notion of weak identifiability, where relationships are recovered up to an affine transformation (Khemakhem et al., 2020c; Kivva et al., 2022).

2.2.2 *Causal Representation Learning*

Modelling causal relationships becomes feasible when interventional or non-i.i.d. data is available. Ahuja et al. (2022) employ an injective polynomial decoder and train their model using both observational and interventional data. Varici et al. (2023) explore the case of an injective linear decoder, optimizing the score function of the distribution in both the latent and observation spaces. In Locatello et al. (2020a), observations are captured before and after unknown interventions (i.e., counterfactual data), while Brehmer et al. (2022) extend this concept to more complex causal graphs. For non-i.i.d. data, Lippe et al. (2022) extract causal factors from spatio-temporal data through interventions over time. Some methods assume partial supervision, such as access to ground-truth causal factors. Other works, like Yang et al. (2021), directly model exogenous noise, which is mapped to causal latent variables using a linear SCM. Recently, (Yao et al., 2024a) unified these settings into a single framework, preserving known invariances in the data. These invariances can stem from multi-view, temporal, or counterfactual data. One missing component is structural assumptions on the interplay between latent distribution and generating functions.

2.2.3 *Quantised Representation Learning.*

Van Den Oord, Vinyals, et al., 2017 propose Vector Quantized Variational Autoencoders (VQ-VAE) to learn discrete latent representations by mapping a continuous latent space to a fixed number of *codebook embeddings*. The codebook embeddings are learned by minimizing the mean squared error between continuous and nearest codebook embeddings. The learning process can also be improved upon using the Gumbel softmax trick Jang et al., 2016; Maddison et al., 2016. The VQ-VAE has been further explored for text-to-image generation (Ding et al., 2021; Esser et al., 2021; Gu et al., 2021; Ramesh et al., 2021). One major challenge in the quantization approach is effectively utilizing codebook vectors. Yu et al., 2021; Santhirasekaram et al., 2022; Santhirasekaram et al., 2023 address this by projecting the codebook vectors onto Hyperspherical and Poincare manifold. Träuble et al., 2022 propose a discrete key-value bottleneck layer and show the effect of discretization on non-i.i.d samples, generalizability, and robustness tasks.

Let $\mathbf{z} = \Phi_e(\mathbf{x}) \in \mathbb{R}^{N \times d_z}$ denote an encoded representation of an input datapoint $\mathbf{x}$ consisting of N embedding vectors in $\mathbb{R}^{d_z}$. Discrete representation learning aims to map each $\mathbf{z}$ to a set of elements in a codebook $\mathfrak{S}$ consisting of M embedding vectors. The codebook vectors are randomly initialized at the start of training and are updated to align with $\mathbf{z}$. The codebook is initialised in a specific range based on the choice of sampling as detailed below:

1. **Euclidean:** The embeddings are randomly initialized in $(-1/M, 1/M)$ as described by Van Den Oord, Vinyals, et al., 2017. The discrete embeddings $\tilde{\mathbf{z}}$ are then sampled with respect to the Euclidean distance as follows: $\tilde{\mathbf{z}} = \arg\min_{\mathfrak{S}_j} ||\mathbf{z} - \mathfrak{S}_j||_2^2, \ \forall \ \mathfrak{S}_j \in \mathfrak{S}$.

2. **Cosine:** The embeddings are initialized on a unit hypersphere (Yu et al., 2021; Santhirasekaram et al., 2022). The representations $\mathbf{z}$ are first normalized to have unit norm: $\hat{\mathbf{z}} = \mathbf{z}/||\mathbf{z}||$. The discrete embeddings are sampled following: $\tilde{\mathbf{z}} = \arg\max_{\mathfrak{S}_j} \langle \hat{\mathbf{z}}, \mathfrak{S}_j \rangle, \ \forall \ \mathfrak{S}_j \in \mathfrak{S}$, where $\langle ., . \rangle$ denotes vector inner product. The resulting discrete representations are then upscaled as $\tilde{\mathbf{z}} \cdot ||\mathbf{z}||$.

3. **Gumbel:** The embeddings are initialised similarly to the Euclidian codebook. The representations $\mathbf{z}$ are projected onto discrete embeddings in $\mathfrak{S}$ by measuring the pair-wise similarity between the $\mathbf{z}$ and the M codebook elements. The projected vector $\hat{\mathbf{z}}$ is used in the Gumbel-Softmax trick (Jang et al., 2016; Maddison et al., 2016) resulting in the continuous approximation of one-hot representation given by $\text{softmax}((\mathbf{g}_i + \hat{\mathbf{z}}_i)/\tau)$, where g_i is sampled from a Gumbel distribution, and τ is a temperature parameter.

2.2.4 *Unsupervised Object Discovery.*

Several notable works in the field object discovery, including Burgess et al., 2019; Engelcke et al., 2019; Greff et al., 2019, employ an iterative variational inference approach (Marino et al., 2018), whereas Lin et al., 2020; Van Steenkiste et al., 2020 adopt more of a generative perspective. More recently, the use of iterative attention mechanisms in object *slot*-based models has garnered significant interest (Locatello et al., 2020b; Engelcke et al., 2021; Singh et al., 2021; Emami et al., 2022; Singh et al., 2022a; Wang et al., 2023a). The focus of most of these approaches remains centered around the disentanglement of slot representations, which can be understood as tackling the *segregation* and *representation* aspects of the binding problem (Greff et al., 2020). Lin et al., 2020; Van Steenkiste et al., 2020; Singh et al., 2021 focus primarily on tackling the *composition* aspect of the binding problem, either in a controlled setting or with a predefined set of prompts. However, unlike our approach, they do not specifically learn *grounded* symbol-like entities for scene composition. Another line of research tackling the binding problem involves capsules (Sabour et al., 2017; Hinton et al., 2018; Ribeiro et al., 2020). However, these methods face scalability issues and are typically used for discriminative learning. Kipf et al., 2021; Elsayed et al., 2022 perform conditional slot attention with weak supervision, such as using the center of mass of objects in the scene or object bounding boxes from the first frame.

2.2.5 *Multiview nonlinear ICA.*

It has been noted that addressing the challenge of nonlinear ICA can involve incorporating a learnable clustering task within the latent representations, thereby imposing asymmetry in the latent distribution (Willetts and Paige, 2021; Kivva et al., 2022). Moreover, Gresele et al. (2020) delve into multiview nonlinear ICA, particularly in scenarios involving corrupted observations, where they aim to recover invariant representations while accounting for certain ambiguities. Along similar lines, Von Kügelgen et al. (2021) and Daunhawer et al. (2023) explore the concept of style-content identification using contrastive learning, focusing on addressing the multiview nonlinear ICA problem. Here, we work along similar lines by emphasising the learning of invariant content and identifiable object-centric representations. We achieve this by formulating a reconstruction objective where the enforced invariance and equivariance stem from the underlying probabilistic graphical model rather than relying on a contrastive learning objective.

2.2.6 *Multi-view Object-centric learning.*

Recent progress in multi-view object-centric learning has seen notable contributions from methods like MulMON (Li et al., 2020), ROOTS (Chen et al., 2021), SlotLifter(Liu et al., 2025), and uOCF(Luo et al., 2024), each offering distinct approaches to compositional representation learning. However, these methods rely heavily on viewpoint annotations, which limit their applicability in fully unsupervised settings. MulMON refines object representations iteratively using annotated viewpoint-image pairs, while ROOTS, SlotLifter, uOCF estimates 3D object positions performing an inverse rendering operation within a grid and projects them into image space via viewpoint transformations. In contrast, we deal with fully unsupervised framework without the need of viewpoint annotations while providing approximate viewpoint equivariance for object representations.

2.2.7 *Temporal Object-centric learning.*

An alternative approach to bypass the need for viewpoint annotations leverages temporal information. Methods for learning from single-viewpoint video sequences, such as Relational N-EM (Van Steenkiste et al., 2018), SQAIR (Kosiorek et al., 2018), SILOT (Crawford and Pineau, 2020), and SAVi (Kipf et al., 2021), focus on modeling object motion, interactions, and identity tracking across frames, even under occlusion. However, these methods assume fixed viewpoints, making them unsuitable for multi-view scenarios where objects appear in different spatial configurations. Additionally, object motion affects individual objects independently, unlike viewpoint changes, which influence the entire scene. Recent advances such as DyMON (Li et al., 2021) extend multi-view approaches like MulMON (Li et al., 2020) to dynamic scenes by disentangling object motion and viewpoint changes, assuming one dominates in adjacent frames. However, DyMON relies on viewpoint annotations, limiting its utility in unsupervised settings. Temporal methods such as SIMONe (Luo et al., 2024) address this by leveraging temporal coherence across multi-view videos, using spatial and temporal positional embeddings to disentangle object and viewpoint representations. Yet, SIMONe's reliance on temporal continuity restricts its generalizability to scenarios where such coherence is absent.

2.2.8 *Slot-attention.*

The Slot attention (Locatello et al., 2020b) module is an architectural mechanism designed to interface with perceptual representations—such as those produced by a convolutional neural network—and generate a set of task-dependent abstract representations, referred to as slots. These slots bind to arbitrary objects in the input through a competitive attention-based refinement process that unfolds over multiple iterative steps. Slot attention takes a set of feature embeddings $\mathbf{z} \in \mathbb{R}^{N \times d_z}$ as input, and applies an iterative attention to produce K object representations called slots $\mathbf{s} \in \mathbb{R}^{K \times d_s}$. Let $\mathcal{Q}_\gamma, \mathcal{K}_\beta$ and $\mathcal{V}_\phi$ denote query, key and value projection networks with parameters β, γ and ψ respectively acting on $\mathbf{z}$. To simplify our exposition

later on, let f and g be shorthand notation for the *slot update* and *attention* functions respectively:

$$f(\boldsymbol{A}, \mathbf{v}) = \boldsymbol{A}^T \mathbf{v}, \quad A_{ij} = \frac{g(\mathbf{q}, \mathbf{k})_{ij}}{\sum_{l=1}^{K} g(\mathbf{q}, \mathbf{k})_{lj}} \quad \text{and}$$

$$g(\mathbf{q}, \mathbf{k}) = \frac{e^{M_{ij}}}{\sum_{l=1}^{N} e^{M_{il}}}, \quad \boldsymbol{M} = \frac{\mathbf{k}\mathbf{q}^T}{\sqrt{d_s}}, \tag{2.3}$$

where $\mathbf{q} = \mathcal{Q}_\gamma(\mathbf{z}) \in \mathbb{R}^{K \times d_s}$, $\mathbf{k} = \mathcal{K}_\beta(\mathbf{z}) \in \mathbb{R}^{N \times d_s}$, and $\mathbf{v} = \mathcal{V}_\phi(\mathbf{z}) \in \mathbb{R}^{N \times d_s}$ are the query, key and value vectors respectively. The attention matrix is denoted by $\boldsymbol{A} \in \mathbb{R}^{N \times K}$. Unlike self-attention (Vaswani et al., 2017), the queries in slot attention are a function of the slots $\mathbf{s} \sim \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}, \boldsymbol{\sigma}) \in \mathbb{R}^{K \times d_s}$, and are iteratively refined over T attention iterations (see refinement module in Fig. 3.2). The slots are randomly initialized at $t = 0$. The queries at iteration t are given by $\hat{\mathbf{q}}^t = \mathcal{Q}_\gamma(\mathbf{s}^t)$, and the slot update process can be summarized as: $\mathbf{s}^{t+1} = f(g(\hat{\mathbf{q}}^t, \mathbf{k}), \mathbf{v})$. Lastly, a Gated Recurrent Unit (GRU), which we denote by $\mathcal{H}_\theta$, is applied to the slot representations $\mathbf{s}^{t+1}$ at the end of each SA iteration, followed by a generic MLP skip connection.

2.2.9 *Compositionality*

Object compositionality as defined by Brady et al., 2023 is a structure imposed on the slot decoder mapping f_d which implies that each image pixel is a function of at most one slot representation, thereby enforcing a local sparsity structure on the Jacobian matrix of f_d.

Definition 2.1 (Compositional Contrast). *For a differentiable mapping $f_d : \mathcal{Z} \to \mathcal{X}$, the compositional contrast of f_d at $\mathbf{z}$ is given by:*

$$C_{comp}(f_d, \mathbf{z}) = \sum_{n=0}^{N} \sum_{k=1}^{K} \sum_{j=k+1}^{K} \left\| \frac{\partial f_d(\mathbf{z})_n}{\partial \mathbf{z}_k} \right\| \left\| \frac{\partial f_d(\mathbf{z})_n}{\partial \mathbf{z}_j} \right\|.$$

Brady et al. (2023)'s main result (Theorem 1) relies on *compositionality* and *invertibility* of f_d to guarantee slot-identifiability when both the compositional contrast and the reconstruction loss equal zero. However, using $C_{\text{comp}}(f_d, \mathbf{z})$ as a metric or as part of an objective function is computationally prohibitive[1].

2.3 COMPUTATIONAL ARGUMENTATION BACKGROUND

Given in this thesis we also focus on reasoning in addition to representation learning, now we briefly provide a context for the kind of reasoning we consider, namely argumentative reasoning as understood in *computational argumentation* (Atkinson et al., 2017). This field focuses on the formal modelling and implementation of argumentative reasoning within artificial systems. It seeks to endow machines with the capacity to engage in structured debates, resolve conflicting information, and justify conclusions in a manner analogous to human argumentative practices. The field is grounded in two main paradigms: abstract argumentation (Dung, 1995), where arguments are treated as atomic entities connected through attack relations; and structured argumentation (Besnard et al., 2014), which accounts for the internal structure of arguments, including premises, inference rules, and domain knowledge. These frameworks are evaluated using various semantics that determine the acceptability of arguments based on logical coherence and conflict resolution. Computational argumentation has gained prominence for its role in supporting explainable AI (Čyras et al., 2021), particularly in domains where transparency, reasoning under uncertainty, and stakeholder dialogue are essential, such as legal analysis, medical diagnosis, and automated negotiation. In this thesis, we focus on two forms of computational argumentation, briefly reviewed next.

2.3.1 *(Quantified) Bi-polar Argumentation Frameworks*

BAFs (Cayrol and Lagasquie-Schiex, 2005), are described with triples $\langle \mathit{Args}, \mathbf{A}, \mathbf{S} \rangle$ such that Args is a finite set of *arguments*, $\mathbf{A} \subseteq \mathit{Args} \times \mathit{Args}$ is

1 E.g. for a CNN with 500K parameters with batch size 32, $\geq$ 125GB of GPU memory is needed

an *attack* relation and $\mathbf{S} \subseteq \mathit{Args} \times \mathit{Args}$ is a *support* relation. For all BAFs here, we assume that $\mathbf{A} \cap \mathbf{S} = \emptyset$. For any $\alpha \in \mathit{Args}$, $\mathbf{A}(\alpha) = \{\beta \in \mathit{Args} | (\beta, \alpha) \in \mathbf{A}\}$ are the *attackers* of α and $\mathbf{S}(\alpha) = \{\beta \in \mathit{Args} | (\beta, \alpha) \in \mathbf{S}\}$ are the *supporters* of α. Given BAFs $\mathbf{B} = \langle \mathit{Args}, \mathbf{A}, \mathbf{S} \rangle$, $\mathbf{B}' = \langle \mathit{Args}', \mathbf{A}', \mathbf{S}' \rangle$, we say that $\mathbf{B} \sqsubseteq \mathbf{B}'$ iff $\mathit{Args} \subseteq \mathit{Args}'$, $\mathbf{A} \subseteq \mathbf{A}'$ and $\mathbf{S} \subseteq \mathbf{S}'$; also, we use $\mathbf{B}' \setminus \mathbf{B}$ to denote $\langle \mathit{Args}' \setminus \mathit{Args}, \mathbf{A}' \setminus \mathbf{A}, \mathbf{S}' \setminus \mathbf{S} \rangle$. We also say that $\mathbf{B} = \mathbf{B}'$ iff $\mathbf{B} \sqsubseteq \mathbf{B}'$ and $\mathbf{B}' \sqsubseteq \mathbf{B}$, and $\mathbf{B} \sqsubset \mathbf{B}'$ iff $\mathbf{B} \sqsubseteq \mathbf{B}'$ but $\mathbf{B} \neq \mathbf{B}'$.

As conventional in the literature Baroni et al., 2019, a BAF $\langle \mathit{Args}, \mathbf{A}, \mathbf{S} \rangle$ may be equipped with *gradual semantics* $\sigma : \mathit{Args} \to \mathbb{I}$ assigning to arguments $\alpha \in \mathit{Args}$ values in $\mathbb{I}$, which is some set equipped with a pre-order $\leq$ (where, as usual $v < w$ denotes $v \leq w$ and $w \not\leq v$). In line with Rago et al., 2023, we refer to σ as *evaluation method* and to $\mathbb{I}$ as *evaluation range*, to indicate their use by agents to evaluate arguments internally. We say σ is *dialectically monotonic* iff, as in Baroni et al., 2019:

- given two BAFs $\mathbf{B} = \langle \mathit{Args}, \mathbf{A}, \mathbf{S} \rangle$ and $\mathbf{B}' = \langle \mathit{Args}', \mathbf{A}', \mathbf{S}' \rangle$, where $\mathit{Args}' = \mathit{Args} \cup \{\alpha\}$, $\mathbf{A}' \cup \mathbf{S}' = \mathbf{A} \cup \mathbf{S} \cup \{(\alpha, \beta)\}$, it is always the case that if $(\alpha, \beta) \in \mathbf{A}'$, then $\sigma(\mathbf{B}', \beta) \leq \sigma(\mathbf{B}, \beta)$, while if $(\alpha, \beta) \in \mathbf{S}'$, then $\sigma(\mathbf{B}', \beta) \geq \sigma(\mathbf{B}, \beta)$; and
- given two BAFs $\mathbf{B} = \langle \mathit{Args}, \mathbf{A}, \mathbf{S} \rangle$ and $\mathbf{B}' = \langle \mathit{Args}', \mathbf{A}', \mathbf{S}' \rangle$, where for an argument $\alpha \in \mathit{Args} \cap \mathit{Args}'$, $\mathbf{A}'(\alpha) = \mathbf{A}(\alpha)$, $\mathbf{S}'(\alpha) = \mathbf{S}(\alpha)$, $\exists \beta \in \mathbf{A}'(\alpha) \cup \mathbf{S}'(\alpha)$ such that $\sigma(\mathbf{B}', \beta) > \sigma(\mathbf{B}, \beta)$ and $\forall \gamma \in \mathbf{A}'(\alpha) \cup \mathbf{S}'(\alpha) \setminus \{\beta\}$, $\sigma(\mathbf{B}', \gamma) = \sigma(\mathbf{B}, \gamma)$ it is always the case that if $\beta \in \mathbf{A}'(\alpha)$, then $\sigma(\mathbf{B}', \alpha) \leq \sigma(\mathbf{B}, \alpha)$, while if $\beta \in \mathbf{S}'(\alpha)$, then $\sigma(\mathbf{B}', \alpha) \geq \sigma(\mathbf{B}, \alpha)$.

The first bullet states that an argument's strength cannot increase/decrease when a new attacker against/supporter for (respectively) the argument is added, all else being equal; the second bullet states that an argument's strength cannot increase/decrease when an attacker against/-supporter for (respectively) the argument is strengthened, all else being equal.

Given a BAF $\mathbf{B} = \langle \mathit{Args}, \mathbf{A}, \mathbf{S} \rangle$, for $\alpha, \beta \in \mathit{Args}$, we let a *path* from α to β be defined as $(\gamma_0, \gamma_1), \ldots, (\gamma_{n-1}, \gamma_n)$ for some $n > 0$ (the *length* of the path) where $\gamma_0 = \alpha$, $\gamma_n = \beta$ and, for any $1 \leq i \leq n$, $(\gamma_{i-1}, \gamma_i) \in \mathbf{A} \cup \mathbf{S}$. We also use $\mathsf{paths}(\alpha, \beta)$ to denote the set of all paths from α to β (leaving the BAF implicit), and use $|p|$ for the length of path p. Also, we may see paths as

sets of pairs. Then, as in Tarlé et al., 2022; Rago et al., 2023, we say that **B** is a *BAF for explanandum* $e \in Args$ iff i.) $\nexists(e, \alpha) \in \mathbf{A} \cup \mathbf{S}$; ii.) $\forall \alpha \in Args \setminus \{e\}$, there is a path from α to e; and iii.) $\nexists \alpha \in Args$ with a path from α to α.

2.3.2 *Assumption-Based Argumentation (ABA)*

ABA (Dung et al., 2009) is a well-known symbolic formalism for modelling non-monotonic reasoning. A (flat) ABA framework (Dung et al., 2009) is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ \ } \rangle$ where: (i) $\langle \mathcal{L}, \mathcal{R} \rangle$ is a deductive system where $\mathcal{L}$ is the language and $\mathcal{R}$ is the set of (inference) rules; (ii) $\mathcal{A} \subseteq \mathcal{L}$ is the set of assumptions; (iii) $\overline{\ \ }$ is a total mapping from $\mathcal{A}$ to $\mathcal{L}$ where $\bar{a}$ is referred to a contrary of a ($a \in \mathcal{A}$). The elements of $\mathcal{L}$ can be any sentences. For sake of simplicity we omit $\mathcal{L}$ as this can be derived using $\langle \mathcal{R}, \mathcal{A}, \overline{\ \ } \rangle$. We illustrate with a simple example, where, as in (Dung et al., 2009), we use schemata to write rules, assumptions and contraries, using the variable `A`.

Example 2.1. *A simple ABA (Assumption-Based Argumentation) framework for image classification is given by the tuple* $\langle \mathcal{R}, \mathcal{A}, \overline{\ \ } \rangle$*, where:*

$$
\begin{aligned}
\mathcal{R} = \{\ & \rho_1 : \texttt{circle(A) :- A = img\_1,} \\
& \rho_2 : \texttt{circle(A) :- A = img\_2,} \\
& \rho_3 : \texttt{square(A) :- A = img\_2,} \\
& \rho_4 : \texttt{c\_1(A) :- alpha(A), circle(A),} \\
& \rho_5 : \texttt{c\_alpha(A) :- square(A)}\ \}
\end{aligned}
$$

$\mathcal{A} = \{\texttt{alpha(A)}\}, \overline{\texttt{alpha(A)}} = \texttt{c\_alpha(A)}$

Here, $\mathcal{R}$ *is a set of* rules*, each named* ρ_i*, where the head of the rule precedes* `:-` *and the body follows it.* $\mathcal{A}$ *is a set of* assumptions*, in this case containing a single assumption. Each assumption is associated with a* contrary*; for instance, the contrary of* `alpha(A)` *is given by* `c_alpha(A)`*.*

We define a *fact* as a rule with distinct variables in the head and only equalities in the body. To decide which conclusions may be drawn from an ABA framework, arguments and attacks between them are first obtained,

then acceptance of arguments is determined using an extension-based semantics, in our case of *stable extensions*. An *argument* for the claim $c \in \mathcal{L}$ supported by $A \subseteq \mathcal{A}$ and $R \subseteq \mathcal{R}$ (denoted as $A \vdash_R c$) is a finite tree with nodes labelled by sentences in $\mathcal{L}$ or by τ denoting *true*, the roots labelled by c, the leaves either *true* or assumptions in A, and non-leaves c' with, as children, the elements of the body of some rule in R with the head c'. An argument $A \vdash_R c$ *attacks* an argument $A' \vdash_{R'} c'$ iff there is an assumption $a \in A'$ such that $\overline{a} = c$. A set of arguments E is *stable* iff the set is conflict-free (i.e. no argument in E attacks an an argument also in E) and for every argument not in E there is an argument in E attacking it. We illustrate these notions with the earlier example.

Example 2.2. *The following arguments can be obtained (amongst others) from the ABA framework in example 2.1:*

$$\{\texttt{alpha(img\_1)}\} \vdash_{\{\rho_1,\rho_4\}} \texttt{c\_1(img\_1)},$$
$$\{\texttt{alpha(img\_2)}\} \vdash_{\{\rho_2,\rho_4\}} \texttt{c\_1(img\_2)},$$
$$\{\} \vdash_{\{\rho_2,\}} \texttt{circle(img\_2)}$$
$$\{\} \vdash_{\{\rho_3,\rho_5\}} \texttt{c\_alpha(img\_2)}.$$

Intuitively, each argument is a deduction from (possibly empty) sets of assumptions (the premises) to claims (e.g. `c_1(img_1)` *for the first argument), using sets of rules. Attacks between arguments result from undercutting assumptions in the premises of arguments. Here, the third argument attacks the second, as the former is a deduction of the contrary of the assumption occurring in the premise of the latter. The first and third argument belongs to the single stable extension admitted by this simple ABA framework, as they cannot be attacked by any other arguments.*

To learn these frameworks, we use the ASP-ABALearn method by De Angelis et al., 2023. This takes in input a Background ABA framework (admitting at least one stable extension), sets $\mathcal{E}^+$ and $\mathcal{E}^-$ of positive and negative examples, respectively, and returns in output a Learnt ABA framework (admitting at least one stable extension) such that all positive examples are accepted in all stable extensions and no negative example is accepted in all the stable extensions. Computationally, ASP-ABALearn leverages the fact that flat ABA frameworks (where assumptions cannot be claims of arguments supported by other assumptions) can be mapped to logic

programs. This is done by replacing each assumption $\alpha(X)$ with *not* $p(X)$ where $\overline{\alpha(X)} = p(X)$.

Part-1

Foundations of Object-Centric Learning

Draṣṭuṃ nirīkṣituṃ caiva, pratītiḥ tu tad-adhikam |
Yatra dṛṣṭe'tivṛttaṃ syāt, tatra buddheḥ prakāṣanam ||

"To see is to observe; to perceive is to see beyond."

This page is intentionally left blank.

3

GROUNDING IN OBJECT-CENTRIC LEARNING

"Perception is not something that happens to us, or in us. It is something we do."

– Alva Noë

The extraction of modular object-centric representations for downstream tasks is an emerging area of research. Learning grounded representations of objects that are guaranteed to be stable and invariant promises robust performance across different tasks and environments. Slot Attention (SA) learns object-centric representations by assigning objects to *slots*, but presupposes a *single* distribution from which all slots are randomly initialised. This results in an inability to learn *specialized* slots which bind to specific object types and remain invariant to identity-preserving changes in object appearance. To address this, we present **Conditional Slot Attention** (CoSA) using a novel concept of **Grounded Slot Dictionary** (GSD) inspired by vector quantization. Our proposed GSD comprises (i) canonical object-level property vectors and (ii) parametric Gaussian distributions, which define a prior over the slots. We demonstrate the benefits of our method in multiple downstream tasks such as scene generation, composition, and task adaptation, whilst remaining competitive with SA in popular object discovery benchmarks.

3.1 MOTIVATION

The **grounding problem** refers to the challenge of connecting learnt latent representations to real-world objects, their function, and meaning (Harnad, 1990). It can be understood as the challenge of learning abstract, *canonical*

representations of objects.[1] The **binding problem** refers to the challenge of how objects are combined into a single context (Revonsuo and Newman, 1999). Both of these problems affect a system's ability to understand the world in terms of symbol-like entities, or true factors of variation, which are crucial for systematic generalization (Bengio et al., 2013a; Greff et al., 2020).

Greff et al., 2020 proposed a functional division of the binding problem into three concrete sub-tasks: (i) *segregation*: the process of forming grounded, modular object representations from raw input data; (ii) *representation*: the ability to represent multiple object representations in a common format, without interference between them; (iii) *compositionality*: the capability to dynamically relate and compose object representations without sacrificing the integrity of the constituents. The mechanism of *attention* is believed to be a crucial component in determining which objects appear to be bound together, segregated, and recalled (Vroomen and Keetels, 2010; Hinton, 2022). Substantial progress in object-centric learning has been made recently, particularly in unsupervised objective discovery, using iterative attention mechanisms like Slot Attention (SA) (Locatello et al., 2020b) and many others (Engelcke et al., 2019; Engelcke et al., 2021; Singh et al., 2021; Chang et al., 2022; Seitzer et al., 2022).

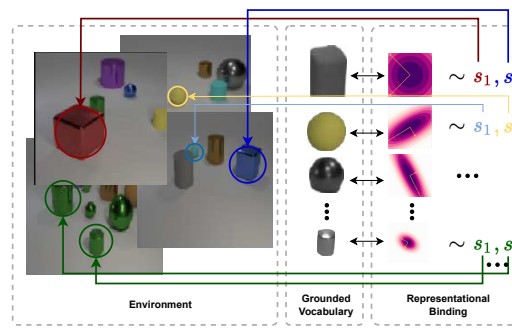


Figure 3.1: The leftmost block illustrates various scenes within an environment, each featuring different object instances. In the middle block, we depict our acquired *grounded vocabulary* of canonical object-centric representations, effectively capturing object *types*. The rightmost block displays a collection of *specialized slot* distributions associated with their respective canonical representations. These distributions are employed to sample initial slots for object instances within a scene. This process, known as *object binding*, is elucidated by the placeholder slots s_1 and s_2. These slots are linked to specific object types in the environment and undergo further refinement. Notably, this differs from the SA, which relies on a *single* distribution for random slot initialization and does not encourage slots to remain invariant in the face of identity-preserving changes in object appearance.

1 Representations learned by neural networks are directly grounded in their input data, unlike classical notions of *symbols* whose definitions are often subject to human interpretation.

Despite recent progress, several major challenges in the field of object-centric representation learning remain, including but not limited to: (i) respecting object symmetry and independence, which requires isolating individual objects irrespective of their orientation, viewpoint and overlap with other objects; (ii) dynamically estimating the total number of unique and repeated objects in a given scene; (iii) the binding of *dynamic* representations of an object with more *permanent* (canonical) identifying characteristics of its *type*.

As Treisman, 1999 explains, addressing the challenge (iii) is a pre-requisite for human-like perceptual binding. In this work, we argue this view of binding *temporary* object representations (so-called "object files") to their respective *permanent* object *types*, is a matter of learning a vocabulary of *grounded*, canonical object representations. Therefore, our primary focus is on addressing challenge (iii), which we then leverage to help tackle challenges (i) and (ii). To that end, our proposed approach goes beyond standard slot attention (SA) (Locatello et al., 2020b), as shown in Figure 3.1 and described next.

Contributions. Slot attention learns composable representations using a dynamic inference-level binding scheme for assigning objects to *slots*, but presupposes a *single* distribution from which all slots are randomly initialised. This results in an inability to learn *specialized* slots that bind to specific object types and remain invariant to identity-preserving changes in object appearance. To address this, we present *Unsupervised* C*onditional* S*lot* A*ttention* (CoSA) using a novel *Grounded Slot Dictionary* (GSD) inspired by vector quantization techniques. Our GSD is *grounded* in the input data and consists of: (i) canonical object-level property vectors which are learnable in an unsupervised fashion; and (ii) parametric Gaussian distributions defining a prior over object slots. In summary, our main contributions are as follows:

1. We propose the *Grounded Slot Dictionary* for object-centric representation learning, which unlocks the capability of learning *specialized* slots that bind to specific object *types*;

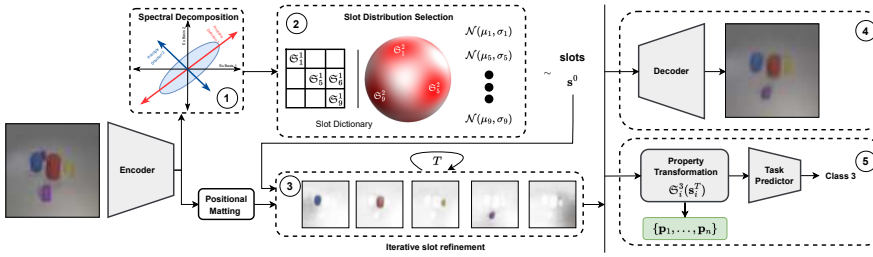


Figure 3.2: CoSA is an unsupervised autoencoder framework for *grounded* object-centric representation learning, and it is composed of five unique sub-modules. ① The **abstraction** module extracts all the *distinct* objects in a scene using spectral decomposition. ② The **grounded slot dictionary** (GSD) module maps the object representation to *grounded* (canonical) slot representations, which are then used for sampling initial slot conditions. ③ The **refinement** module uses slot attention to iteratively refine the initial slot representations. ④ The **discovery** module maps the slot representations to observational space (used for object discovery and visual scene composition). ⑤ The **reasoning** module involves object property transformation and the prediction model (used for reasoning tasks).

2. We provide a probabilistic perspective on unsupervised object-centric dictionary learning and derive a principled end-to-end objective for this model class using variational inference methods;

3. We introduce a simple strategy for dynamically quantifying the number of unique and repeated objects in a given input scene by leveraging spectral decomposition techniques;

4. Our experiments demonstrate the benefits of grounded slot representations in multiple tasks such as scene generation, composition and task adaptation whilst remaining competitive with standard slot attention in object discovery-based tasks.

3.2 Unsupervised conditional slot attention: formalism

In this section, we present our proposed conditional slot attention (CoSA) framework using a novel *grounded* slot dictionary (GSD) inspired by vector quantization. CoSA is an unsupervised autoencoder framework for *grounded* object-centric representation learning, and it consists of five sub-modules in total (as depicted in Figure 3.2), each of which we will describe in detail next.

Notation. Let $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ denote a dataset of images $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$ and their *properties* $\mathcal{Y} \subseteq \mathcal{Y}_1 \times \mathcal{Y}_2 \in \mathbb{R}^{Y}$. There are Y *properties* in total, where $\mathcal{Y}_1$ corresponds to the space of image labels, and $\mathcal{Y}_2$ consists of additional information about the images like object size, shape, location, object material, etc. Let $\Phi_e : \mathcal{X} \to \mathcal{Z}$ denote an encoder and $\Phi_d : \mathcal{Z} \to \mathcal{X}$ denote a decoder, mapping to and from a latent space $\mathcal{Z}$. Further, let $\Phi_r : \mathcal{Z} \to \mathcal{Y}$ denote a classifier for *reasoning* tasks.

(1) **Abstraction Module.** The purpose of the abstraction module is to enable dynamic estimation of the number of slots required to represent the input. Since multiple instances of the same object can appear in a given scene, we introduce the concept of an *abstraction function* denoted as: $\mathcal{A} : \mathbb{R}^{N \times d_z} \to \mathbb{R}^{\tilde{N} \times d_z}$, which maps N input embeddings to $\tilde{N}$ output embeddings representing $\tilde{N}$ *distinct* objects. This mapping learns the canonical vocabulary of unique objects present in an environment, as shown in Figure 3.1. To accomplish this, we first compute the latent covariance matrix $\boldsymbol{C} = \mathbf{z} \cdot \mathbf{z}^T \in \mathbb{R}^{N \times N}$, where $\mathbf{z} = \Phi_e(\mathbf{x}) \in \mathbb{R}^{N \times d_z}$ are feature representations of an input $\mathbf{x}$. We then perform a spectral decomposition resulting in $\mathbf{C} = \boldsymbol{V} \Lambda \boldsymbol{V}^T$, where $\boldsymbol{V}$ and $\text{diag}(\Lambda)$ are the eigenvector and eigenvalue matrices, respectively. The eigenvectors in $\mathbf{V}$ correspond to the directions of maximum variation in $\mathbf{C}$, ordered according to the eigenvalues, which represent the respective magnitudes of variation. We project $\mathbf{z}$ onto the top $\tilde{N}$ principal vectors, resulting in abstracted, property-level representation vectors in a new coordinate system (principal components).

The spectral decomposition yields high eigenvalues when: (i) a single uniquely represented object spans multiple input embeddings in $\mathbf{z}$ (i.e. a large object is present in $\mathbf{x}$); (ii) a scene contains multiple instances of the same object. To accommodate both scenarios, we consider the maximum area spanned by an object (excluding background) as a hyper-parameter with its corresponding eigenvalue denoted by λ_s. Under this assumption, we can dynamically estimate the number of slots required to represent the input whilst preserving maximal explained variance. To that end, we first filter out small eigenvalues by flooring them $\lambda_i = \lfloor \lambda_i \rfloor$, then compute a sum of eigenvalue ratios w.r.t. λ_s resulting in a total number of slots required: $K = 1 + \sum_{i=2}^{\tilde{N}} \lceil \lambda_i / \lambda_s \rceil$. Intuitively, this ratio dynamically estimates the number of object instances, relative to the 'largest' object in the input.

Note that we do not apply positional embedding to the latent features at this stage, as it encourages multiple instances of the same object to be uniquely represented, which we would like to avoid.

(2) **GSD Module.** To obtain grounded object-centric representations that connect scenes with their fundamental building blocks (i.e. object *types* (Treisman, 1999)), we introduce GSD, as defined in 3.1.

Definition 3.1 (Grounded Slot Dictionary (GSD))**.** *A grounded slot dictionary* $\mathfrak{S}$ *consists of: (i) an object-centric representation codebook* $\mathfrak{S}^1$*; (ii) a set of slot prior distributions* $\mathfrak{S}^2 = \mathcal{N}\left(\mathbf{s}_i^0; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\right)$*. The number of objects* M *in the environment is predefined, and the GSD* $\mathfrak{S}$ *is constructed as follows: Formally, the dictionary is defined as:*

$$\mathfrak{S} := \left\{ \left(\mathfrak{S}_i^1, \mathfrak{S}_i^2 \right) \right\}_{i=1}^{M}. \tag{3.1}$$

The sampling criterion used during both training and inference is described in Definition 3.2.

Definition 3.2 (GSD Sampling)**.** *Let* $\mathbf{x}$ *be an input scene, and let* $\mathbf{z} = \Phi_e(\mathbf{x}) \in \mathbb{R}^{N \times d_z}$ *denote its encoded representation, where* N *is the number of latent tokens and* d_z *is the embedding dimension. An abstraction function* $\mathcal{A}(\cdot) : \mathbb{R}^{N \times d_z} \to \mathbb{R}^{\tilde{N} \times d_z}$ *is applied to* $\mathbf{z}$*. Finally, for each abstracted summary vector, we retrieve the closest codebook vector from* $\mathfrak{S}^1$ *using a distance function* $\hat{d}(\cdot, \cdot)$*:*

$$\tilde{\mathbf{z}} = \underset{\mathfrak{S}_i^1 \in \mathfrak{S}^1}{\arg\min}\, \hat{d}\left(\mathcal{A}(\mathbf{z}), \mathfrak{S}_i^1 \right), \quad \tilde{\mathbf{z}} \in \mathbb{R}^{\tilde{N} \times d_z}. \tag{3.2}$$

These selected vectors $\tilde{\mathbf{z}}$ *and the corresponding slot priors* $\mathfrak{S}_i^2$ *are then used in downstream reasoning or training tasks.*

As per Equation 3.2, the codebook $\mathfrak{S}^1$ induces a mapping from input-dependent continuous representations $\mathbf{z}' = \mathcal{A}(\Phi_e(\mathbf{x}))$, to a set of discrete (canonical) object-centric representations $\tilde{\mathbf{z}}$ via a distance function $\hat{d}(\cdot, \cdot)$ (judicious choices for $\hat{d}(\cdot, \cdot)$ are discussed in Chapter 2). Each slot prior distribution $p(\mathbf{s}_i^0) = \mathcal{N}\left(\mathbf{s}_i^0; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\right)$ in $\mathfrak{S}^2$ is associated with one of the M object representations in the codebook $\mathfrak{S}^1$. These priors define marginal distributions over the initial slot representations $\mathbf{s}^0$. We use diagonal covariances and the learn the parameters $\{\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\}_{i=1}^M$ during training.

(3) **Iterative Refinement Module.** After randomly sampling initial slot conditions from their respective marginal distributions in the grounded slot dictionary $\mathbf{s}^0 \sim \mathfrak{S}(\mathbf{z}) \in \mathbb{R}^{K \times d_z}$, we iteratively refine the slot representations using slot attention as described in Chapter 2 and Algorithm 1. The subset of slot priors we sample from for each input corresponds to the K respective codebook vectors which are closest to the output of the abstraction function $\mathbf{z}' = \mathcal{A}(\Phi_e(\mathbf{x}))$, as outlined in Equation 3.2.

Remark (Slot Posterior): The posterior distribution of the slots $\mathbf{s}^{t=T}$ at iteration T given $\mathbf{x}$ is:

$$p(\mathbf{s}^T \mid \mathbf{x}) = \delta\left(\mathbf{s}^T - \mathcal{H}_\theta\left(\mathbf{s}^{T-1}, f(g(\hat{\mathbf{q}}^{T-1}, \mathbf{k}), \mathbf{v})\right)\right), \tag{3.3}$$

where $\delta(\cdot)$ is Dirac delta distributed given randomly sampled initial slots from their marginals $\mathbf{s}^0 \sim p(\mathbf{s}^0 \mid \tilde{\mathbf{z}}) = \prod_{i=1}^{K} \mathcal{N}\left(\mathbf{s}_i^0; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\right)$, associated with the codebook vectors $\tilde{\mathbf{z}} \sim q(\tilde{\mathbf{z}} \mid \mathbf{x})$. The distribution over the initial slots $\mathbf{s}^0$ induces a distribution over the refined slots $\mathbf{s}^T$. One important aspect worth re-stating here is that the sampling of the initial slots $\left\{\mathbf{s}_i^0 \sim p(\mathbf{s}_i^0 \mid \tilde{\mathbf{z}}_i)\right\}_{i=1}^{K} \subset \mathfrak{S}^2$ depends on the *indices* of the associated codebook vectors $\tilde{\mathbf{z}} \subset \mathfrak{S}^1$, as outlined in Definition 3.1, and is not conditioned on the values of $\tilde{\mathbf{z}}$ themselves. In Equation 3.3, $\mathcal{H}_\theta\left(\mathbf{s}^{T-1}, f(g(\hat{\mathbf{q}}^{T-1}, \mathbf{k}), \mathbf{v})\right)$ reflects the final slot in the sequence (or the final output of the routing iteration).

(4) **Discovery Module.** For object discovery and scene composition tasks, we need to translate object-level slot representations into the observational space of the data. To achieve this, we use a spatial broadcast decoder, as employed by IODINE and slot attention (Greff et al., 2019; Locatello et al., 2020b). This decoder reconstructs the image $\mathbf{x}$ via a softmax combination of K individual reconstructions from each slot. Each reconstructed image $\mathbf{x}_s = \Phi_d(\mathbf{s}_k)$ consists of four channels: RGB plus a mask. The masks are normalized over the number of slots K using softmax, therefore they represent whether each slot is bound to each pixel in the input.

We can now define a generative model of $\mathbf{x}$ which factorises as: $p(\mathbf{x}, \mathbf{s}^0, \tilde{\mathbf{z}}) = p(\mathbf{x} \mid \mathbf{s}^0)p(\mathbf{s}^0 \mid \tilde{\mathbf{z}})p(\tilde{\mathbf{z}})$, recalling that $\mathbf{s}^0 \coloneqq \mathbf{s}_1^0, \ldots, \mathbf{s}_K^0$ are the initial slots at iteration $t = 0$, and $\tilde{\mathbf{z}}$ are our discrete latent variables described in Definition 3.1. For training, we derive a variational lower bound on the marginal

Algorithm 1 Conditional Slot Attention (CoSA).

Input: inputs $\mathbf{z} = \Phi_e(\mathbf{x}) \in \mathbb{R}^{N \times d_z}$, $\mathbf{k} = \mathcal{K}_\beta(\mathbf{z}) \in \mathbb{R}^{N \times d_s}$, and $\mathbf{v} = \mathcal{V}_\phi(\mathbf{z}) \in \mathbb{R}^{N \times d_s}$
Spectral Decomposition:
  $\mathbf{z} \cdot \mathbf{z}^T = \boldsymbol{V} \Lambda \boldsymbol{V}^T$ — V: eigenvectors, $\Lambda = \text{diag}(\lambda_i)$: eigenvalues
  $\mathbf{z}' = \boldsymbol{V}^T \cdot \mathbf{z}$ — $\mathbf{z}'$: project onto principle components
  $K = 1 + \sum_{i=2}^{\tilde{N}} \lceil \lfloor \lambda_i \rfloor / \lambda_s \rceil$ — Dynamically estimate number of slots
for $i = 0, \ldots, R$ — R Monte Carlo samples
  $\text{slots}_i^0 \sim \mathfrak{S}(\mathbf{z}') \in \mathbb{R}^{K \times d_s}$ — Sample K initial slots from GSD
  for $t = 1, \ldots, T$ — Refine slots over T attention iterations
    $\text{slots}_i^t = f\left(g\left(\mathcal{Q}_\gamma\left(\text{LayerNorm}\left(\text{slots}_i^{t-1}\right)\right), \mathbf{k}\right), \mathbf{v}\right)$
    — Update slot representations
    $\text{slots}_i^t \mathrel{+}= \text{MLP}\left(\text{LayerNorm}\left(\mathcal{H}_\theta\left(\text{slots}_i^{t-1}, \text{slots}_i^t\right)\right)\right)$
    — GRU update & skip connection
return $\sum_i \text{slots}_i / R$ — MC estimate

log-likelihood of $\mathbf{x}$, a.k.a. the Evidence Lower Bound (ELBO), as detailed in Proposition 3.1.

Proposition 3.1 (ELBO for Object Discovery)**.** *Under a categorical distribution over our discrete latent variables $\tilde{\mathbf{z}}$, and the object-level prior distributions $p(\mathbf{s}_i^0) = \mathcal{N}\left(\mathbf{s}_i^0; \boldsymbol{\mu}_i, \sigma_i^2\right)$ contained in $\mathfrak{S}^2$, we show that variational lower bound on the marginal log-likelihood of $\mathbf{x}$ can be expressed by* $\text{ELBO}(\mathbf{x}) :=$*:*

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\tilde{\mathbf{z}} \sim q(\tilde{\mathbf{z}}|\mathbf{x}), \mathbf{s}^0 \sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \left[\log p(\mathbf{x} \mid \mathbf{s})\right] - D_{\text{KL}}\left(q\left(\tilde{\mathbf{z}} \mid \mathbf{x}\right)\right) \parallel p(\tilde{\mathbf{z}})\right), \quad (3.4)$$

where $\mathbf{s} := \prod_{t=1}^{T} \mathcal{H}_\theta\left(\mathbf{s}^{t-1} \mid f(g(\hat{\mathbf{q}}^{t-1}, \mathbf{k}), \mathbf{v})\right)$ denotes the output of the iterative refinement procedure described in Algorithm 1 applied to the initial slot $\mathbf{s}^0$ representations.

The proof is given in Appendix A.2.

3.3 EXPERIMENTS

Our empirical study evaluates CoSA on a variety of popular *object discovery* (Table 3.1, 4.3). In an *object discovery* context, we demonstrate our method's

Table 3.1: Foreground ARI on CLEVR, Tetrominoes, ObjectsRoom, and COCO datasets, for all baseline models and CoSA-cosine variant. For COCO we use the DINOSAUR variant (Seitzer et al., 2022) (with ViT-S16 as feature extractor) as a baseline model and use the same architecture for IMPLICT and CoSA (adaptation details in Appendix A.8.3).

Method	Tetrominoes	CLEVR	ObjectsRoom	COCO
SA (Locatello et al., 2020b)	0.99 ± 0.005	0.93 ± 0.002	0.78 ± 0.02	-
BlockSlot (Singh et al., 2022a)	0.99 ± 0.001	0.94 ± 0.001	0.77 ± 0.01	-
SlotVAE (Wang et al., 2023a)	-	-	0.79 ± 0.01	-
DINOSAUR (Seitzer et al., 2022)	-	-	-	0.28 ± 0.02
IMPLICIT (Chang et al., 2022)	0.99 ± 0.001	0.93 ± 0.001	0.78 ± 0.003	0.28 ± 0.01
CoSA	0.99 ± 0.001	**0.96** ± 0.002	**0.83** ± 0.002	**0.32** ± 0.01



(a) Embedding-7 ($\mathfrak{S}^1_7$) **(b)** Embedding-14 ($\mathfrak{S}^1_{14}$) **(c)** Embedding-25 ($\mathfrak{S}^1_{25}$) **(d)** Embedding-55 ($\mathfrak{S}^1_{55}$)

Figure 3.3: GSD binding: we can observe that *cheeks* being bound to $\mathfrak{S}^1_7$, *forehead* to $\mathfrak{S}^1_{14}$, *eyes* to ($\mathfrak{S}^1_{25}$), and *facial hair* to $\mathfrak{S}^1_{55}$, illustrating the notion of object binding achieved in GSD, in the case of bitmoji dataset for CoSA model trained with cosine sampling stratergy.

ability to: (i) dynamically estimate the number of slots required for each input (Figure 3.4); (ii) map the grounded slot dictionary elements to particular object *types* (Figure 3.3); (iii) perform reasonable slot composition without being explicitly trained to do so (Figure 3.5). We also provide detailed ablation studies on: (i) the on choice of codebook sampling (Appendix A.8); (ii) different codebook regularisation techniques to avoid collapse (Appendix A.7); (iii) the choice of abstraction function (Appendix A.9). For details on the various datasets we used, please refer to Appendix A.3. For training details, hyperparameters, initializations and other computational aspects refer Appendix A.12.

In this study, we provide a thorough evaluation of our framework for object discovery across multiple datasets, including: CLEVR (Johnson et al., 2017), Tetrominoes (Kabra et al., 2019), Objects-Room (Kabra et al., 2019), Bitmoji (Mozafari, 2020), FFHQ (Karras et al., 2020), and COCO (Lin et al., 2014). In terms of evaluation metrics, we use the foreground-adjusted Rand Index score (ARI) and compare our method's results with standard SA (Locatello et al., 2020b), SlotVAE (Wang et al., 2023a), BlockSlot (Singh et al.,

2022a), IMPLICIT (Chang et al., 2022) and DINO (Seitzer et al., 2022). We evaluated the majority of the baseline models on the considered datasets using their original implementations, with the exception of SlotVAE, where we relied on the original results from the paper. For DINOSAUR, we specifically assessed its performance on a real-world COCO dataset, following the code adaptation outlined in Appendix A.8.3. Additionally, we measured the reconstruction score using MSE and the FID (Heusel et al., 2017). We analyse three variants of CoSA: Gumbel, Euclidian, and Cosine, corresponding to three different sampling strategies used for the GSD. Detailed analysis and ablations on different types of abstraction function are detailed in Appendix A.9. The inclusion of GSD provides two notable benefits: (i) it eliminates the need for a hard requirement of parameter K during inference, and (ii) since slots are initialized with samples from their grounded counterparts, this leads to improved initialization and allows us to average across multiple samples, given that the slots are inherently aligned. These design advantages contribute to the enhanced representational quality for object discovery, as evidenced in Table 3.1 and 4.3, while also enhancing generalizability which will be demonstrated later. To evaluate the quality of the generated slots without labels, we define a some additional metrics like the overlapping index (OPI) and average slot FID (SFID) (ref. Section A.4, Table A.6).

Figure 3.4 showcases the reconstruction quality of CoSA and its adaptability to varying slot numbers. For additional results on all datasets and a method-wise qualitative comparison, see Appendix A.8. Regarding the quantitative assessment of slot number estimation, we calculate the mean absolute error (MAE) by comparing the estimated slot count to the actual number of slots. Specifically, the MAE values for the CLEVR, Tetrominoes, and Objects-Room datasets are observed to be **2.06, 0.02**, and **0.32**, respectively. On average, dynamic slot number estimation reduces **38%**, **26%**, and **23%** FLOPs on CLEVR, Objects-Room, and Tetrominoes as opposed to fixed number of slots. In Figure 3.3, we illustrate the grounded object representations within GSD, which are obtained by visualising the obtained slots as a result of sampling a particular element in the GSD. Furthermore, we explore the possibility of generating novel scenes using the learned GSD, although out models were not designed explicitly for this purpose. To that

Table 3.2: Object discovery results on multiple benchmarking datasets and previously existing methods. We measure both MSE and FID metrics to compare the results.

Method	CLEVR		Tetrominoes		Objects-Room		Bitmoji		FFHQ	
	MSE ↓	FID ↓	MSE ↓	FID ↓	MSE ↓	FID ↓	MSE ↓	FID ↓	MSE ↓	FID ↓
SA	6.37	38.18	1.49	3.81	7.57	38.12	14.62	12.78	55.14	54.95
Block-Slot	3.73	34.11	0.48	0.42	5.28	36.33	10.24	11.01	52.16	41.56
IMPLICIT	3.95	38.16	0.59	0.41	5.56	36.48	9.87	10.68	47.06	49.95
CoSA	**3.14**	**29.12**	**0.42**	0.41	**4.85**	**28.19**	**8.17**	**9.28**	**33.37**	**36.34**

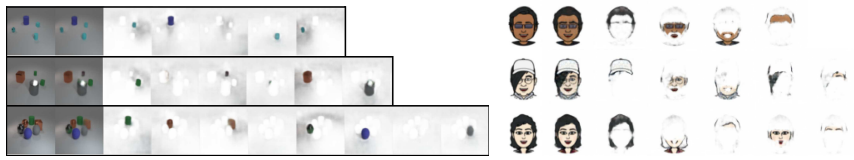


Figure 3.4: Object discovery: reconstruction quality and dynamic slot number selection for CoSA-Cosine on CLEVR and Bitmoji, with an MAE of **2.06** over slot number estimation for CLEVR.



Figure 3.5: Top and bottom left illustrates the randomly prompted slots and their composition. Right demonstrates object discovery results of CoSA on COCO dataset.

end, we sample slots from randomly selected slot distributions and decode them to generate new scenes. The prompting in our approach relies solely on the learned GSD, and we do not need to construct a prompt dictionary like SLATE (Singh et al., 2021). To evaluate scene generation capabilities, we compute FID and SFID scores on randomly generated scenes across all datasets; please refer to Appendix Table A.11. Figure 3.5 provides visual representations of images generated through slot composition on CLEVR, Bitmoji, and Tetrominoes datasets.

In this work, we propose conditional slot attention (CoSA) using a grounded slot dictionary (GSD). CoSA allows us to bind arbitrary instances of an object to a specialized canonical representation, encouraging invariance to identity-preserving changes. This is in contract to vanilla slot attention which presupposes a *single* distribution from which all slots are randomly initialised. Additionally, our proposed approach enables dynamic estimation of the number of required slots for a given scene, saving up to **38%** of FLOPs.

The main limitations of the proposed framework include: (i) limited variation in slot-prompting and composition; (ii) assumption of no representation overcrowding; (iii) maximum object area assumption. (Appendix A.13). An interesting future direction would be to use a contrastive learning approach to learn a dictionary for disentangled representation of position, shape, and texture, resulting in finer control over scene composition.

This page is intentionally left blank.

4

IDENTIFIABILITY IN OBJECT-CENTRIC LEARNING

"All things are subject to interpretation. Whichever interpretation prevails at a given time is a function of power."

– Friedrich Nietzsche

Challenges in object-centric learning include ensuring *identifiability* and handling *spatial ambiguities*, such as occlusions and viewpoint variations. While existing slot-based methods show promising object-binding capabilities, they lack strong theoretical guarantees for identifiability, which is crucial for scaling to high-dimensional images with correctness guarantees. To address this, in this chapter we propose a **probabilistic slot-attention (PSA)** algorithm that imposes an *aggregate mixture prior* over object-centric slot representations, ensuring theoretical identifiability without supervision—up to an equivalence relation. We later extend PSA to form a **viewpoint-invariant slot attention (VISA)** algorithm, which aggregates view-specific slots to disentangle *invariant object content* from *global viewpoint-level information*. Unlike prior single-view approaches, VISA explicitly resolves spatial ambiguities, maintains *theoretical identifiability guarantees*, and requires *no viewpoint annotations*. By leveraging probabilistic modelling across multiple views, VISA enables more structured, interpretable, and generalizable object representations. We provide empirical verification of this result for both algorithms on both simple two-dimensional synthetic data and high-resolution image datasets, demonstrating the effectiveness of PSA and VISA in learning structured representations.

Recent progress in OCL has been limited to learning scene representations from single-viewpoints (Locatello et al., 2020b; Engelcke et al., 2021; Singh et al., 2021; Chang et al., 2022; Seitzer et al., 2022; Kori et al., 2024b; Löwe et al., 2024). Although these approaches can learn meaningful object-specific representations, they encounter significant challenges stemming from spatial ambiguities such as occlusion and view ambiguities (see Figure 4.1 for examples). Additionally, while it has been hypothesised that these models learn un-occluded object representations even in the case of occlusions. Learning from a single viewpoint fails to capture effective object representations, due to the presence of multiple plausibilities of partially or fully occluded objects and the effects of view transformations, as demonstrated in Figure 4.1 and highlighted by the results in Figure 4.1(c) (we will revisit these results later in section 4.7). Another example of spatial ambiguities can be observed in Figure 4.8, where object $\mathcal{O}_4$ in $\mathbf{x}^1$ and $\mathbf{x}^2$ can be interpreted as a cube, but only after considering $\mathbf{x}^3$ we can conclude that being a pyramid.

A handful of approaches, including MULMON(Li et al., 2020), DYMON(Li et al., 2021), OCLOC(Yuan et al., 2024), have considered multiple viewpoints for extracting object representations. Additionally, methods such as (Chen et al., 2021; Luo et al., 2024; Liu et al., 2025) effectively use NERF(Mildenhall et al., 2021) for constructing a 3D environment from multi-viewpoint images, where the occlusions are addressed by construction. Among these methods, MULMON, DYMON, and all NERF based approaches assume that the viewpoint annotations are known, which simplifies the problem of learning to disentangle object representations conditioned on viewpoint information.

Table 4.1: Identifiability strategies (mixing function f or latent dist. $p(\mathbf{z})$), and assumptions made by object-centric learning methods.

METHOD	ASSUMPTION	IDENTIF.
Burgess et al., 2019, Emami et al., 2022, Engelcke et al., 2019, Greff et al., 2019, Locatello et al., 2020b, Wang et al., 2023a	B.1, B.2	N/A
Elsayed et al., 2022, Kipf et al., 2021, Lin et al., 2020, Van Steenkiste et al., 2020	B.1, B.2, B.5	N/A
Kori et al., 2024b	B.1, B.2, B.6	N/A
Brady et al., 2023	B.1, B.2, B.3, B.4, B.7	f
Proposed	B.1, 5.1	$p(\mathbf{z})$

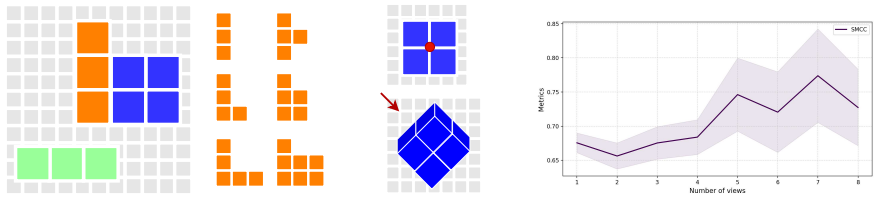


Figure 4.1: (a) Occlusion Ambiguity: the orange object, which is occluded by the blue object, could be any of the six plausible objects shown on the right. **(b) View Ambiguity:** the blue object is observed from two different viewpoints (represented with a red arrow and a dot), leading to a change in its overall shape. In general, identifiable representations resolve ambiguities by determining the most plausible object under occlusion and correct object properties in case of view transformation by leveraging information from multiple viewpoints. **(c) Identifiability results:** Identifiability across a number of views measured with Slot Mean Correlation Coefficient (SMCC).

Object-centric representations are deemed to be a key step for achieving true compositional generalization Bengio et al., 2013a; Lake et al., 2017; Battaglia et al., 2018; Greff et al., 2020, and uncovering causal influence between discrete concepts and their environment Marcus, 2003; Gopnik et al., 2004; Behrens et al., 2018; Gerstenberg et al., 2021; Schölkopf and Kügelgen, 2022. Most existing work approaches object-centric representation learning empirically, leaving theoretical understanding relatively underdeveloped. Establishing the *identifiability* Hyvärinen and Pajunen, 1999; Hyvärinen and Oja, 2000 of representations is important as it clarifies under which conditions object-centric representation learning is theoretically possible Brady et al., 2023.

A well-known result shows that identifiability of latent variables is fundamentally impossible without assumptions about the data generating process Hyvärinen and Pajunen, 1999; Locatello et al., 2019. *Therefore, understanding when object representations can theoretically be identified is important to scale object-centric methods to high-dimensional images.* Recent works Brady et al., 2023; Lachapelle et al., 2024b make important advances on this by explicitly stating the set of assumptions necessary for providing theoretical identifiability of object-centric representations. However, they restrict their attention to properties of the *mixing function*, studying a class of models with *additive* decoders. Although there are merits to this approach, there are practical challenges with the so-called *compositional contrast* objective Brady

et al., 2023, as it involves computing Jacobians and requires second-order optimization via gradient descent. Consequently, satisfying the identifiability conditions explicitly (e.g. compositional contrast must be zero) is computationally restrictive for moderately high-dimensional data. In this work, we present a probabilistic perspective that is not subject to the same scalability issues while still providing theoretical identifiability of object-centric representations without supervision. In Table 4.1, we list object-centric learning methods, their (sometimes implicit) modelling assumptions (see Section 4.3 for additional information and Appendix B.1 for a detailed breakdown and discussion), and their respective identifiability guarantees of object representations. Most methods do not guarantee identifiability, and make the $\mathcal{B}$-disentanglement (B.1) and *additive decoder* (B.2) assumptions. Brady et al. (2023) do provide identifiability guarantees and additionally assume *irreducibility* (B.3) and *compositionality* (B.4). Our method provides identifiability guarantees by introducing latent structure (i.e. via a GMM prior) which generalizes to non-additive decoders. This is advantageous as the computational complexity of additive decoders scales linearly with the number of slots K – whereas our approach is invariant to K. Moreover, non-additive decoders have been found to significantly improve performance in practice Singh et al., 2021; Seitzer et al., 2022; Singh et al., 2022b, though the theoretical basis is under explored. Finally, latent structure can reduce the complexity burden on the mixing function f (decoder), making it easier to learn in practice Falck et al., 2021; Kivva et al., 2022.

We further extend the theoretical foundation of PSA, to resolves spatial ambiguities, provides theoretical guarantees for identifiability, and requires no viewpoint annotations. To the best of our knowledge, this is the first work addressing explicit formalisations of assumptions and theory required for achieving *identifiable* object representations under occlusions with multi-view observational data. To this end, we make use of the spatial Gaussian mixture models(GMM) in latent distribution across viewpoints to encourage identifiability without additional auxiliary data. Our main contributions in this work can be summarised as follows:

Contributions. Our main contributions are the following:

(i) We establish that object-centric representations (i.e., slots) are identifiable without supervision, up to an equivalence relation, under a latent

mixture model framework (Section 4.3). To achieve this, we introduce a PSA algorithm (Section 4.2) that enforces an *aggregate* mixture prior over slot representations, forming a non-degenerate global GMM by integrating per-datapoint local GMMs. This prior remains empirically stable across runs, being identifiable up to affine transformations and slot permutations, and allows for efficient sampling.

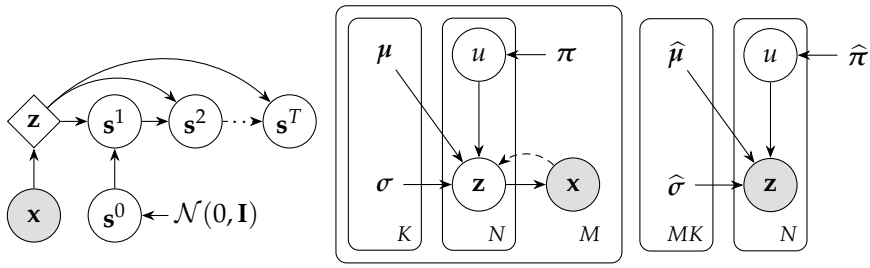
(ii) We demonstrate the effectiveness of our approach with strong empirical validation. Our theoretical identifiability results are supported by visual verification on synthetic two-dimensional data and evaluations on standard imaging benchmarks (Section 4.4).

(iii) We introduce *View-Invariant Slot Attention (VISA)*, a variant of probabilistic slot attention that learns identifiable object-centric representations from multiple viewpoints, resolving spatial ambiguities such as occlusions and viewpoint variations (Section 4.5). We prove that our method ensures identifiability in the presence of partial or full occlusions, without requiring additional viewpoint information, up to an equivalence relation with a mixture model specification (Section 4.6).

(iv) We further validate our method through visual verification on synthetic datasets and demonstrate its scalability on two new, complex multi-view datasets, MVMOVI-C and MVMOVI-D (Section 4.7).

4.2 PROBABILISTIC SLOT ATTENTION

In this section, we present a probabilistic slot attention framework which imposes a mixture prior structure over the slot latent space. This structure will prove to be instrumental in establishing our main identifiability result in Section 4.3. We begin by approaching standard slot attention (Locatello et al., 2020b) from a graphical modelling perspective. As shown in Figure 4.2 and explained in Chapter 2, applying slot attention to a deterministic encoding $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ yields a set of K object slot representations $\mathbf{s}_{1:K} \coloneqq \mathbf{s}_1, \ldots, \mathbf{s}_K$. This process induces a stochastic encoder $q(\mathbf{s}_{1:K} \mid \mathbf{x})$, where the stochasticity comes from the random initialization of the slots: $\mathbf{s}_{1:K}^{t=0} \sim \mathcal{N}(\mathbf{s}_{1:K}; 0, \mathbf{I}) \in \mathbb{R}^{K \times d}$. Since each slot is a deterministic function



(a) Slot Attention Encoder **(b)** Probabilistic Slot Attention **(c)** Aggregate Posterior

Figure 4.2: Graphical models of probabilistic slot attention. **(a)** Stochastic encoder of standard slot attention (Locatello et al., 2020b) with T attention iterations. **(b)** Proposed model – each image in the dataset $\{\mathbf{x}_i\}_{i=1}^M$ is encoded into a respective latent representation $\mathbf{z} \in \mathbb{R}^{N \times d}$, to which a (local) Gaussian mixture model with K components is fit via expectation maximisation. The resulting K Gaussians serve as slot posterior distributions: $\mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \sigma_k^2)$, for $k = 1, \ldots, K$. **(c)** Aggregate posterior distribution obtained by marginalizing out the data: $q(\mathbf{z}) = \sum_{i=1}^M q(\mathbf{z} \mid \mathbf{x}_i)/M$. We prove that $q(\mathbf{z})$ is a tractable, non-degenerate Gaussian mixture distribution which: (i) serves as the theoretically optimal prior over slots; (ii) is empirically stable across runs (i.e. identifiable up to an affine transformation and slot permutation); (iii) can be tractably sampled from and used for scene composition tasks.

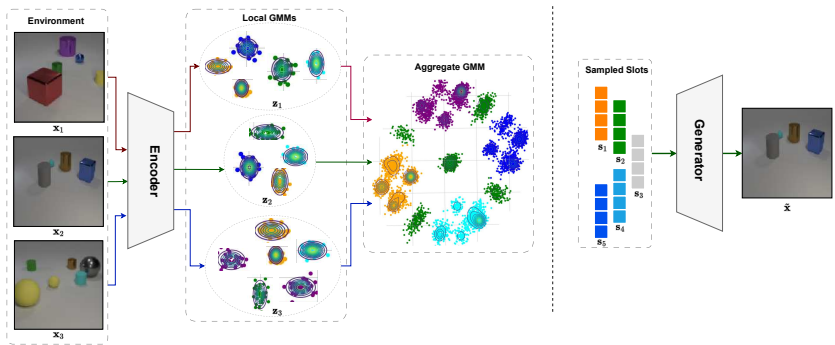


Figure 4.3: Probabilistic slot attention and the identifiable aggregate slot posterior. (Left) Slot posterior GMMs per datapoint (local) and the induced *aggregate* posterior GMM (global). (Right) Sampling slot representations from the aggregate slot posterior is tractable.

of its previous state $\mathbf{s}^t := f_s(\mathbf{z}, \mathbf{s}^{t-1})$ it is possible to randomly sample initial states $\mathbf{s}^0$ and obtain stochastic estimates of the slots.[1] However, since each transition depends on $\mathbf{z}$, which in turn depends on the input $\mathbf{x}$, *we do not get a generative model we can tractably sample from*. This can conceivably be remedied by placing a tractable prior over $\mathbf{z}$ and using the

1 Note that we may use $\mathbf{s}^t$ or $\mathbf{s}(t)$ interchangeably to denote slot representations at slot attention iteration t.

VAE framework along the lines of (Wang et al., 2023b), however, here *we propose an entirely different approach which does not require making additional variational approximations* (see Appendix 4.2 for further discussion).

LOCAL SLOT MIXTURES. Probabilistic slot attention augments standard slot attention by introducing a per-datapoint (i.e. local) GMM for learning slot distributions. Intuitively, a local GMM can be understood as a way to cluster features within a given image, encouraging the grouping of similar features into object representations. However, unlike regular clustering, here the clustered points are dynamically transformed representations of the actual data. Specifically, we use an encoder function f_e that maps each image $\mathbf{x}_i \in \mathbb{R}^{H\times W\times C}$ in the dataset $\{\mathbf{x}_i\}_{i=1}^M$, to a latent spatial representation $\mathbf{z}_i \in \mathbb{R}^{N\times d}$. The latent variable $\mathbf{z}$ may be deterministic or stochastic, and we consider the case where $N < HW$ to reflect a modest downscaling `w.r.t.` $\mathbf{x}$. The goal is to dynamically map each of the N, d-dimensional vector representations in each $\mathbf{z}$, to one-of-K object slot distributions within a mixture. A *local* GMM can be fit to each posterior latent representation $\mathbf{z} \sim q(\mathbf{z} \mid \mathbf{x}_i)$[2] on the fly by maximizing likelihood:

$$p(\mathbf{z} \mid \boldsymbol{\pi}_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_{ik} \mathcal{N}(\mathbf{z}_n; \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik}^2), \tag{4.1}$$

where $\boldsymbol{\mu}_i = (\boldsymbol{\mu}_{i1}, \ldots, \boldsymbol{\mu}_{iK})$, $\boldsymbol{\sigma}_i^2 = (\boldsymbol{\sigma}_{i1}^2, \ldots, \boldsymbol{\sigma}_{iK}^2)$ and $\boldsymbol{\pi}_i = (\pi_{i1}, \ldots, \pi_{iK})$ are the respective means, diagonal covariances and mixing coefficients of the i^{th} K-component mixture. Figure 4.2b illustrates the resulting probabilistic graphical model (PGM) in more detail. To maximize the likelihood in Equation 4.1 per datapoint $\mathbf{x}_i$, we present a bespoke expectation-maximisation (EM) algorithm for slot attention, yielding closed-form update equations for the parameters as shown in Algorithm 2, and explained next.

PROBABILISTIC PROJECTIONS. A powerful property of slot attention and cross-attention more broadly (Vaswani et al., 2017), is its ability to decouple the *agreement* mechanism from the representational content. That is, the dot-product is used to measure agreement between each *query* (slot)

2 The parametric form of q can be e.g. Gaussian or Dirac delta.

Algorithm 2 Probabilistic Slot Attention.

Input: $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ ▷ representation
$\mathbf{k} \leftarrow \boldsymbol{W}_k \mathbf{z} \in \mathbb{R}^{N \times d}$ ▷ compute keys
$\mathbf{v} \leftarrow \boldsymbol{W}_v \mathbf{z} \in \mathbb{R}^{N \times d}$ ▷ optional $\mathbf{v} := \mathbf{k}$
$\forall k,\ \boldsymbol{\pi}(0)_k \leftarrow \frac{1}{K},\ \boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d),\ \boldsymbol{\sigma}(0)_k^2 \leftarrow \mathbf{1}_d$
for $t = 0, \ldots, T-1$
  $A_{nk} \leftarrow \dfrac{\boldsymbol{\pi}(t)_k \mathcal{N}\left(\mathbf{k}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_k, \boldsymbol{\sigma}(t)_k^2\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\mathbf{k}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_j, \boldsymbol{\sigma}(t)_j^2\right)}$
  $\hat{A}_{nk} \leftarrow A_{nk} / \sum_{l=1}^{N} A_{lk}$ ▷ normalize
  $\boldsymbol{\mu}(t+1)_k \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \mathbf{v}_n$ ▷ update slots
  $\boldsymbol{\sigma}(t+1)_k^2 \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \left(\mathbf{v}_n - \boldsymbol{\mu}(t+1)_k\right)^2$
  $\boldsymbol{\pi}(t+1)_k \leftarrow \sum_{n=1}^{N} A_{nk} / N$ ▷ update mixing
return $\boldsymbol{\mu}(T), \boldsymbol{\sigma}(T)^2$ ▷ K slot distributions

vector and all the *key* vectors, to dictate how much of each *value* vector (content from $\mathbf{z}_i$) should be represented in each slot's revised representation. To retain this flexibility and decouple the attention computation from the content, we incorporate key-value projections into our probabilistic approach. For brevity, the i subscript is implicit in the following, keeping in mind that these are local quantities (per-datapoint $\mathbf{x}_i$). The parameters of the K Gaussian slot distributions are initialized (at attention iteration $t = 0$) as follows:

$$\forall k, \qquad \boldsymbol{\pi}(0)_k = K^{-1}, \qquad \boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d), \qquad \boldsymbol{\sigma}(0)_k^2 = \mathbf{1}_d. \tag{4.2}$$

The respective queries $\mathbf{q}$, keys $\mathbf{k}$, and values $\mathbf{v}$ are then given by:

$$\mathbf{q}(t) = \boldsymbol{W}_q \boldsymbol{\mu}(t), \qquad \mathbf{k} = \boldsymbol{W}_k \mathbf{z}, \qquad \mathbf{v} = \boldsymbol{W}_v \mathbf{z}, \tag{4.3}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$, whereas $\mathbf{q}(t)$ denotes the queries at attention iteration t. To measure agreement between each input feature (key) and slot

(query), we evaluate the normalized probability density of each key under a Gaussian model defined by each slot:

$$A_{nk} = \frac{1}{Z}\boldsymbol{\pi}(t)_k \mathcal{N}\left(\mathbf{k}_n; \mathbf{q}(t)_k, \boldsymbol{\sigma}(t)_k^2\right),$$
$$Z = \sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\mathbf{k}_n; \mathbf{q}(t)_k, \boldsymbol{\sigma}(t)_j^2\right), \tag{4.4}$$

where A_{nk} corresponds to the posterior probability that slot (query) k is responsible for input feature (key) n. This process yields the slot attention matrix $\boldsymbol{A} \in \mathbb{R}^{N\times K}$. As shown in Algorithm 5, the mixture parameters $\boldsymbol{\pi}(t), \boldsymbol{\mu}(t), \boldsymbol{\sigma}(t)^2$ are then updated using the attention matrix and the values $\mathbf{v}$. If the values are chosen to be equal to the keys $\mathbf{v} := \mathbf{k}$, then the procedure is more in line with standard EM, but the agreement mechanism and the content become entangled. After T probabilistic slot attention iterations, the resulting K Gaussians serve as slot posterior distributions:

$$\mathbf{s}(T)_k \sim \mathcal{N}\left(\boldsymbol{\mu}(T)_k, \boldsymbol{\sigma}(T)_k^2\right), \qquad \text{for } k = 1, \ldots, K, \tag{4.5}$$

where $\boldsymbol{\mu}(T)$ and $\boldsymbol{\sigma}(T)^2$ are the parameters of all the Gaussians in the mixture given a particular datapoint $\mathbf{x}$. The slots $\mathbf{s}(T)_{1:K}$ are then used for input reconstruction, e.g. by maximizing a (possibly Gaussian) likelihood $p(\mathbf{x} \mid \mathbf{s}(T)_{1:K})$ parameterized by a (possibly additive) decoder f_d.

PSA AND EM: As a result of projecting with a separate query matrix, the mean and variance updates are coupled with some affine transformation; for EM to have an exact solution, this transformation needs to be the identity (this can be seen when we derive the update equations for mean and variance (Bishop, 2006)). PSA, as described in Algorithm 3, closely mirrors the structure of the EM algorithm for GMMs in particular. The input features $\mathbf{z}_n = f_e(\mathbf{x})$ act as observed data, while the latent slots, parametrized by means $\boldsymbol{\mu}_k$, variances $\boldsymbol{\sigma}_k^2$, and mixing coefficients $\boldsymbol{\pi}_k$ correspond to the mixture components. At each iteration t, the attention weights A_{nk} serve as soft responsibilities (E-step), assigning input $\mathbf{z}_n$ to slot k based on a Gaussian likelihood. Notably, the slot means are projected via a learned matrix $\boldsymbol{W}_q$, introducing an affine transformation that breaks the exact probabilistic

Algorithm 3 Probabilistic Slot Attention (no **k** or **v**)

Input: $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ ▷ input representation

$\forall k,\ \boldsymbol{\pi}(0)_k \leftarrow 1/K,\ \boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d),\ \boldsymbol{\sigma}(0)_k^2 \leftarrow \mathbf{1}_d$

for $t = 0, \ldots, T-1$

$$A_{nk} \leftarrow \frac{\boldsymbol{\pi}(t)_k \mathcal{N}\left(\mathbf{z}_n; \mathbf{W}_q \boldsymbol{\mu}(t)_k, \boldsymbol{\sigma}(t)_k^2\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\mathbf{z}_n; \mathbf{W}_q \boldsymbol{\mu}(t)_j, \boldsymbol{\sigma}(t)_j^2\right)}$$

$\hat{A}_{nk} \leftarrow A_{nk} / \sum_{l=1}^{N} A_{lk}$ ▷ normalize attention

$\boldsymbol{\mu}(t+1)_k \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \mathbf{z}_n$ ▷ update slot mean

$\boldsymbol{\sigma}(t+1)_k^2 \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \left(\mathbf{z}_n - \boldsymbol{\mu}(t+1)_k\right)^2$

$\boldsymbol{\pi}(t+1)_k \leftarrow \sum_{n=1}^{N} A_{nk} / N$ ▷ update mixing coeff.

return $\boldsymbol{\mu}(T), \boldsymbol{\sigma}(T)^2$ ▷ K slot distributions

interpretation of EM. The normalised weights $\hat{A}_{nk}$ are used to compute updates for the slot means $\boldsymbol{\mu}(t+1)_k$ and variances $\boldsymbol{\sigma}(t+1)_k^2$, analogous to the M-step updates in EM. Mixing coefficients $\boldsymbol{\pi}(t+1)_k$ are updated using the average responsibilities over tokens. However, unlike standard EM, in PSA when using Algorithm 4, with different projections for mean and variance, the algorithm might not converge to the maximum likelihood solution and not guarantees monotonic improvement, PSA is trained end-to-end with a reconstruction objective, using EM like updates as inductive biases within a differentiable neural architecture, rather than exact maximum likelihood steps.

COMPARISON WITH AUTOENCODING VARIATIONAL BAYES As explained in Chapter 2, applying slot attention to a deterministic encoding $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ yields a set of K object slot representations $\mathbf{s}_{1:K} \coloneqq \mathbf{s}_1, \ldots, \mathbf{s}_K$. In combination, this process induces a stochastic encoder $q(\mathbf{s}_{1:K} \mid \mathbf{x})$, where the stochasticity comes from the random initialization of the slots in the first iteration: $\mathbf{s}_{1:K}^{t=0} \sim \mathcal{N}(\mathbf{s}_{1:K}; 0, \mathbf{I}) \in \mathbb{R}^{K \times d}$. Since each slot is a deterministic function of its previous state $\mathbf{s}^t \coloneqq f_s(\mathbf{z}, \mathbf{s}^{t-1})$ it is possible to randomly sample initial states $\mathbf{s}^0$ and obtain stochastic estimates of the slots. However, since each transition depends on $\mathbf{z}$, which in turn depends on the input $\mathbf{x}$, *we do not get a generative model we can tractably sample from*. This can be

Algorithm 4 Probabilistic Slot Attention V.2 (PSA-Proj)

Input: $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ ▷ input representation

$\mathbf{k} \leftarrow \boldsymbol{W}_k \mathbf{z} \in \mathbb{R}^{N \times d}$ ▷ compute keys

$\forall k,\ \boldsymbol{\pi}(0)_k \leftarrow 1/K,\ \boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d),\ \boldsymbol{\sigma}(0)_k^2 \leftarrow \mathbf{1}_d$

for $t = 0, \ldots, T-1$

$$A_{nk} \leftarrow \frac{\boldsymbol{\pi}(t)_k \mathcal{N}\left(\mathbf{k}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_k, \boldsymbol{\sigma}(t)_k^2\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\mathbf{k}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_j, \boldsymbol{\sigma}(t)_j^2\right)}$$

$\hat{A}_{nk} \leftarrow A_{nk} / \sum_{l=1}^{N} A_{lk}$ ▷ normalize attention

$\boldsymbol{\mu}(t+1)_k \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \boldsymbol{W}_v \mathbf{z}_n$ ▷ update slot mean

$\boldsymbol{\sigma}(t+1)_k^2 \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \left(\boldsymbol{W}_v \mathbf{z}_n - \boldsymbol{\mu}(t+1)_k\right)^2$

$\boldsymbol{\pi}(t+1)_k \leftarrow \sum_{n=1}^{N} A_{nk} / N$ ▷ update mixing coeff.

return $\boldsymbol{\mu}(T), \boldsymbol{\sigma}(T)^2$ ▷ K slot distributions

remedied by placing a tractable prior over $\mathbf{z}$ and using the VAE framework along the lines of (Wang et al., 2023b). Specifically, Wang et al. (Wang et al., 2023b) propose the Slot-VAE, which is a generative model that integrates slot attention and the VAE framework under a two-layer hierarchical latent model. However, under their formulation, there is a key challenge in calculating the KL term as the slot attention function is permutation equivariant meaning the slots have no fixed order across the posterior and prior. To compensate for this, the authors introduce a heuristic auxiliary loss and a parallel image processing path with an additional slot attention operation which is computationally costly. In contrast, our approach does not suffer from such drawbacks. This was achieved by designing the slot attention operation itself as probabilistic, and proving that having a local (per-datapoint) GMM results in the *aggregate slot posterior* and the concatenated slots being GMM distributed (Section 4.2). We then proved a new slot identifiability result using this insight (Section 4.3). Our use of the aggregate posterior is inspired by but differs substantially in both method and application from previous works on VAEs (Hoffman and Johnson, 2016; Van Den Oord, Vinyals, et al., 2017; Tomczak and Welling, 2018). Our primary goal is not to learn a better generative prior but to obtain slot identifiable representations. Lastly, since the concatenated slots are provably GMM distributed, our

model is reminiscent of a GMM-VAE (Dilokthanakul et al., 2016), but with a unique slot-based structure.

COMPUTATIONAL COMPLEXITY. PSA retains the $\mathcal{O}(TNKd)$ computational complexity of slot attention. The additional operations we introduce for calculating slot mixing coefficients and slot variances (under diagonal slot covariance structure) have complexities of $\mathcal{O}(NK)$ and $\mathcal{O}(NKd)$ respectively, which do not alter the dominant term. When PSA is combined with an additive decoder, it can *lower* computational complexity by eliminating the need to decode inactive slots. In the following, we outline a principled approach for pruning inactive slots.

AUTOMATIC RELEVANCE DETERMINATION (ARD). An open problem in slot-based modelling is the dynamic estimation of the number of slots K needed for each input (Locatello et al., 2019; Kori et al., 2024b). Probabilistic slot attention offers an elegant solution to this problem using the concept of ARD (Neal, 1996). ARD is a statistical framework that prunes irrelevant features by imposing data-driven, sparsity-inducing priors on model parameters to regularize the solution space. Since the output mixing coefficients $\boldsymbol{\pi}(T) \in \mathbb{R}^K$ are input dependent (i.e. local), irrelevant components (slots) will naturally be pruned after T attention iterations, i.e. $\boldsymbol{\pi}(T)_k \to 0$ for any unused slot k. We can either use a probability threshold $\tau \in [0, 1)$ to prune unused slots or place a Dirichlet prior over the mixing coefficients to explicitly induce sparsity. For simplicity, we take the former approach:

$$\mathbf{s}_\tau \coloneqq \left\{ \mathbf{s}(T)_k \mid k \in [K], \boldsymbol{\pi}(T)_k > \tau \right\}, \tag{4.6}$$

where $\mathbf{s}_\tau$ denotes the set of *active* slots with mixing coefficient greater than τ, and each slot is (optionally) sampled from its Gaussian distribution: $\mathbf{s}(T)_k \sim \mathcal{N}\left(\boldsymbol{\mu}(T)_k, \boldsymbol{\sigma}(T)_k^2\right)$.

AGGREGATE POSTERIOR GAUSSIAN MIXTURE. As previously explained, probabilistic slot attention goes beyond standard slot attention by introducing a per-datapoint (i.e. local) GMM to learn *distributions* over slot representations. This imposes structure over the latent space and gives us

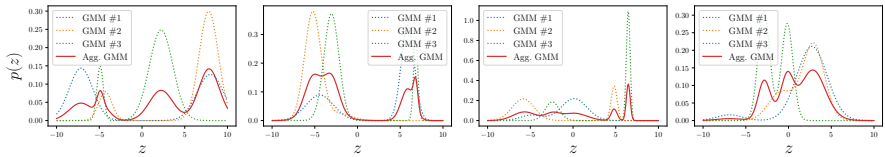


Figure 4.4: Aggregate Gaussian Mixture Density. Examples of aggregate posterior mixtures. For each plot, we compute the aggregate mixture (red line) based on three random bimodal Gaussian mixtures, and plot the respective densities. The three GMMs here are analogous to the local GMMs obtained from probabilistic slot attention (Algorithm 5), and the aggregate GMM represents $q(\mathbf{z})$.

access to *posterior* slot distributions after the attention iterations. Rather than constraining slot posteriors to be close to a tractable prior – e.g. via the VAE framework (Kingma and Welling, 2013; Wang et al., 2023a) which requires further variational approximations – we leverage our probabilistic setup to compute the optimal (global) prior over slots.

We propose to compute the *aggregate slot posterior* by marginalizing out the data: $q(\mathbf{z}) = \sum_{i=1}^{M} q(\mathbf{z} \mid \mathbf{x}_i)/M$, given a pre-trained probabilistic slot attention model (Figure 4.4). In Section 4.3, we prove that the aggregate posterior is a tractable, non-degenerate Gaussian mixture distribution which: **(i)** Serves as the theoretically optimal prior over slots; **(ii)** Is empirically stable across runs (i.e. identifiable up to an affine transformation and slot permutation, Section 4.3); **(iii)** Can be tractably sampled from and (optionally) used for slot-based scene composition tasks. Since GMMs are universal density approximators given enough components (even GMMs with diagonal covariances), the resulting aggregate posterior $q(\mathbf{z})$ is highly flexible and multimodal. It often suffices to approximate it using a sufficiently large subset of the dataset, if marginalizing out the entire dataset becomes computationally restrictive, although we did not observe this to be the case in practice in our set of experiments.

4.3 PSA: THEORETICAL ANALYSIS

In this section, we leverage the properties of the probabilistic slot attention method proposed in Section 4.2 to prove a new object-centric identifiability result. We show that object-centric representations (i.e. slots) are identifiable without supervision (up to an equivalence relation) under mixture model-

like assumptions about the latent space. This contrasts with existing work, which provides identifiability guarantees within a specific class of mixing functions, i.e. additive decoders (Lachapelle et al., 2024b). Our result unifies generative model identifiablity (Hyvarinen et al., 2019b; Khemakhem et al., 2020a; Kivva et al., 2022) and object-centric learning.

Definition 4.1 (Identifiability.). *Given an observation space $\mathcal{X}$, a probabilistic model p with parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is said to be identifiable if the mapping $\boldsymbol{\theta} \in \boldsymbol{\Theta} \mapsto p_{\boldsymbol{\theta}}(\mathbf{x})$ is injective:*

$$\left(p_{\boldsymbol{\theta}_1}(\mathbf{x}) = p_{\boldsymbol{\theta}_2}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}\right) \implies \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2. \tag{4.7}$$

Interpretation

Definition 4.1 says that if any two choices of model parameters lead to the same marginal density, they are equal. This is often referred to as strong identifiability (Hyvarinen et al., 2019b; Khemakhem et al., 2020a), and it can be too restrictive, as guaranteeing identifiability up to a simple transformation (e.g. affine) is acceptable in practice. To reflect weaker notions of identifiability, we let $\sim$ denote an equivalence relation on $\boldsymbol{\Theta}$, such that a model can be said to be identifiable up to $\sim$, or $\sim$-identifiable.

Definition 4.2 ($\sim_s$-equivalence). *Let $f_{\boldsymbol{\theta}} : \mathcal{S} \to \mathcal{X}$ denote a mapping from slot representation space $\mathcal{S}$ to image space $\mathcal{X}$ (satisfying Assumption 5.1), the equivalence relation $\sim_s$ w.r.t. to parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is defined bellow, where $\boldsymbol{P} \in \mathcal{P} \subseteq \{0,1\}^{K \times K}$ is a slot permutation matrix, $\boldsymbol{H} \in \mathbb{R}^{d \times d}$ is an affine transformation matrix, and $\mathbf{c} \in \mathbb{R}^d$:*

$$\forall \mathbf{x} \in \mathcal{X}, \quad \boldsymbol{\theta}_1 \sim_s \boldsymbol{\theta}_2 \iff \exists \boldsymbol{P}, \boldsymbol{H}, \mathbf{c} : f_{\boldsymbol{\theta}_1}^{-1}(\mathbf{x}) = \boldsymbol{P}(f_{\boldsymbol{\theta}_2}^{-1}(\mathbf{x})\boldsymbol{H} + \mathbf{c}). \tag{4.8}$$

Lemma 4.1 (Aggregate Posterior Mixture). *Given that probabilistic slot attention induces a local (per-datapoint $\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^M$) GMM with K components, the aggregate posterior $q(\mathbf{z})$ obtained by marginalizing out $\mathbf{x}$ is a non-degenerate global Gaussian mixture with MK components:*

$$q(\mathbf{z}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} \mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right). \tag{4.9}$$

Proof Sketch. The full proof is given in Appendix B.2. The result is obtained by integrating the product of the latent variable posterior density $q(\mathbf{z} \mid \mathbf{x})$ and the (local) GMM density given $\mathbf{z}$, w.r.t. $\mathbf{x}$. We then proceed by verifying that the mixing coefficients sum to one over all the components in the new mixture (Corollary B.1), proving aggregated posterior to be a well-defined probability distribution, this can be empirically confirmed in Figure 4.4. Next, we use $q(\mathbf{z})$ in our identifiablity result.

Theorem 4.1 (Mixture Distribution of Concatenated Slots)**.** *Let f_s denote a permutation equivariant PSA function such that $f_s(\mathbf{z}, P\mathbf{s}^t) = Pf_s(\mathbf{z}, \mathbf{s}^t)$, where $P \in \{0,1\}^{K\times K}$ is an arbitrary permutation matrix. Let $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_K) \in \mathbb{R}^{Kd}$ be a random variable defined as the concatenation of K individual slots, where each slot is Gaussian distributed within a K-component mixture: $\mathbf{s}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall k \in \{1, \dots K\}$. Then, $\mathbf{s}$ is also GMM distributed with $K!$ mixture components:*

$$p(\mathbf{s}) = \sum_{p=1}^{K!} \boldsymbol{\pi}_p \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p), \qquad \text{where}$$
$$\boldsymbol{\pi} \in \Delta^{K!-1}, \quad \boldsymbol{\mu}_p \in \mathbb{R}^{Kd}, \quad \boldsymbol{\Sigma}_p \in \mathbb{R}^{Kd\times Kd}. \tag{4.10}$$

Proof Sketch. The proof is in Appendix B.2. We observe that the permutation equivariance of the PSA function f_s induces $K!$ ways of concatenating sampled slots $\mathbf{s}_k$, where each permutation maps to a different mode with block diagonal covariance structure in a GMM living in $\mathbb{R}^{Kd}$ (e.g. Figure 4.7, B.1).

Theorem 4.2 ($\sim_s$-Identifiable Slot Representations)**.** *Given that the aggregate posterior $q(\mathbf{z})$ is an optimal, non-degenerate mixture prior over slot space (Lemma 4.2), $f : \mathcal{S} \to \mathcal{X}$ is a piecewise affine weakly injective mixing function (Assumption 5.1), and the slot representation, $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_K)$ can be observed as a sample from a GMM (Theorem C.1), then $p(\mathbf{s})$ is identifiable as per Definition 4.2.*

Proof Sketch. The proof is given in Appendix B.4. Lemma 4.2 and Corollary B.1 state that the optimal latent variable prior in our case is GMM

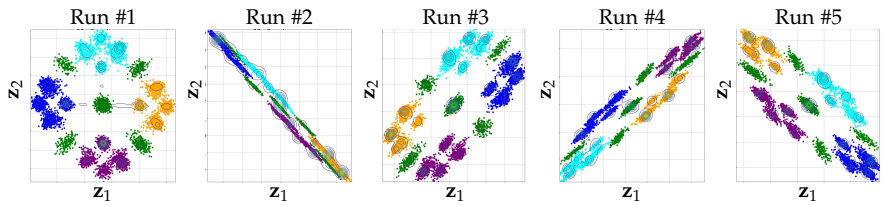


Figure 4.5: Aggregate posterior identifiability. Recovered (latent) aggregate posteriors $q(\mathbf{z})$ across 5 runs of our PSA model. As detailed in Section 4.4, we used a 2D synthetic dataset with 5 total 'object' clusters, with each observation containing at most 3. This provides strong evidence of recovery of the latent space up to affine transformations, empirically verifying our identifiability claim.

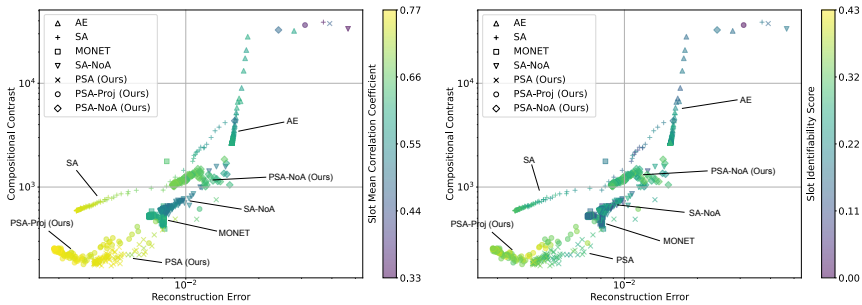


Figure 4.6: Experiments comparing slot-identifiability scores. Colour coding represents the level of slot identifiability achieved by each model as measured by the SMCC (left) and SIS (right). SMCC offers a more consistent metric correlating with reconstruction loss.

distributed, non-degenerate and equates to the aggregate posterior $q(\mathbf{z})$. This permits us to extend (Kivva et al., 2022)'s result to show that if $q(\mathbf{z})$ is distributed according to a non-degenerate GMM and the mixing function f_d is piecewise affine and weakly injective, then the *slot* distribution representation, $p(\mathbf{s})$ which is also a GMM (Theorem C.1) is identifiable up to an affine transformation and arbitrary slot permutation.

Corollary 4.1 (Individual Slot Identifiability)**.** *If the distribution over concatenated slots $p(\mathbf{s})$, where $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_K) \in \mathbb{R}^{Kd}$, is $\sim_s$-identifiable (Theorem 4.2) then this implies $q(\mathbf{z})$ is identifiable up to an affine transformation and permutation of the slots $\mathbf{s}_k$. Therefore, each slot distribution $\mathbf{s}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall k \in \{1, \dots K\}$ is also identifiable up to an affine transformation.*

Given that the focus of this work is theoretical, the primary goal of our experiments is to provide strong empirical evidence of our main identifiability result (ref. Figures 4.5, 4.6). With that said, we also extend our experimental study to popular imaging benchmarks to demonstrate that our method scales to higher-dimensional settings (ref. Tables 4.3, 4.2).

DATASETS & EVALUATION METRICS. Our experimental analysis involves standard benchmark datasets from object-centric learning literature including SPRITEWORLD (Brady et al., 2023), CLEVR (Johnson et al., 2017), and OBJECTSROOM (Kabra et al., 2019). We report the foreground-adjusted rand index (FG-ARI) and FID (Heusel et al., 2017) to quantify both object-level *binding* capabilities and image quality. Our main goal is to measure slot-identifiability, so we use the slot identifiability score (Brady et al., 2023) and the mean correlation coefficient (MCC) across slot representations – we call the latter slot-MCC (SMCC). For two sets of slots $\{\mathbf{s}_i\}_{i=1}^M$, and $\{\tilde{\mathbf{s}}_i\}_{i=1}^M$, where $\mathbf{s}_i \in \mathbb{R}^{K\times d}$, $\tilde{\mathbf{s}}_i \in \mathbb{R}^{K\times d}$, extracted from M images $\{\mathbf{x}_i\}_{i=1}^M$, the SMCC between any $\mathbf{s}$ and $\tilde{\mathbf{s}}$ is obtained by matching the slot representations and their order. The order is matched by mapping slots in $\tilde{\mathbf{s}}$ w.r.t $\mathbf{s}$ assigned by π, followed by a learned affine mapping $\boldsymbol{A}$ between aligned $\tilde{\mathbf{s}}_{\pi(i)}$ and $\mathbf{s}$:

$$\text{SMCC}(\mathbf{s}, \tilde{\mathbf{s}}) := \frac{1}{K \times d} \sum_{i=0}^{K} \sum_{j=0}^{d} \rho(\mathbf{s}_{ij}, \boldsymbol{A}\tilde{\mathbf{s}}_{\pi(i)j}). \tag{4.11}$$

For more details on the metrics please refer to Appendix B.5.

MODELS & BASELINES. We consider three ablations on our proposed probabilistic slot attention (PSA) method: (i) PSA base model (Algorithm 3); (ii) PSA-PROJ model (Algorithm 5); and (iii) PSA-NOA model, which is equivalent to PSA-PROJ but without an additive decoder. We experiment with two types of decoders: (i) an additive decoder similar to (Watters et al., 2019b)'s spatial broadcasting model; and (ii) standard convolutional decoder. In all cases, we use LeakyReLU activations to satisfy the weak injectivity conditions (Assumption 5.1). In terms of object-centric learning baselines,

we compare with standard additive autoencoder setups following (Brady et al., 2023), SA (Locatello et al., 2020b), and MONET (Burgess et al., 2019).

CASE STUDY: IMAGING DATA. Although our focus is primarily theoretical, we now demonstrate that our method generalizes/scales to higher-dimensional imaging modalities. To that end, we first use the SPRITEWORLD (Watters et al., 2019a) dataset to evaluate the SMCC and SIS w.r.t. ground truth latent variables. Figure 4.6 presents our identifiability results against the baselines. Similar to (Brady et al., 2023), we observe higher SIS when compositional contrast and reconstruction error decreases. However, when the mixing function is *not* additive, the compositional contrast does not decrease drastically while maintaining higher SMCC and SIS – this verifies our identifiability claim using only piecewise affine decoders. As shown in Figure 4.6, we also observe that PSA routing with additive decoder models achieves lower compositional contrast and reconstruction errors when compared with other methods. This indicates that stronger identifiability of slot representations is achievable when combining both slot latent structure and inductive biases in the mixing function. Unlike for the SPRITEWORLD dataset, all the ground truth generative factors are *unobserved* for the CLEVR and OBJECTSROOM datasets we use. Therefore, for evaluation in these cases, we train multiple models with different seeds and compute the SMCC and SIS measures across different runs. This is similar to our earlier synthetic experiment and standard practice in identifiability literature. Table 4.3 presents our main identifiability results on CLEVR and OBJECTSROOM. We observe similar trends in favour of our proposed PSA method as measured by both SMCC and (slot averaged) R2 score relative to the baselines.

Table 4.2: Pascal VOC2012 benchmark results using probabilistic slot attention. All baselines are standard results from (Löwe et al., 2024). SA MLP (w/ DINO) denotes our replication of the DINOSAUR MLP baseline from (Seitzer et al., 2022), whereas (‡) denotes the use of slot attention masks rather than decoder alpha masks for evaluation.

MODEL	$\text{MBO}_i\uparrow$	$\text{MBO}_c\uparrow$
BLOCK MASKS	$0.247 \pm .000$	$0.259 \pm .000$
SA	$0.222 \pm .008$	$0.237 \pm .008$
SLATE	$0.310 \pm .004$	$0.324 \pm .004$
ROTATING FEATURES	$0.282 \pm .006$	$0.320 \pm .006$
DINO K-MEANS	$0.363 \pm .000$	$0.405 \pm .000$
DINO CAE	$0.329 \pm .009$	$0.374 \pm .010$
DINOSAUR MLP	$0.395 \pm .000$	$0.409 \pm .000$
Ours:		
SA MLP (w/ DINO)	$0.384 \pm .000$	$0.397 \pm .000$
SA MLP (w/ DINO)‡	$0.400 \pm .000$	$0.415 \pm .000$
PSA MLP (w/ DINO)	$0.389 \pm .009$	$0.422 \pm .009$
PSA MLP (w/ DINO)‡	$\mathbf{0.405} \pm .010$	$\mathbf{0.436} \pm .011$

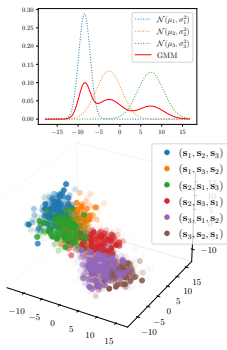


Table 4.3: Comparing slot identifiability scores (SMCC and slot averaged R2) with existing object-centric learning methods.

Method	CLEVR		Objects-Room	
	SMCC ↑	R2 ↑	SMCC ↑	R2 ↑
AE	$0.43 \pm .02$	$0.26 \pm .02$	$0.46 \pm .05$	$0.45 \pm .06$
MONET	$0.32 \pm .01$	$0.39 \pm .09$	$0.43 \pm .04$	$0.41 \pm .10$
SA	$0.56 \pm .02$	$0.55 \pm .05$	$0.66 \pm .01$	$0.54 \pm .00$
SA-NoA	$0.23 \pm .03$	$0.24 \pm .02$	$0.48 \pm .02$	$0.47 \pm .01$
Ours:				
PSA-NoA	$0.56 \pm .05$	$0.42 \pm .06$	$0.44 \pm .03$	$0.45 \pm .05$
PSA	$0.58 \pm .06$	$0.48 \pm .02$	$0.66 \pm .01$	$\mathbf{0.64 \pm .02}$
PSA-Proj	$\mathbf{0.66 \pm .06}$	$\mathbf{0.62 \pm .08}$	$\mathbf{0.71 \pm .00}$	$0.62 \pm .02$

Figure 4.7: Concatenated Slot Gaussian Mixture. Example of the higher-dim GMM (1D → 3D) induced by the permutation equivariance of PSA and the $K!$ ways of concatenating sampled slots.

Verifying Slot Identifiability: Gaussian Mixture of Objects.

To provide conclusive empirical evidence of our identifiability claim (Theorem 4.2), we set up a synthetic modelling scenario under which it is possible to visualize the aggregate posterior across different runs. The goal is to show that PSA is $\sim_s$-identifiable (Theorem 4.2) in the sense that it can recover the same latent space distribution up to an affine transformation and slot order permutation. For the data generating process, we defined a K=5 component GMM, with differing mean parameters $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_5\}$, and shared isotropic covariances. The 5 components emulate 5 different object types in a given environment. To create a single data point, we randomly chose 3 of the 5 components and sampled 128 points uniformly at random from each mode. Figure B.3 shows some data samples, where different colours correspond to different objects. We used 1000 data points in total for training our PSA model. As shown in Figure 4.5, the aggregate posterior is either rotated, translated, skewed, or flipped across different runs as predicted by our theory – this contrasts with all baselines wherein the aggregate posterior is intractable. We observed an SMCC of 0.93 ± 0.04, and R2-score of 0.50 ± 0.08.

Case study: complex decoder structure. To empirically test slot identifiability using more complex non-additive decoders, we used transformer decoders following SLATE (Singh et al., 2021), and simply replaced the slot attention module with probabilistic slot attention. On the

CLEVR dataset, we observed a significantly improved SMCC of $\mathbf{0.73} \pm .01$ and R2 of $\mathbf{0.55} \pm .06$ relative to Table 4.3. To demonstrate that probabilistic slot attention can scale to large-scale real-world data we ran additional experiments on the Pascal VOC2012 (Li et al., 2018) dataset, following the exact DINOSAUR strategies and setups described in (Seitzer et al., 2022; Kakogeorgiou et al., 2024) for fairness, then simply swapping out the slot attention module with probabilistic slot attention. As shown in Table 4.2, we find that probabilistic slot attention is competitive with standard slot attention on real-world data. We also tested more complex, *non-additive* decoders based on autoregressive transformers, following the DINOSAUR (Seitzer et al., 2022) setup. For our PSA Transformer (w/ DINO) model, we observed an mBO_i of $\mathbf{0.447}$, and mBO_c of $\mathbf{0.521}$ which is competitive with the DINOSAUR Transformer baseline (Seitzer et al., 2022) of 0.44 and 0.512 respectively. In this case, we found that a lower maximum learning rate of 10^{-4} was beneficial for stabilizing PSA training.

In summary, our experiments corroborate our theoretical results and suggest why *non-additive* decoder structures can still work well given the appropriate latent structure and inference procedures are in place. With that said, there is a trade-off between identifiability and expressivity induced by the choice of decoder structure (Lachapelle et al., 2024b), so depending on the use case, it may indeed be advantageous to combine both latent and additive decoder structures in practice. We will use these findings to develop VISA mainly to address the problem of spatial ambiguities a well known problem in scene understanding and object composition.

4.5 View Invariant Slot Attention

Let $\mathbf{x}^{1:V} = \{\mathbf{x}^1, \dots \mathbf{x}^V\} \in \mathcal{X} = \mathcal{X}^1 \times \cdots \times \mathcal{X}^V$, V views of the same scene observed from different viewpoints with an observational space $\mathcal{X} \subseteq \mathbb{R}^{V \times H \times W \times C}$. We consider $[V]$ as a shorthand notation for $\{1, \dots, V\}$. Let $\mathcal{O}^e = \mathcal{O}^1 \cup \cdots \cup \mathcal{O}^V$ correspond to an abstract notion of object sets of an environment, while $\mathcal{O}^v, \forall v \in [V]$ is a set of objects present in a considered viewpoint v. Importantly, we consider that the number of

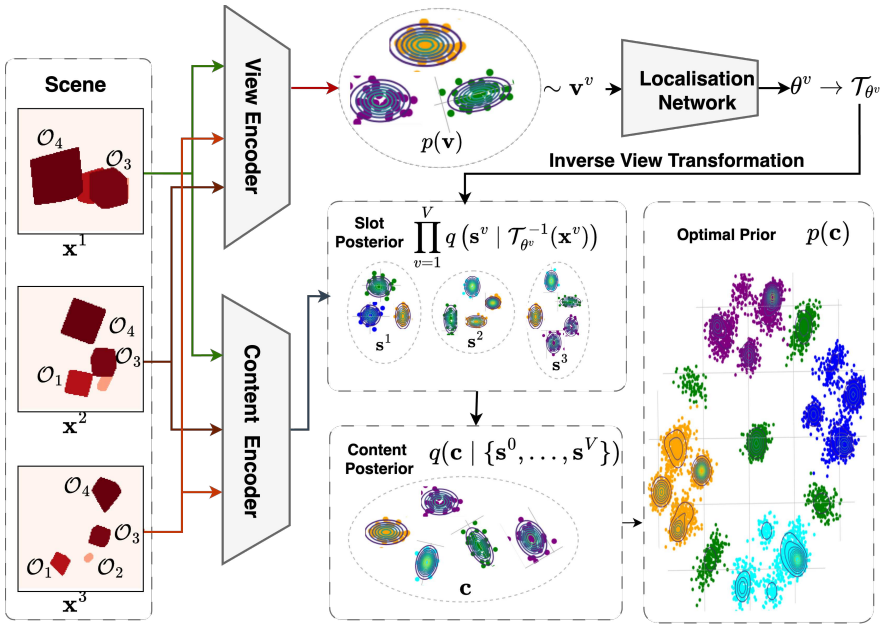


Figure 4.8: Overview-VISA: The figure illustrates a scene with four objects $\mathcal{O}^s = \{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4\}$, observed from three different viewpoints, each described with a set of clearly visible objects: $\mathcal{O}^1 = \{\mathcal{O}_3, \mathcal{O}_4\}, \mathcal{O}^2 = \{\mathcal{O}_1, \mathcal{O}_3, \mathcal{O}_4\}, \mathcal{O}^3 = \{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4\}$. The corresponding images are passed through view and content encoders, and sampled global view vector $\mathbf{v}$ is used to estimate transformation function $\mathcal{T}_{\theta^v}$ given by parameters θ^v predicted using a localisation network. We apply a view-specific inverse $\mathcal{T}_{\theta^v}^{-1}$ on respective images projecting them to an *implicit* space, which is used to learn view conditioned slot posterior corresponding to GMMs represented by $q(\mathbf{s}^v \mid \mathcal{T}_{\theta^v}^{-1}(\mathbf{x}^v))$, which are further aggregated to marginalize viewpoint information, resulting in a content posterior, also a GMM $q(\mathbf{c} \mid \{\mathbf{s}^1, \ldots, \mathbf{s}^V\})$, which is further accumulated across all samples resulting in optimal prior $p(\mathbf{c})$.

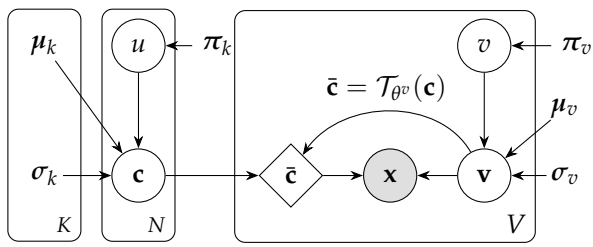


Figure 4.9: Graphical model for view invariant slot attention: For every image in a dataset a view $\mathbf{v} \in \mathbb{R}^{d_v} \sim p(\mathbf{v})$, this view is used to compute transformation $\mathcal{T}_{\theta^v}$. Similarly, desired number ($< K$) of content representations $\mathbf{c} \in \mathbb{R}^{N \times d_s}$ are sampled content distribution $p(\mathbf{c})$. Finally, the image $\mathbf{x}$ is generated using the transformed content $\mathcal{T}_{\theta^v}(\mathbf{c})$ and view $\mathbf{v}$.

objects per viewpoint can vary, *i.e.*, $|\mathcal{O}^1 \cup \cdots \cup \mathcal{O}^V| \geq |\mathcal{O}^v| \ \forall \ v \in [V]$, allowing for partial or full occlusion in some viewpoints. Let $\mathbf{v}^{1:V} \in \mathcal{V} = \mathcal{V}^1 \times \cdots \times \mathcal{V}^V \subseteq \mathbb{R}^{V \times d_v}$ be inferred viewpoint-specific information[3], while $\mathbf{s}_{1:K}^{1:V} \in \mathcal{S} = \mathcal{S}^1 \times \cdots \times \mathcal{S}^V \subseteq \mathbb{R}^{V \times K \times d_s}$ correspond to a viewpoint-specific slot representation. Let $\mathbf{c}_{1:K} \in \mathcal{C} \subseteq \mathbb{R}^{K \times d_c}$ capture the notion of an *aggregate*, effectively accumulating the object knowledge across viewpoints. For any subset A of $[V]$, we represent scene observations as $\mathbf{x}^A = \{\mathbf{x}^i : \forall i \in A\} \in \times_{i \in A} \mathcal{X}^i$. The inferred viewpoints and the view specific slots are denoted as $\mathbf{v}^A = \{\mathbf{v}^i : \forall i \in A\} \in \times_{i \in A} \mathcal{V}^i$, and $\mathbf{s}_{1:K}^A = \{\mathbf{s}_{1:K}^i : \forall i \in A\} \in \times_{i \in A} \mathcal{S}^i$, respectively. We define $p_A(\mathbf{c})$ as the distribution of $\mathbf{c}$ over A. A more comprehensive summary of notations and terminologies is provided in Appendix C.1.

In modelling, `w.l.o.g`, we consider access to a certain subset $A \subseteq [V]$, ensuring the model's applicability across different scenarios. Furthermore, to simplify notation, we sometimes do not include the superscript denoting the full set of views, thereby using $\mathbf{x} = \mathbf{x}^A$, $\mathbf{s}_{1:K} = \mathbf{s}_{1:K}^A$, and $\mathbf{v} = \mathbf{v}^A$ interchangeably. Likewise, if we do not specify the subscripts for $\mathbf{c}$ and $\mathbf{s}$, it implies they represent the entire collection of objects, specifically as $\mathbf{s} = \mathbf{s}_{1:K}^A$ and $\mathbf{c} = \mathbf{c}_{1:K}$. Lastly, for any function f that operates on two distinct inputs $\mathbf{x} = f(\mathbf{z}, \mathbf{v})$, its inverse is denoted by $\mathbf{z} = f^{-1}(\mathbf{x}; \mathbf{v})$, which signifies the reversal of f conditioned on a variable $\mathbf{v}$. In the rest of this section, we introduce all the components involved in our model. We also introduce assumptions, examples, and intuition wherever necessary. Considering the generative model Eqn. 4.12, which is overviewed in graphical model Figure 5.3, any scene $\mathbf{x}$ is generated using view $\mathbf{v}$ and content $\mathbf{c}$. Here, both $\mathbf{c}$ and $\mathbf{v}$ are latent variables learned with variational inference (Kingma and Welling, 2013).

$$p(\mathbf{x}) = \iint p(\mathbf{x}^v \mid \mathcal{T}_{\theta^v}(\mathbf{c}), \mathbf{v}^v) \ p(\mathbf{c}) \ p(\mathbf{v}^v) \ d\mathbf{v}^v \ d\mathbf{c} \tag{4.12}$$

VIEW MODEL. Given that the view property remains consistent across all objects, we treat the view as a global, image-level property as op-

3 We abuse the terminology by considering viewpoint, lighting, object dimension, to be encoded in a representation $\mathbf{v}$. Note that the $\mathbf{v}$ is inferred by the model.

posed to Yuan et al. (2024), where view is treated as an object-level property. Assuming access to a discrete set of viewpoints denoted by A, we consider prior over a view distribution to be a GMM represented by $p(\mathbf{v}) = \sum_{v=1}^{|A|} \boldsymbol{\pi}^v \mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_v, \boldsymbol{\sigma}_v^2)$. To learn the parameters of this GMM, we consider the posterior of the form $q_\phi(\mathbf{v} \mid \mathbf{x}^v) \;\forall\; v \in A$[4]. In both prior and posterior, we consider the covariance to be diagonal, implicitly making an ICA assumption (Khemakhem et al., 2020a). The sampled variable $\mathbf{v} \sim q_\theta(\mathbf{v} \mid \mathbf{x}^v)$ is used to estimate transformation parameters $\theta^v \in \mathbb{R}^{3\times 2}$ as in Jaderberg et al. (2015) which makes an affine transformation map $\mathcal{T}_{\theta^v}$, which is later applied on content $\mathbf{c}$ and on view-specific slots $\mathbf{s}$. It is important to note that we use the same set of parameters ϕ across all viewpoints in A for inferring view information $\mathbf{v}$.

VIEWPOINT SPECIFIC SLOTS. As illustrated in Figure 4.8 the inference of $\mathbf{c}$ depends on the view-specific slots $\mathbf{s}$. For a considered image $\mathbf{x}^v, v \in A$, we first apply an inverse view transformation $\mathcal{T}_{\theta^v}^{-1}$ and model the slot distribution as a spatial mixture model represented by $q(\mathbf{s}_{1:K}^A \mid \mathcal{T}_{\theta^v}^{-1}(\mathbf{x}^A))$. The inverse transformation makes sure that the estimated object representations across all view in A are in a common implicit representation space. As this is an intermediate variable which does not show up in our generative model in Eqn. 4.12, we update the corresponding parameters with closed-form equations via expectation maximisation algorithm as in Kori et al., 2024c. The resulting slot posterior is a conditional GMM as described in Eqn. 4.13, where $\bar{\mathbf{x}}^v = \mathcal{T}_{\theta^v}^{-1}(\mathbf{x}^v)$ is a transformed inputs, $(\mu_k(\bar{\mathbf{x}}^v), \sigma_k^2(\bar{\mathbf{x}}^v), \pi_k(\bar{\mathbf{x}}^v))$ are mean, diagonal covariance, and mixing coefficients for the considered a view and object.

$$q(\mathbf{s}^v \mid \bar{\mathbf{x}}^v) = \sum_{k=1}^{K} \boldsymbol{\pi}_k(\mathbf{x}^v) \mathcal{N}\Big(\mathbf{s}_k^v; \boldsymbol{\mu}_k(\bar{\mathbf{x}}^v), \boldsymbol{\sigma}_k^2(\bar{\mathbf{x}}^v)\Big) \tag{4.13}$$

REPRESENTATION MATCHING. Given the permutation equivariance property of slot representations, we use a matching function with a permutation matrix $\boldsymbol{P}^v$, $m_s : \mathcal{S}^A \to \mathcal{S}^A$ such that $m_s(\mathbf{s}_{1:K}^v) = \boldsymbol{P}^v \mathbf{s}_{1:K}^v$ mapping representation axis `w.r.t` $\boldsymbol{P}^v$. The permutation matrix $\boldsymbol{P}^v$ is estimated by considering the slots of the first viewpoint $\mathbf{s}^1$ as a base representation, and

4 We consider the parametric form of q to be Gaussian.

other representations $\mathbf{s}^v \, \forall v \in A$ are matched to align with it. We utilise Hungarian matching, as illustrated in Locatello et al., 2020b; Wang et al., 2023a, to estimate this permutation matrix $\boldsymbol{P}^v$, to control the noise in the matching algorithm, we introduce view-warmup strategy, which we detail in Appendix C.5.5.

CONTENT AGGREGATOR. We consider $g : \mathcal{S} \to \mathcal{C}$ as a content aggregator function, which marginalises the effect of view conditioning. To achieve this, we consider a convex combination of all the aligned slot representations (aligned to a base representation), considering mixing coefficients $\boldsymbol{\pi}_k(\mathbf{x}^v)$ (we use $\boldsymbol{\pi}_k^v$ for simplicity) in Eqn. 4.13 as a combination weight. The convex combination accounts for potential object occlusions, which may cause objects to be absent in particular views ensuring only active representations are combined (refer to an intuition below), resulting in a content posterior ($q(\mathbf{c} \mid \mathbf{s})$), which is a GMM with mixing coefficients $\tilde{\boldsymbol{\pi}}_k = \left(\sum_{v=1}^{|A|} \boldsymbol{\pi}_k^v\right) / |A|$ and the parameters described in Eqn. 4.15 (`w.l.o.g` we consider $\mathbf{s}, \boldsymbol{\pi}$ to represent aligned representations), refer to Lemma C.3, with $w_i = 1/|A| \, \forall i \in A$. Additionally, algorithm 5 details the entire forward process.

Intuition: Content aggregation

Based on illustrated example in Figure 4.8, for images $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$, the resulting matched slots and mixing coefficients correspond to $\mathbf{s}^1 = \{\mathbf{s}_r^1, \mathbf{s}_r^1, \mathbf{s}_{\mathcal{O}_3}^1, \mathbf{s}_{\mathcal{O}_4}^1, \mathbf{s}_b^1\}, \mathbf{s}^2 = \{\mathbf{s}_{\mathcal{O}_1}^2, \mathbf{s}_r^2, \mathbf{s}_{\mathcal{O}_3}^2, \mathbf{s}_{\mathcal{O}_4}^2, \mathbf{s}_b^2\}, \mathbf{s}^3 = \{\mathbf{s}_{\mathcal{O}_1}^3, \mathbf{s}_{\mathcal{O}_2}^3, \mathbf{s}_{\mathcal{O}_3}^3, \mathbf{s}_{\mathcal{O}_4}^3, \mathbf{s}_b^3\}$, where $\mathbf{s}_{\mathcal{O}_i}^v, \mathbf{s}_r^v$, and $\mathbf{s}_b^v$ correspond to slot representation for object $\mathcal{O}_i$, random slot representation and background information, respectively, with mixing coefficients $\boldsymbol{\pi}^1 = \{0, 0, 1, 1, 1\}, \boldsymbol{\pi}^2 = \{1, 0, 1, 1, 1\}$, and $\boldsymbol{\pi}^1 = \{1, 1, 1, 1, 1\}$. Proposed aggregation merges the slots ignoring the random slots $\mathbf{s}_r^v$, resulting in $\mathbf{c}_{\mathcal{O}_1} = (\mathbf{s}_{\mathcal{O}_1}^2 + \mathbf{s}_{\mathcal{O}_1}^3)/2, \mathbf{c}_{\mathcal{O}_2} = \mathbf{s}_{\mathcal{O}_2}^3$ and so on.

$$g(\mathbf{s}_{1:K}^{1:V}, \boldsymbol{\pi}^{1:V}) = \sum_{v=1}^{|A|} \frac{\boldsymbol{\pi}_{1:k}^{v}}{|A|\tilde{\boldsymbol{\pi}}_k^v} \mathbf{s}_{1:K}^{v}; \tag{4.14}$$

$$\tilde{\boldsymbol{\mu}}_k(\mathbf{x}^A) = \sum_{v=1}^{|A|} \frac{\boldsymbol{\pi}_k^v}{|A|\tilde{\boldsymbol{\pi}}_k^v} \boldsymbol{\mu}_k(\mathbf{x}^v);$$

$$\tilde{\sigma}^2(\mathbf{x}^A) = \sum_{v=1}^{|A|} \left(\frac{\boldsymbol{\pi}_k^v}{|A|\tilde{\boldsymbol{\pi}}_k^v} \right)^2 \sigma_k^2(\mathbf{x}^v); \tag{4.15}$$

MIXING FUNCTION AND TRAINING OBJECTIVE. We consider both *additive and non-additive* (ref. definition C.1) mixing functions $f_d : \mathcal{C} \times \mathcal{V}^v \to \mathcal{X}^v$. For additive decoders, we use a spatial-broadcasting (Greff et al., 2019) and MLP decoders, and for non-additive mixing function, we use auto-regressive transformers (Vaswani et al., 2017). We use the shared decoder f_d for all views and objects, modelling the conditional distribution $p(\mathbf{x}^v \mid \mathcal{T}_{\theta^v}(\mathbf{c}), \mathbf{v}^v)$.

To train our model in an end-to-end fashion, we maximise the log-likelihood of the joint $p(\mathbf{x}^A)$, which results in the evidence lower bound (ELBO), Eqn. 4.16, check Lemma C.1. Here, we consider the distribution form of $p(\mathbf{x}^v \mid \mathbf{c}, \mathbf{v}^v)$ to be Gaussian with learnable mean with isotropic covariance.

$$\mathbb{E} \log p(\mathbf{x} \mid \mathcal{T}_{\theta^v}(\mathbf{c}), \mathbf{v}) - \mathrm{KL}\left(q(\mathbf{v} \mid \mathbf{x}) \parallel p(\mathbf{v})\right) \tag{4.16}$$

4.6 VISA: THEORETICAL ANALYSIS

In this section, we leverage the properties of the proposed model to theoretically demonstrate the learning of identifiable representations under challenging spatial ambiguities. In this work, we consider our data-generating process to satisfy a viewpoint sufficiency assumption (refer to 4.1).

Assumption 4.1 (View-point sufficiency). *For any set $A \subseteq [V]$, we consider set A to be view-point sufficient iff $|\mathcal{O}^A| = |\mathcal{O}^e|$. This basically means that all the*

Algorithm 5 View Invariant Slot Attention VISA

Input: $A \in [V]$, $\mathbf{z}^A = \{f_e(\mathbf{x}^v)\ \forall v \in A\} \in \mathbb{R}^{|A| \times N \times d}$ ▷ input
View: $\mathbf{v}^A = \{\mathbf{v}^v \sim \mathcal{N}(\mathbf{v}^v; \boldsymbol{\mu}(\mathbf{z}^v), \boldsymbol{\sigma}^2(\mathbf{z}^v))\ \forall v \in A\} \in \mathbb{R}^{|A| \times d}$ ▷ view rep.
View Transformation: $\theta^A = \{\theta^v = \text{Localisation}(\mathbf{v}^v)\ \forall v \in A\} \in \mathbb{R}^{|A| \times 2 \times 3}$
  $\text{key}^A \leftarrow \boldsymbol{W}_k \mathcal{T}_{\theta^v}^{-1}(\mathbf{z}^A) \in \mathbb{R}^{|A| \times N \times d}$,
  $\text{value}^A \leftarrow \boldsymbol{W}_v \mathcal{T}_{\theta^v}^{-1}(\mathbf{z}^A) \in \mathbb{R}^{|A| \times N \times d}$ ▷ optional value := key
$\mathbf{s} \leftarrow \emptyset$; $\hat{\boldsymbol{\pi}} \leftarrow \emptyset$
for $v \in A$ **do**
  $\forall k$, $\boldsymbol{\pi}(0)_k \leftarrow 1/K$, $\boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d)$, $\boldsymbol{\sigma}(0)_k^2 \leftarrow \mathbf{1}_d$
  for $t = 0 \rightarrow T - 1$ **do**
    $A_{nk} \leftarrow \frac{\boldsymbol{\pi}(t)_k \mathcal{N}\left(\text{key}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_k, \boldsymbol{\sigma}(t)_k^2\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\text{key}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_j, \boldsymbol{\sigma}(t)_j^2\right)}$ ▷ compute attention
    $\hat{A}_{nk} \leftarrow \frac{A_{nk}}{\sum_{l=1}^{N} A_{lk}}$ ▷ normalize attention
    $\boldsymbol{\mu}(t+1)_k \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \cdot \text{value}_n$ ▷ update slot mean
    $\boldsymbol{\sigma}(t+1)_k^2 \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \cdot (\text{value}_n - \boldsymbol{\mu}(t+1)_k)^2$ ▷ update slot variance
    $\boldsymbol{\pi}(t+1)_k \leftarrow \frac{1}{N} \sum_{n=1}^{N} A_{nk}$ ▷ update mixing coefficient
  end for
  $\mathbf{s} \leftarrow \mathbf{s} \cup \{(\boldsymbol{\mu}_{1:K}(T), \boldsymbol{\sigma}_{1:K}^2(T))\}$; $\hat{\boldsymbol{\pi}} \leftarrow \hat{\boldsymbol{\pi}} \cup \{\boldsymbol{\pi}_{1:K}(T)\}$ ▷ slot collection
end for
return $\mathbf{c} = \text{ConvexCombination}(\mathbf{s}, \hat{\boldsymbol{\pi}})$ ▷ K view invariant content

objects are visible across all the considered views A, even when an individual view may not contain all the objects.

Example: Viewpoint-sufficiency

Example 4.1. *Based on illustrated example in Figure 4.8, the scene is composition of four objects $\mathcal{O}^e = \{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4\}$, view point subset $A = [V] = \{1, 2, 3\}$ is considered to be view point sufficient since $\bigcup_{v \in A} \mathcal{O}^v = \{\mathcal{O}_3, \mathcal{O}_4\} \cup \{\mathcal{O}_1, \mathcal{O}_3, \mathcal{O}_4\} \cup \{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4\} = \mathcal{O}^e$.*

Given that we learn the parameters of our view-specific spatial GMM with closed-form updates, we do not use an explicit prior minimising KL divergence. Instead, we rely on the fact that marginalising the effect of data points from posterior (*aggregate posterior*) is an optimal prior (Hoffman and Johnson, 2016; Kori et al., 2024c), resulting in $p(\mathbf{c}) = \int\int q(\mathbf{c}|\mathbf{s}^A, \mathbf{x}^A) d\mathbf{s}^A d\mathbf{x}^A$. Given that GMMs are universal density approximates given enough com-

ponents (even GMMs with diagonal covariances), the resulting aggregate posterior $q(\mathbf{c}) = p(\mathbf{c})$ is highly flexible and multi-modal. It often suffices to approximate it using a sufficiently large subset of the dataset if marginalising out the entire dataset becomes computationally restrictive.

Lemma 4.2 (Optimal Prior)**.** *The proof is detailed in Appendix C.4. For $A \in [V]$, given the a local content distribution $q(\mathbf{c}_{1:K} \mid \mathbf{s}^A_{1:K}, \mathbf{x}^A)$ (per-scene $\mathbf{x}^A \in \{\mathbf{x}^A_i\}_{i=1}^M$), which can be expressed as a GMM with K components, the aggregate posterior $q(\mathbf{c})$ is obtained by marginalizing out $\mathbf{x}, \mathbf{s}$ is a non-degenerate global Gaussian mixture with MK components:*

$$p(\mathbf{c}) = q(\mathbf{c}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} \mathcal{N}\left(\mathbf{c}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\sigma}^2_{ik}\right). \tag{4.17}$$

Proof Sketch. The result is obtained by integrating the product of involved posterior densities $q(\mathbf{c} \mid \mathbf{s}) q(\mathbf{s} \mid \mathbf{x}) p(\mathbf{x})$. Further, we verify if the mixing coefficients sum to one in the new mixture, proving the aggregate to be well-defined. □

With this, we show three main results: firstly, we show that aggregate content representations ($\mathbf{c}$) are identifiable without supervision (up to $\sim_s$). Secondly, we show that these representations are invariant to the choice of viewpoints under assumption 4.1. Finally, we show that the model exhibits in an approximate view equivariance.

Theorem 4.3 (Affine Equivalence)**.** *For any subset $A \subseteq [V]$, such that $|A| > 0$, given a set of images $\mathbf{x}^A \in \mathcal{X}^A$ and a corresponding aggregate content $\mathbf{c} \in \mathcal{C}$ and a non-degenerate content posterior $q(\mathbf{c} \mid \mathbf{s}^A)$, considering two mixing function $f_d, \tilde{f}_d$ satisfying assumption 5.1, with a shared image, then $\mathbf{c}$ are identifiable up to $\sim_s$ equivalence.*

Intuition: Affine equivalence

Considering an example 4.1, with two perfectly trained models f_d and $\tilde{f}_d$. Resulting aggregate contents are described as $\mathbf{c} = f_d^{-1}(\mathbf{x}^A; \mathbf{v}^A) = \{\mathbf{c}_{\mathcal{O}_1}, \mathbf{c}_{\mathcal{O}_2}, \mathbf{c}_{\mathcal{O}_3}, \mathbf{c}_{\mathcal{O}_4}, \mathbf{c}_{\mathcal{O}_b}\}$ and $\tilde{\mathbf{c}} = \tilde{f}_d^{-1}(\mathbf{x}^A; \mathbf{v}^A) = \{\tilde{\mathbf{c}}_{\mathcal{O}_2}, \tilde{\mathbf{c}}_{\mathcal{O}_4}, \tilde{\mathbf{c}}_{\mathcal{O}_3}, \tilde{\mathbf{c}}_{\mathcal{O}_1}, \tilde{\mathbf{c}}_{\mathcal{O}_b}\}$ for $A = [V] = \{1, 2, 3\}$. $\sim_s$ equivalence states that there exists a permutation matrix $\boldsymbol{P}$ which aligns the object order in $\tilde{\mathbf{c}}$ to match with $\mathbf{c}$ and there exists and invertible affine mapping $\boldsymbol{A}$ such that $\tilde{\mathbf{c}}_{\mathcal{O}_k} = \boldsymbol{A}\mathbf{c}_{\mathcal{O}_k} \forall k \in \{1, 2, 3, 4\}$.

Proof Sketch. The proof is detailed in Appendix C.4. To prove the following result, we follow multiple steps as described below: (i). We demonstrate the distribution $p(\mathbf{c})$ obtained as a result of lemma 4.2 is non-degenerate and a valid distribution, (ii). With the above results, we demonstrate invertibility restrictions on mixing functions, (iii). Finally, we constrain the subspace to affine, demonstrating $\sim_s$ of aggregate content $\mathbf{c}$. □

Theorem 4.4 (Invariance of aggregate content)**.** *For any subset $A, B \subseteq [V]$, such that $|A| > 0, |B| > 0$ and both A, B satisfy an assumption 4.1, we consider aggregate content to be invariant if $f_A \sim_s f_B$ for data $\mathcal{X}^A \times \mathcal{X}^B$.*

Intuition: Invariant slots

Considering an example 4.1, with $A = \{1, 3\}, B = \{2, 3\}$, such that sets A, B are viewpoint sufficient. Let f_A and f_B, be trained models on $\mathcal{X}^A$ and $\mathcal{X}^B$ respectively. Resulting in $\mathbf{c} = f_A^{-1}(\mathbf{x}^A; \mathbf{v}^A) = \{\mathbf{c}_{\mathcal{O}_1}, \mathbf{c}_{\mathcal{O}_2}, \mathbf{c}_{\mathcal{O}_3}, \mathbf{c}_{\mathcal{O}_4}, \mathbf{c}_{\mathcal{O}_b}\}$ and $\tilde{\mathbf{c}} = \tilde{f}_B^{-1}(\mathbf{x}^B; \mathbf{v}^B) = \{\tilde{\mathbf{c}}_{\mathcal{O}_2}, \tilde{\mathbf{c}}_{\mathcal{O}_4}, \tilde{\mathbf{c}}_{\mathcal{O}_3}, \tilde{\mathbf{c}}_{\mathcal{O}_1}, \tilde{\mathbf{c}}_{\mathcal{O}_b}\}$. Theorem 4.4 states that the representations $\mathcal{T}_{\theta^B}^{-1}(\tilde{\mathbf{c}}_{\mathcal{O}_k})$ can be mapped to $\mathcal{T}_{\theta^A}^{-1}(\mathbf{c}_{\mathcal{O}_k})$ by permuting object indices and an affine transformation.

Proof Sketch. The proof is detailed in Appendix C.4. To prove this, we extend the proof of Theorem 4.3, and establish that there exist two inevitable affine functions h_A, h_B for mixing functions $f_A, f_B : \mathcal{C} \times \mathcal{V} \to \mathcal{X}$ to map representations $\mathbf{c}$ with a given view set $\mathbf{v}^A$ to observations $\mathbf{x}^A$. Later, we show that, in the case of invariance, an affine mapping exists from h_A to h_B. □

Theorem 4.5 (Approximate representational equivariance)**.** *For a given aggregate content $\mathbf{c}$, for any two views $\mathbf{v}, \tilde{\mathbf{v}} \sim p_A(\mathbf{v})$, resulting in respective scenes $\mathbf{x} \sim p_A(\mathbf{x} \mid \mathbf{v}, \mathbf{c})$ and $\tilde{\mathbf{x}} \sim p_A(\mathbf{x} \mid \tilde{\mathbf{v}}, \mathbf{c})$, for any homeomorphic transformation*

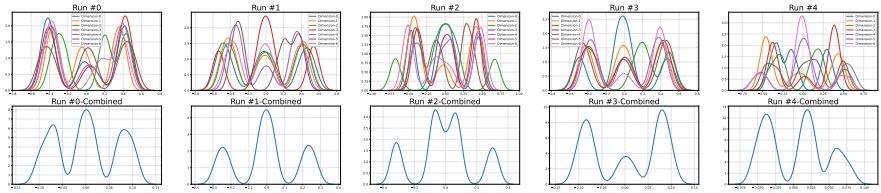


Figure 4.10: Identifiability of $q(\mathbf{c})$. The top row indicates individual feature distribution across five different runs. The bottom row reflects the feature feature distribution, which we use as a proxy for multi-dimensional features given Lemma C.2. As observed, mean feature distribution across runs is either scaled, shifted, or split (increase in number of modes); this provides strong evidence of recovery of the latent space up to affine transformations, empirically verifying our claims in Theorem 4.3.

$h_x \in \mathcal{H}_x$ such that $h_x(\mathbf{x}) = \tilde{\mathbf{x}}$, their exists another homeomorphic transformation $h_v \in \mathcal{H}_v$ such that $\mathcal{H}_v \subseteq \mathcal{H}_x \subseteq \mathbb{R}^{\dim(\mathbf{x})}$ and $\mathbf{v} = h_v^{-1}\left(f_d^{-1}(h_x(\mathbf{x}); \mathbf{c})\right)$.

Remark 4.1. *Note that the theorem only says that the transformation function transforming the view representations $\mathbf{v}$ as an effect of the homeomorphic transformation of $\mathbf{x}$ lies in the same subspace of input transformations.*

Intuition: Approximate equivariance

In the scenario when the cameras are positioned such that they have overlapping fields of view, and their relative pose (rotation and translation) must avoid degeneracies like aligning on the same plane or mapping points to infinity. This results in the transformation between views being smooth, invertible, and consistent. If the scene is planar or depth variations are minimal, the homography can capture the transformation accurately without the need for inverse rendering. Notably, the cameras should have non-zero rotation and translation to avoid collapsing the scene, and their intrinsic parameters must be known or identical to prevent distortions. When the scenario satisfies all the above properties, the 2D homography transformation $\boldsymbol{H}$ between two camera views can be learned as a homeomorphic transformation (Hartley and Zisserman, 2003).

Proof Sketch. The proof is detailed in Appendix C.4. We prove the following result by following the steps in Theorem 4.4, over a view distribution $p(\mathbf{v})$ but for a fixed content vector $\mathbf{c}$. □

Table 4.4: Comparing identifiability of $q(\mathbf{s})$, $q(\mathbf{c})$, and $p(\mathbf{v})$ scores wrt existing OCL methods.

METHOD	CLEVR-MV			GQN			GSO		
	SMCC ↑	INV-SMCC ↑	MCC ↑	SMCC ↑	INV-SMCC ↑	MCC ↑	SMCC ↑	INV-SMCC ↑	MCC ↑
AE	$0.32 \pm .02$	-	-	$0.29 \pm .02$	-	-	0.24 ± 0.08	-	-
SA	$0.47 \pm .03$	-	-	$0.38 \pm .02$	-	-	0.28 ± 0.06	-	-
PSA	$0.49 \pm .02$	-	-	$0.38 \pm .02$	-	-	0.30 ± 0.04	-	-
MulMON	$0.61 \pm .03$	$0.62 \pm .02$	-	$0.59 \pm .06$	$0.61 \pm .02$	-	0.56 ± 0.04	0.48 ± 0.06	-
OCLOC	$0.63 \pm .02$	$0.64 \pm .01$	$0.48 \pm .04$	$0.60 \pm .03$	$0.60 \pm .01$	$0.42 \pm .08$	0.58 ± 0.04	0.54 ± 0.03	0.46 ± 0.04
VISA	$0.67 \pm .01$	$0.66 \pm .01$	$0.60 \pm .04$	$0.59 \pm .01$	$0.63 \pm .01$	$0.52 \pm .03$	$0.60 \pm .03$	$0.61 \pm .02$	$0.58 \pm .03$

4.7 VISA: EMPIRICAL EVALUATION

Given the work's theoretical focus, experimentally, we aim to provide strong empirical evidence of our identifiability, invariance, and equivariance claims in a multiview setting. We also extend our experiments to standard imaging benchmarks, including CLEVR-MV, CLEVR-AUG, GQN (Li et al., 2020); we additionally demonstrate the framework's scalability to highly diverse setting with GSO (Downs et al., 2022) and proposed datasets MV-MOVIC, MV-MOVID which are multiview versions of MoViC dataset with fixed and varying scene-specific cameras (Greff et al., 2022).

EXPERIMENTAL SETUP. To verify our claims on (i) identifiability claim, we train our model on a given view subset $A \subseteq [V]$ and compare view averaged slot mean correlation coefficient (SMCC) measure Kori et al., 2024c, (ii) invariance claim, we train multiple models on different subsets of viewpoints $A, B \subseteq [V]$ and compare the aggregate content representations across models, quantifying the similarities with SMCC, we consider this measure to be invariant SMCC (INV-SMCC), and finally, (iii) for subspace equivariance, we consider a trained model with a view subset $A \subseteq [V]$ and compute MCC of view information $\mathbf{v}$ by applying random homeomorphic transformations on samples $\mathbf{x}^A \sim \mathcal{X}^A$ (which can also be done by considering samples $\mathbf{x}^B \sim \mathcal{X}^B$, where cameras relative position satisfy the required constraints 4.5, and analyse $p(\mathbf{v}^A)$ and $p(\mathbf{v}^B)$).

MODELS & BASELINES. We consider two ablations with two types of decoders: (i) additive with MLPs and spatial broadcasting CNNs and (ii) non-additive decoders, which include transformer models. In all cases, we use LeakyReLU activations to satisfy the weak injectivity conditions

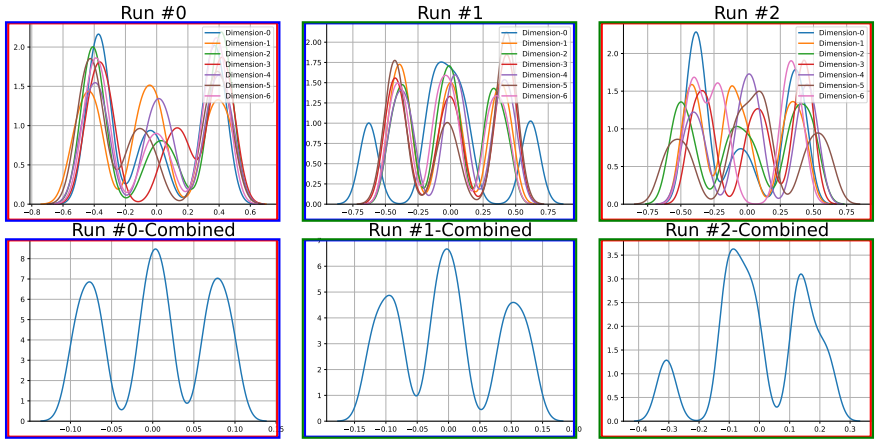


Figure 4.11: Viewpoint invariance of $q(\mathbf{c})$. The top and bottom row indicates individual feature levels and mean feature distributions, respectively. Each columns reflect marginalised aggregate content distribution $q(\mathbf{c})$ when trained with different view pairs {(blue, red), (green, blue), and (green, red)}, respectively. As the resulting distributions with different datasets only vary by an affine transformation, providing strong evidence for Theorem 4.4.

Table 4.5: Identifiability and generalisability analysis on MV-MoViC dataset.

Method	In-domain Results				Out-of-domain Results			
	mBO ↑	SMCC ↑	INV-SMCC ↑	MCC ↑	mBO ↑	SMCC ↑	INV-SMCC ↑	MCC ↑
SA-MLP	0.28 ± 0.091	0.36 ± 0.004	-	-	0.26 ± 0.08	0.38 ± 0.006	-	-
PSA-MLP	0.30 ± 0.022	0.38 ± 0.002	-	-	0.30 ± 0.03	0.40 ± 0.005	-	-
VISA-MLP	0.28 ± 0.021	0.52 ± 0.021	0.61 ± 0.023	0.54 ± 0.026	0.27 ± 0.02	0.51 ± 0.029	0.58 ± 0.031	0.52 ± 0.021
SA-Transformer	0.34 ± 0.014	0.36 ± 0.016	-	-	0.33 ± 0.041	0.36 ± 0.043	-	-
PSA-Transformer	0.37 ± 0.021	0.38 ± 0.007	-	-	0.37 ± 0.033	0.39 ± 0.016	-	-
VISA-Transformer	0.38 ± 0.008	0.44 ± 0.003	0.46 ± 0.001	0.53 ± 0.011	0.36 ± 0.017	0.46 ± 0.033	0.46 ± 0.018	0.55 ± 0.082

(Assumption 5.1). In terms of object-centric learning baselines, we compare with standard additive autoencoder setups following (Brady et al., 2023), slot-attention (SA) (Locatello et al., 2020b), probabilistic slot-attention (PSA) (Kori et al., 2024c), MulMON (Li et al., 2020), and OCLOC (Yuan et al., 2024).

CASE STUDY 1: ILLUSTRATION OF IDENTIFIABILITY. To definitively show the validity of our claims about identifiability (Theorem 4.3, Theorem 4.4, and Theorem 4.5), we created a synthetic unconfounded scenario for modelling. This provides us with two data modalities, Figure C.1 (i) projected point cloud data, and (ii) corresponding imagery data, we detail point cloud illustrations in appendix C.5.1. Additionally, this dataset also provides us with the ground truth object and viewpoint features for evalua-

tion. To visualise the aggregate mixture, following Lemma C.2, we use the projected GMM to interpret the distribution of random variables in $\mathbb{R}^d$.

The data-generating process is thoroughly explained in the Appendix C.2.1. In Figure 4.10, we display the distributions of marginalized aggregate content distribution $q(\mathbf{c})$, comparing individual features and a mean feature across different runs that are either scaled, shifted, or split (increase in number of modes), which is reflective of affine transformation of features across runs. To quantitatively measure the same, we computed SMCC and observed it to be $\mathbf{0.72} \pm 0.04$, empirically verifying our Theorem. 4.3. Furthermore, to illustrate the invariance of distribution $q(\mathbf{c})$ across viewpoints (Theorem 4.4), we consider three different viewpoints. We use all possible pairs to learn $q(\mathbf{c})$ distributions as illustrated in Figure 4.11, where the distributions are described `w.r.t` viewpoints described by {g, r}, {r, b}, and {g, b}, respectively. These distributions were also found to have similar properties as before, with an observed SMCC of $\mathbf{0.71} \pm 0.11$, further confirming the claims in Theorem 4.4. Additionally, Figure 4.1(c) demonstrates the improvement in identifiability as the number of viewpoints increases.

CASE STUDY 2: IMAGING APPLICATIONS. We first evaluate the framework on standard benchmarks, specifically focusing on CLEVR-MV, CLEVR-AUG, GQN, and GSO with simple objects. Given the *true generative factors* are unobserved, we derive our quantitative assessments from multiple runs. The results are shown in Table 4.4, confirming the validity of our theory on imaging datasets. Regarding the baseline comparisons that utilize a single viewpoint, the INV-SMCC mirrors the SMCC due to its inherent design (*i.e.*, aggregation of a set with a single element is the same element). Moreover, in the case of AE, SA, PSA, and MULMON, the models do not estimate view information but either treat them independently or use the observed view conditioning, rendering the MCC metric inapplicable. Figure C.6 showcases how the number of viewpoints impacts the identifiability of the $\mathbf{s}, \mathbf{v}$, and $\mathbf{c}$ variables; the involved experiments reflect the increase in performance with an increase in the number of views, across all benchmark datasets.

Additionally, we demonstrate our methodology on proposed complex datasets, MV-MOVIC and MV-MOVID, the latter dataset enables us to examine the model performs when the assumption 4.1 is not satisfied. To

evaluate model behaviour in an environment with consistent objects but with different viewpoints, we conducted in-domain and out-of-domain (OOD) evaluations. For in-domain analysis, the model is trained and assessed on the same viewpoint group $A = [1, 2, 3]$. Conversely, for OOD evaluation, we consider the previously trained model but test it against a new set of viewpoints $B = [3, 4, 5]$. The findings presented in Table 4.5 regarding the MV-MoViC dataset reveal that the SMCC, INV-SMCC, and MCC metrics show similar performance across both domains. This indicates that the distributional characteristics remain unchanged when both the training and testing environments contain the same objects. The MV-MoViD dataset analysis can be found in Appendix C.5.

4.8 CONCLUSION & DISCUSSION

Understanding when object-centric representations are identifiable is crucial for scaling slot-based methods to high-dimensional images with correctness guarantees. Unlike prior work that primarily examines slot mixing functions, we establish identifiability through distributional assumptions on the slot latent space, drawing inspiration from mixture models. We propose a probabilistic slot-attention algorithm that enforces an aggregate mixture prior over slot representations, ensuring stability across runs and tractability in sampling. Our empirical results verify the theoretical identifiability claim and demonstrate that our approach achieves minimal compositional contrast without explicit optimization toward that objective.

Additionally, we extend identifiability guarantees to multi-view setups by ensuring the disentanglement of view-specific slots, invariant content representations, and view representations—without requiring additional supervision. We validate these findings both theoretically and empirically using synthetic datasets, object-centric benchmarks, and large-scale datasets with complex decoders, including transformer-based architectures. The results confirm the model's scalability and effectiveness in resolving view ambiguities, making it a robust framework for learning structured object-centric representations in diverse imaging applications.

Limitations & Future Work. We recognize that our assumptions, that may not always hold in practice. Specifically, the *weak injectivity* of the

mixing function may not be valid for all architectures, as detailed in the sufficiency conditions (Appendix B.3). Similarly, while generally applicable, our use of piecewise-affine functions may not always align with real-world problem structures, particularly in cases of model misspecification. Like other object-centric learning methods, we assume that the mixing function is *permutation-invariant* to slots, making it non-invertible. Addressing this constraint in future work could strengthen identifiability guarantees. In multi-view scenarios, our assumption of *viewpoint sufficiency* is strong but did not appear limiting on the MV-MoVID dataset. However, its impact on real-world applications requires further investigation.

Despite these limitations, our theoretical contributions provide a valuable extension to the foundations of object-centric learning, with potential future directions including relaxing identifiability assumptions, studying slot compositional properties, and exploring applications in generalization, world modelling, and planning.

Part-2

Foundations of Human-like Visual Reasoning

Dravyāṇy-evānu darśanaṃ paramāṇava iva sthitāḥ |
Saṃyoge buddhir-utpannā teṣāṃ tattvavivecanāt ||

"Objects exist in perception like atoms in space; From their composition, understanding arises."

This page is intentionally left blank.

5

COMPOSITIONAL REASONING

"Causal models are mental abstractions that encode our beliefs about how the world works."

– Judea Pearl

The goal of Object-Centric Learning (OCL) is to enable machine learning systems to decompose complex scenes into discrete, interacting objects, supporting compositional generalization and human-like reasoning. However, existing OCL methods often fail to capture interactions from both attribute-level (*semantic*) and object-level (*spatial*) perspectives. While scene graph methods complement OCL by abstracting scenes as structured graphs, they typically rely on supervision. This position paper argues for a *probabilistic perspective* on Scene Graph Modelling (SGM), grounded in *causal abstraction* as a unifying view on causality, OCL, and scene graphs by considering object interactions as invariant mechanisms within *object-level* graphs, enabling us to generate causally consistent scene compositions. We substantiate our position with thorough conceptual discussion, rigorous definitions, conjectures, and examples, demonstrating how this perspective bridges the gap between unsupervised object discovery and explicit scene graph reasoning.

5.1 MOTIVATION

To achieve human-level understanding, machine learning systems must be capable of representing the world in terms of distinct, interacting objects (Bengio et al., 2013b). The concept of *objectness*—the ability to perceive the environment as a composition of discrete, identifiable entities—has long been recognized as critical for enabling *situation-aware AI systems* that

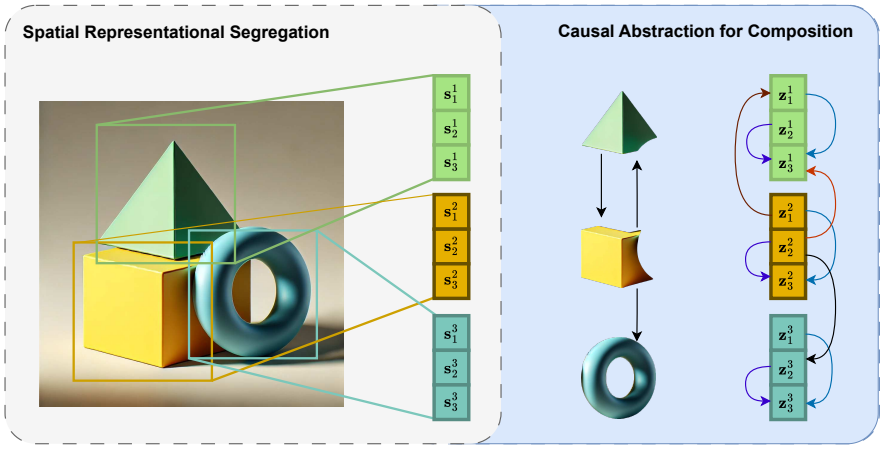
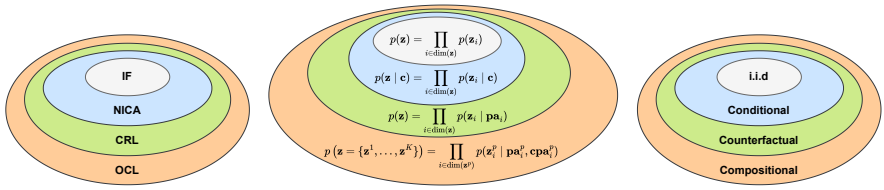


Figure 5.1: Figure illustrates the process of causal abstraction for a composition resulting in causally related variables in $\mathbf{z}$ using spatially segregated object-centric representations $\mathbf{s}$.

exhibit *human-like reasoning capabilities* (Lake et al., 2017; Schölkopf and Kügelgen, 2022). This fundamental ability to decompose complex scenes into objects and parts forms the basis of *compositional generalization*, a key feature that allows humans to efficiently learn from limited examples and adapt to new environments (Rock, 1973; Hinton, 1979).

The fundamental requirement of compositional OCL is to ensure attribute and object level disentanglement. Attributes are common features across all objects involved in the scene generation for example features like *light-position, camera position, and so on.* Attribute disentanglement is a key objective in generative modelling such as *Variational Auto Encoders (VAEs)* (Kingma and Welling, 2013). The identifiability of these features has been a common area of research in identifiable representation learning, particularly with the *Non-linear Independent Component Analysis (NICA)* learning paradigm (Khemakhem et al., 2020a; Khemakhem et al., 2020c; Willetts and Paige, 2021). However, given these attributes are learned and not well defined, learning causal relationships among these attributes is another ill-defined problem and is the core of *Causal Representation Learning (CRL)* (Schölkopf et al., 2021; Yao et al., 2024a).

While NICA and CRL have made substantial advances in learning complex latent structures with identifiability guarantees, their applications in OCL remain limited. Most of the works in OCL that perform attribute



(a) Learning **(b)** Prior factorisation **(c)** Generation

Figure 5.2: The figure illustrates a hierarchical progression across learning paradigms, latent variable factorization, and sample generation capabilities grouped by color-coded regions. **IF** assumes fully independent latent variables, with a prior factorization $\prod_i p(\mathbf{z}_i)$, enabling *iid sample* generation. **NICA** introduces conditional independencies in the latent space, factorized as $\prod_i p(\mathbf{z}_i \mid \mathbf{c})$, enabling *conditional samples*. **CRL** generalizes NICA by uncovering cause-effect relationships via SCMs, with factorization $p(\mathbf{z}_i \mid \mathbf{pa}_i)$, allowing *conditional and counterfactual sample generation*. **OCL** extends CRL by structuring the latent space as a mixture of objects, capturing both feature-level and object-level causal interactions, with factorization $\prod_i p(\mathbf{z}_i^p \mid \mathbf{pa}_i^p, \mathbf{cpa}_i^p)$, enabling *conditional, counterfactual, and compositional sample* generation. In all cases, an appropriate mixing function is assumed.

disentanglement, including (Kim and Mnih, 2018a; Burgess et al., 2019; Greff et al., 2019; Greff et al., 2020; Chen et al., 2021; Kori et al., 2024c), rely on *Independent Factorisation (IF)* of attributes similar to VAEs. Some recent advances in OCL have demonstrated the potential to imbue machine learning models with the notion of objectness by learning structured, spatially disentangled representations directly from raw observations (Locatello et al., 2020b; Engelcke et al., 2021; Kori et al., 2024b; Löwe et al., 2024). While in most of these methods, the main focus is on object discovery, the interaction between and across object features is, in our view, yet to be thoroughly explored. Here, identifiability is crucial because it dictates the conditions under which latent causes can be reliably recovered from observed data (Hyvärinen and Pajunen, 1999; Hyvärinen and Oja, 2000; Locatello et al., 2019). Recent research in OCL has established *theoretical guarantees* with assumptions on *mixing functions or latent distributions* that ensure object-centric representations are identifiable (Brady et al., 2023; Kori et al., 2024c; Lachapelle et al., 2024b). However, these methods rely on the IF setting, limiting the possibilities of learning relations among and across object features. To model these relations as illustrated in Figure 5.1, we need concepts from CRL to capture scene compositions . The problem of composition is non-trivial as the interactions increase combinatorially

with the increase in number of objects in a scene (Kügelgen et al., 2020; Tangemann et al., 2021; Wiedemer et al., 2023).

Here we argue that an ideal OCL method should aim to capture higher-order *(i) conditional, (ii) causal, and (iii) compositional relationships* between attributes and objects, where the system learns representations that are inherently compositional and capture both feature-level and object-level causal interactions. This view aligns with philosophical perspectives, such as Kant's view that causality is a necessary structure imposed by the mind to organise sensory experience (Kant, 1908), and Fodor's idea of compositional symbolic representations (Fodor, 1975). It also resonates with neuroscientific insights into hierarchical processing in the brain, where dynamic interactions between attributes and objects occur, and causal reasoning, akin to Bayesian inference mechanisms used in the prefrontal cortex (Tenenbaum et al., 2011). By integrating these principles, ideal OCL can bridge low- and high-level semantic understanding, reflecting the complex nature of human perception and the structured nature of real-world environments.

Scene Graph Modelling (SGM) is another line of work which deals with the abstraction of objects/components in the scene and their complex relationships and provides rich semantic information of an image (Gu et al., 2019), in the form of `<subject, object, predicate>`. Scene graphs are mainly considered in scenarios like visual question answering, image captioning, and action recognition (Teney et al., 2017; Yao et al., 2018; Li et al., 2022). Scene graphs naturally align with the goals of OCL, representing a scene's latent structure as a graph where nodes correspond to objects, and edges represent semantic relationships, such as spatial, functional, or causal interactions (Johnson et al., 2015; Krishna et al., 2017; Gu et al., 2019); however, most of these methods heavily rely on supervision for identifying objects and their interactions. Sequential unobserved object discovery (Eslami et al., 2018a; Kosiorek et al., 2018; Engelcke et al., 2019; Kügelgen et al., 2020) implicitly set an order of discovery, having some implicit object-to-object relations. Singh et al. (2021) propose auto-regressive object interaction, built upon the spatial discovery method of Locatello et al. (2020b). Recent methods (Gao et al., 2024; Kim et al., 2024; Wang et al., 2024; Wei et al., 2024) use large language models (LLMs) or large

vision language models (VLMs) as an implicit scene graph which is used in conditioning the generation of samples. Most of these do not provide explicit control over attributes and object interactions. (Gao et al., 2024) uses GPT4 to generate scene graphs from text prompts, providing object-level control with explicit scene graphs. However, these methods do not provide identifiability guarantees or explicit scenarios for identifiability. Kügelgen et al. (2020) provide a graphical abstraction for layer-wise compositional abstraction, Deng et al. (2021) provide a graphical abstraction of part-object relations, capturing the hierarchy involved in scene generation. In a similar spirit, we propose *causal abstraction for composition*, where we model object interactions as interactions between *object-level* graph. Unlike (Deng et al., 2021; Yuan et al., 2024), which aims to learn to pose and view equivariant object representations, the *object-level* graph in this setting captures the *invariant mechanisms* across all the objects.

Beyond scene graph modelling, there exists a broader class of structured probabilistic models that operate on similar principles for relational reasoning, Probabilistic Relational Models (PRMs) (Getoor et al., 2001; Getoor and Taskar, 2007), which extend traditional probabilistic graphical models to relational domains. PRMs define a joint probability distribution over a set of entities and their attributes, along with the probabilistic dependencies among them via explicitly modelled relationships. In essence, PRMs represent structured worlds as graphs, where nodes correspond to object classes with associated attributes, and edges represent probabilistic dependencies between these objects, closely mirroring the structure of scene graphs in vision. Recent adaptations of PRMs into neural and differentiable frameworks (Kazemi and Poole, 2018; Zambaldi et al., 2018). PRMs are not particularly explored in the case of object-centric learning, though they provide a nice connection between classical structured models and modern scene graphs in vision.

However, the challenge of the identifiability of representations and dependencies still persists. Identifiability in modern machine learning models can be considered as a Bayes optimal model that guarantees the test loss to have a unique minimizer, essential to recover the true data-generating process. Here in this chapter, with a probabilistic perspective on SGM, we position compositional OCL as a general learning paradigm that subsumes

CRL, NICA, and IF models as special cases and should be an integral part of the research agenda. With this, we consider the progression of learning paradigms as follows (illustrated in Figure 5.2): (i) **IF**: assumes fully independent latent variables and enables the generation of *iid samples*. (ii) **NICA**: introduces *conditional independencies* in latent variables, allowing *conditional sample generation*. (iii) **CRL**: extends NICA by uncovering cause–effect relationships through Structural Causal Models (SCMs), resulting in *counterfactual reasoning*. (iv) **OCL**: builds upon CRL by structuring the latent space as a *mixture of objects*, capturing both feature-and object-level causal interactions, thus enabling the generation of *compositional samples*. In summary, our main contributions are:

1) Highlighting the modelling scenarios in both CRL and OCL fields, unifying them with *causal abstractions*;

2) Provide a probabilistic interpretation of scene graphs through *compositional graphs*, with multiple examples and scenarios demonstrating its implications;

3) Introduce the concept of *latent compositional models with mechanism invariance*, discuss its implications for learning compositional and causal graphs, along with possible modelling approaches and theoretical extensions to validate guarantees.

5.2 COMPOSITIONAL OCL: A UNIFYING VIEW

Here, we introduce notations, outline the components of the data generation process and introduce the concept of causal abstraction from a compositional perspective.

5.2.1 *Data Generating Process*

Let $\mathcal{K} = \{1, \dots, K\}$ represent all object indices, with a subset $V \subseteq \mathcal{K}$ generating the observation $\mathbf{x} \in \mathcal{X}$. As shown in Figure 5.3 (and in Equation 5.1), $\mathbf{x}^V = \{\mathbf{x}^v \in \mathcal{X}^v, \forall v \in V\}$ represents image partitions, where $\mathbf{x}$ is the composition of elements in $\mathbf{x}^V$, and $\mathcal{X}^v \subset \mathcal{X} \subset \mathbb{R}^{d_x}$. Observed variable y

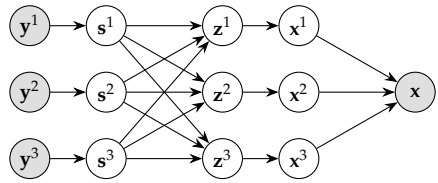


Figure 5.3: This graphical model illustrates the data generation process, starting with observed grouping variables $y^1, \ldots, y^{|V|}$ ($|V| = 3$ shown for illustration), which are used to sample independent object representations $\mathbf{s}^1, \ldots, \mathbf{s}^K$. These representations are then combined to form $\mathbf{z}$, capturing both attribute-level and object-level relationships. Finally, $\mathbf{z}$ is used to generate object-level partitions in the observation space, $\mathbf{x}^1, \ldots, \mathbf{x}^{|V|}$. All partitions together compose the observed scene, $\mathbf{x}$.

groups components ($\mathbf{x}^v$s) across $\mathcal{X}$. Latent variables $\mathbf{z} = \{\mathbf{z}^v \in \mathcal{Z}, \forall v \in V\}$, with $\mathcal{Z} \subset \mathbb{R}^{d_z}$, encode dependencies across features $\mathbf{z}_j^v$. Each partition $\mathbf{x}^v$ depends only on its corresponding $\mathbf{z}^v$ with relation $\mathbf{x}^v = f_d(\mathbf{z}^v)$. Independent counterparts $\mathbf{s} = \{\mathbf{s}^1, \ldots, \mathbf{s}^{|V|}\}$ are defined as $\mathbf{s} = \{g_e(\mathbf{x}, y^1), \ldots, g_e(\mathbf{x}, y^{|V|})\}$, where $\mathbf{s} \in \mathcal{S} \subseteq \mathcal{S}^1 \times \cdots \times \mathcal{S}^K$, $\mathcal{S}^k \subseteq \mathbb{R}^{d_s}$. A volume-preserving transformation $\mathbf{z} = g_z(\mathbf{s})$ captures relationships among object features. Finally, the compound feature extractor is defined as $\phi(\mathbf{s}) := (g_s \circ (f_c \circ f_d \circ g_z))(\mathbf{s})$, extracting features for a composed image rather than an observed image.

$$\begin{aligned}\mathbf{x} &= f_c([\mathbf{x}^1, \ldots, \mathbf{x}^{|V|}]) = f_c(f_d([\mathbf{z}^1, \ldots, \mathbf{z}^{|V|}])) = f_c([f_d(\mathbf{z}^1), \ldots, f_d(\mathbf{z}^{|V|})]) \\ &= f_c([f_d(g_z(y^1, \mathbf{s})), \ldots, f_d(g_z(y^{|V|}, \mathbf{s}))])\end{aligned} \tag{5.1}$$

Illustration OCL: Grouping from a compositional perspective refers to spatial disentanglement of and categorisation of objects in V, the total number of distinct groups. Providing control over the generation of scenes with objects of interest given by y^v, this grouping can also be with respect to size, colour, or any of the observable properties which capture the uniqueness of objects in a scene.

Illustration CRL: Note that in the case of CRL methods, we do not need $f_c, \mathbf{x}$ as $\mathbf{x}^v$ are observed, y behaves as the observational grouping variable. This grouping of observed variables is common in many real-world applications, as detailed in Morioka and Hyvärinen (2023), and is used in many different forms, as in the form of multiple views in Yao et al.

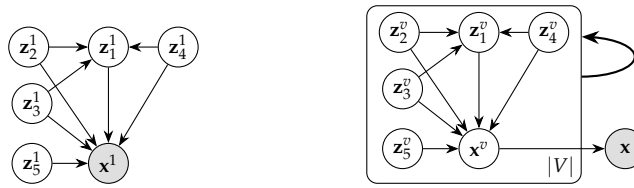


Figure 5.4: (a) Instantaneous causal graph representing the causal relationships among variables for a single object, where $\mathbf{z}_1^1$ to $\mathbf{z}_5^1$ correspond to *object-shadow, object-size, light-position, object-position, and object-color*, respectively. **(b) Compositional graph** depicted using plate notation, where the plate labelled $|V|$ represents multiple instances of the invariant causal graph for different objects. Interactions across objects are illustrated with elliptical arrows spanning the plate, and the observed data $\mathbf{x}$ is modelled as the combined result of all $|V|$ objects and their interactions.

(2023), as augmentations in Von Kügelgen et al. (2021), and as interventional set in Brehmer et al. (2022).

5.2.2 *Causal Abstraction for Composition*

We introduce the notion of compositional graphs, building the formalisation of causal graphs (Pearl, 2009) and viewing them as probabilistic interpretations of scene graphs.

Remark 5.1. *A causal graph $\mathcal{G}$ (definition 5.1)factorises the distribution $p(\mathbf{z}) = \prod_{i=1}^{n} p(\mathbf{z}_i \mid \mathbf{pa}_i)$, where $\mathbf{pa}_i \subseteq \{\mathbf{z}_1, \ldots, \mathbf{z}_n\} \setminus \mathbf{z}_i$ is a set of parents of $\mathbf{z}_i$ w.r.t. graph $\mathcal{G}$.*

Example 5.1. *(Instantaneous Causal Effects.) As shown in Figure 5.1, the observation is modelled as the output of an SCM, with inputs including light source position, object positions, shadows, sizes, and colours. The interactions between these features are depicted in Figure 5.4a. Notably, object-specific properties such as pose and orientation are treated as confounders and are excluded from the causal graph, as we focus solely on features that represent invariant mechanisms across all objects in the dataset.*

Definition 5.1 (Causal Graph: $\mathcal{G}$)**.** *Let the causal graph $\mathcal{G} = (V, E)$ be a directed acyclic graph (DAG) that represents causal relationships:*

- *$V = \mathbf{z} \in \mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\} \in \{\mathcal{Z}_1, \ldots, \mathcal{Z}_n\}$ are the nodes, representing causal variables.*

- $E \subseteq V \times V$ *are the directed edges encoding pairwise causal dependencies between variables, i.e.,* $\mathbf{z}_i \to \mathbf{z}_j$ *represents that* $\mathbf{z}_i$ *causally influences* $\mathbf{z}_j$.

Definition 5.2 (Compositional Graph: $\mathfrak{G}$)**.** *A compositional graph is a* meta-graph *capturing interactions between different causal graphs, which can be denoted with* $\mathfrak{G} = (V^C, H) = (\bigcup_{k \in \mathcal{K}} V^k, \bigcup_{k \in \mathcal{K}} E^k \bigcup E^m)$*, where:*

- $V^C = \{\mathcal{G}^1, \ldots, \mathcal{G}^K\}$ *represents causal graph for individual* objects*, where* $\mathcal{G}^k = (V^k, E^k), \forall k \in [K]$ *is a causal graph as defined above.*
- $E^m = \{e_{ij}^{pq} \mid e_{ij}^{pq}$ *corresponds to* meta *edges capturing interactions between* i^{th} *variable of* $\mathcal{G}^p$ *and* j^{th} *variable of* $\mathcal{G}^q\}$, i.e., *edge* e_{ij}^{pq} *captures the causal influence between* $\mathbf{z}_j^q \to \mathbf{z}_i^p$

Remark 5.2. *In a compositional setting, we have a set of latent representations* $\mathbf{z} = \{\mathbf{z}^1, \ldots, \mathbf{z}^{|V|}\}$*, where* $\mathbf{z}^k$ *is sampled from separate environments given by* $\mathbf{z}^k \sim p(\mathbf{z}^k \mid \mathbf{cpa}^k, y^k)$*, where* y^k *categorical variables conditioning a variable to a particular environment. Similar to parents in causal setting we consider* $\mathbf{cpa}_i^p \subseteq \bigcup_{q \in V, q \neq p} \{\mathbf{z}_j^q \mid \forall j \in \dim(\mathbf{z}^q)\}$*, describing compositional parents, which is the set of all the variables across environments causing the feature* $\mathbf{z}_i^p$*, they can be treated as a causal parents but in a different environment. Invariant edges across environments can be considered causal graphs. Additionally, it is essential to note that compositional graphs are scene-dependent, capturing the interaction between invariant causal graphs. The resulting latent distribution can be factorised as:*

$$\begin{aligned} p(\mathbf{z}) &\propto \prod_{p,q \in \mathcal{K}, p \neq q} p(\mathbf{z}^p \mid \mathbf{z}^q, y^p) \\ &= \prod_{p \in \mathcal{K}} \prod_i p(\mathbf{z}_i^p \mid \mathbf{cpa}_i^p \cup \mathbf{pa}_i^p, y^p) \end{aligned} \tag{5.2}$$

Example 5.2 (Causal Abstraction for Composition.)**.** *Building upon Example 5.1, we extend the discussion to object relations within a scene, using the same instantiation of causal graphs for Torus, Cube, and Pyramid, as shown in Figure 5.1. This instantiation remains valid since the underlying mechanisms—such as shadow formation due to object size and light source—are consistent and invariant across different objects. The edges in the graph (ref. Figure 5.5) represent compositional behaviours; for example, the Cube's position causally influences the Pyramid's*

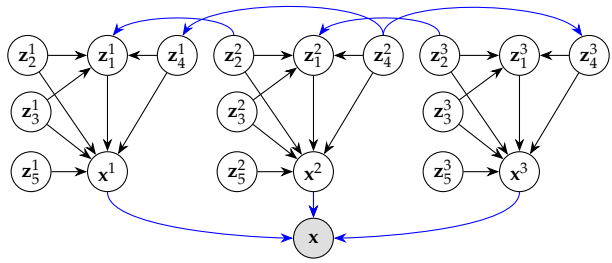


Figure 5.5: Example of particular instantiation of Figure 5.4b (with $V = \{1, 2, 3\}$, resulting in $\mathbf{x}^1, \mathbf{x}^2$ and $\mathbf{x}^3$ corresponding to *torus, cube,* and *pyramid*, respectively), illustrating the scene in Figure 5.1. Here, $\mathbf{z}_1^v$ to $\mathbf{z}_5^v$ correspond to *object-shadow, object-size, light-position, object-position, and object-color, respectively.* Finally, blue arrows in the graph correspond to compositional edges (E^m).

position when the Pyramid is placed above it. Similarly, the Torus' position is causally linked to the Cube's position.

The compositional graph provides a *probabilistic* interpretation of scene graphs, integrating both spatial and semantic relationships. Its edges capture attribute-level (semantic) reasoning, e.g., `<object-size, shadow-length, causes>`, or object-level (spatial) reasoning, e.g., `<torus, cube, leans on>`.

Lemma 5.1 (Acyclicity of composition)**.** *The combined compositional graph remains acyclic when:*

L.1: Circular consistency: No circular interactions are introduced between any graph pairs $(\mathcal{G}^p, \mathcal{G}^q)$, i.e., *there are no loops in* E^m. Example: *If there is an edge* $(\mathbf{z}_i^p \to \mathbf{z}_j^q) \in E^m$, *then* $(\mathbf{z}_j^q \to \mathbf{z}_i^p) \notin E^m$.

L.2: Causal consistency: For any two nodes $\mathbf{z}_i^p, \mathbf{z}_j^q$, *if there is an edge* $(\mathbf{z}_i^p \to \mathbf{z}_j^p) \in E^p$, *then* $(\mathbf{z}_j^q \to \mathbf{z}_i^p) \notin E^m$.

Interpretation: *The first constraint is straight forward, ensuring* E^m *does not contain any direct cycles. The second constraint preserves consistency in the causal direction between features across groups. Together, these constraints ensure* $\mathfrak{G}$ *retains its acyclic nature. In the context of physical systems, as illustrated in Figure 5.5, acyclicity in the composition of interacting objects is inherently maintained. This means that the cause-and-effect relationships between the objects — such as the influence of the light source on the shadows cast by the cube and torus or the way the position and size of one object can impact the other — follow a*

directed, non-circular flow. However, some complex biological systems do break these acyclicity properties (Sachs et al., 2005), which we do not consider in this work.

Assumption 5.1 (*Weak* Injectivity)**.** *Let $f : \mathcal{Z} \to \mathcal{X}$ be a mapping between latent space and image space, where $\dim(\mathcal{Z}) \leq \dim(\mathcal{X})$. The mapping f_d is weakly injective if there exists $\mathbf{x}_0 \in \mathcal{X}$ and $\delta > 0$ such that $|f^{-1}(\{\mathbf{x}\})| = 1$, $\forall \mathbf{x} \in B(\mathbf{x}_0, \delta) \cap f(\mathcal{Z})$, and $\{\mathbf{x} \in \mathcal{X} : |f^{-1}(\{\mathbf{x}\})| = \infty\} \subseteq f(\mathcal{Z})$ has measure zero w.r.t. to the Lebesgue measure on $f(\mathcal{Z})$ (cf. (Kivva et al., 2022)).*

Remark 5.3. *In words, Assumption 5.1 says that a mapping f_d is weakly injective if: (i) in a small neighbourhood around a specific point $\mathbf{x}_0 \in \mathcal{X}$ the mapping is injective – meaning each point in this neighbourhood maps to exactly one point in the latent space $\mathcal{Z}$; and (ii) while f_d may not be globally injective, the set of points in $\mathcal{X}$ that map back to an infinite number of points in $\mathcal{Z}$ (non-injective points) is almost non-existent in terms of the Lebesgue measure on the image of $\mathcal{Z}$ under f_d.*

Definition 5.3 (Additive composition function)**.** *A function $f_c : \mathcal{M} \times \mathcal{X} \to \mathcal{X}$ is said to be additive if, for $\mathbf{x} \sim \mathcal{X}, \mathbf{m} \sim \mathcal{M}$ with $\mathbf{x} = \{\mathbf{x}^1, \ldots, \mathbf{x}^{|V|}\}$ and $\mathbf{m} = \{\mathbf{m}^1, \ldots, \mathbf{m}^{|V|}\}$, it can be expressed as $f_c(\mathbf{m}, \mathbf{x}) = \sum_{v=0}^{|V|} \mathbf{m}^v \odot \mathbf{x}^v = \mathbf{x}$. When $\mathcal{M}$ is constrained, such that $\sum_{v=0}^{|V|} \mathbf{m}^v = 1, \mathbf{m}^v > 0$, the following properties are inherently satisfied:*

- *D.1. when $\mathbf{m}^v \odot \mathbf{x}^v$ satisfy $\beta-$disentanglement assumption B.1, given partition masks, composition function can be approximately inverted with $\mathbf{x}^v = \mathbf{m}^v \odot \mathbf{x}$,*
- *D.2. ordering invariance of elements in $(\mathbf{x}, \mathbf{m})$ together.*

Definition 5.4 (Latent compositional models (LCM))**.** *LCM is given by $\mathcal{M} = \langle \mathcal{X}, g_z, f_d, f_c, \mathcal{Y} \rangle$, which consists of:*

- *D.1. $\mathcal{X} = \mathcal{X}^1 \times \cdots \times \mathcal{X}^{|V|}$: observational data space,*
- *D.2. $g_z : \mathcal{S} \to \mathcal{Z}$: Structural Equation Model (SEM) capturing relation between feature $\mathbf{z}_i^p$ with $\mathbf{pa}_i^p \cup \mathbf{cpa}_i^p$,*
- *D.3. $f_d : \mathcal{Z}^v \to \mathcal{X}^v$: mixing function mapping individual slot subspace to observational subspace $\mathcal{X}^v$,*

D.4. $f_c : \mathcal{X}^1 \times \cdots \times \mathcal{X}^{|V|} \to \mathcal{X}$*: an additive composition function, given by definition 5.3,*

D.5. y*: observed grouping variable.*

To capture the notion of latent distribution respecting compositional nature, we consider parametrised Markov random fields (MRFs), capturing pairwise interactions between the elements of $\mathbf{s}_j^p$ and $\mathbf{z}_i^q$. Based on the parametrization of MRFs, we consider two different potential functions, g_z and $\bar{g}_z$, where g_z is a pairwise potential function, which captures causal relation across objects modelling compositions. In contrast, $\bar{g}_z$ captures the instantaneous causal interactions for a given component. As illustrated in Morioka and Hyvärinen (2023), the parametric interpretation of $g_z, \bar{g}_z$ is general enough to capture many different classes of SEMs, resulting in a latent distribution described in Equation 5.3.

$$p(\mathbf{z}) \propto \prod_{p,q \in \mathcal{K}} \exp\left(\bar{g}_z(\mathbf{z}^p, \mathbf{s}^p; y)\right) \cdot \left[\prod_{(i,j) \in |\mathcal{Z}^p| \times |\mathcal{Z}^q|} \exp\left(\lambda_{ij}^{pq} g_z(\mathbf{z}_i^p, \mathbf{s}_j^q)\right) \right] \tag{5.3}$$

Terms $g_z(\mathbf{z}_i^p, \mathbf{s}_j^q)$ captures the sample-specific functional relations like the influence of cube size on the shadow of a torus in Example 5.2, while λ_{ij}^{pq} captures the existence of these compositional edges. Here, the component level interaction $p(\mathbf{z}^p \mid \mathbf{z}^q)$ is modelled with an axillary variable $\mathbf{s}$ which are considered to satisfy ICA assumption resulting in $p(\mathbf{s}) = \prod p(\mathbf{s}^v)$ (ref. Figure 5.3). We do this by estimating $\mathbf{z}^p$ with all independent base representations $\{\mathbf{s}^1, \ldots, \mathbf{s}^{|V|}\}$.

Remark 5.4. *Note that based on the joint distribution, the potential function $\bar{g}_z$ depends on a component level grouping reflected by $y, \forall k \in [K]$, which in Morioka and Hyvärinen (2023) is assumed to be known in the form of observational groups, while in OCL, this further requires weak supervision (classification labels for objects in the scene), breaking equivariance property, which is implicit in OCL (Locatello et al., 2020b; Emami et al., 2022; Wang et al., 2023a; Kori et al., 2024b; Kori et al., 2024c). Additionally, the potential function g_z, needs a further set of assumptions to factories Equation 5.3 to reflect the directionality of cause and effect.*

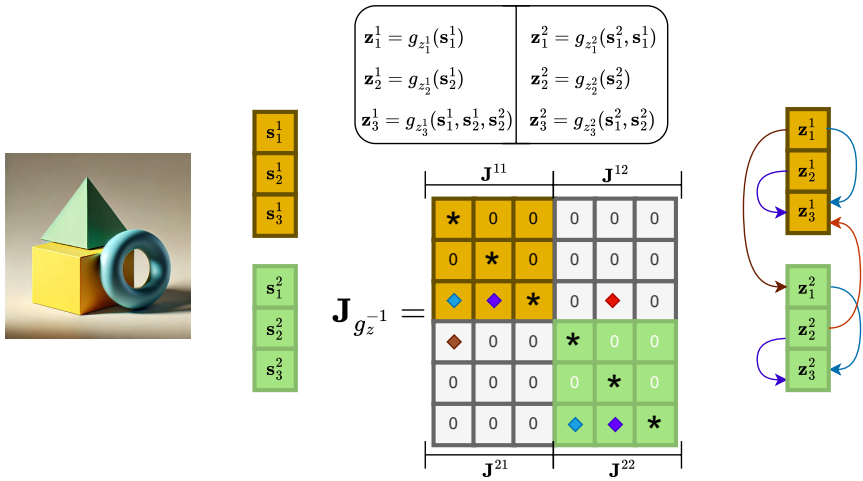


Figure 5.6: The figure demonstrates the Jacobian of g_z, and the resulting compositional graph between attributes $\mathbf{z}$ expressed as a function of independent variables $\mathbf{s}$. Block diagonals reflect the causal relations between attributes, while off-diagonal entries show compositional relations. Coloured diamonds in a Jacobian reflect entries corresponding to their edges in a compositional graph.

Graph Extraction: Considering the invariance of causal graphs across groups, we get the same causal graph $\mathcal{G}$ for all $k \in \mathcal{K}$. It is important to note that causal graph $\mathcal{G}$ is contained in a compositional graph $\mathfrak{G}$ and reflects the inter-group weighted adjacency matrix. Let $\boldsymbol{C} = \{\boldsymbol{C}^{pq}\}_{ij}$ correspond to adjacency matrices of $\mathfrak{G}$ for every node pair $(i, j) \in |\mathcal{Z}^p| \times |\mathcal{Z}^q|$ for a given scene, as detailed in examples 5.1 & 5.2. Similar to Reizinger et al. (2023), we link the notion of Jacobian $\boldsymbol{J}_{g_z^{-1}}$ of function g_z to compositional graph. Note we consider $\boldsymbol{C}$ to be concatenated matrix in $\mathbb{R}^{|V||d_z| \times |V||d_z|}$ resulting the combined Jacobian as illustrated in Figure 5.6. Finally, we consider two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ to be structurally equivalent ($\boldsymbol{A} \sim_{\text{DAG}} \boldsymbol{B}$) if both matrices follow the same sparsity structure.

Invariant Mechanisms: As detailed in examples 5.1 & 5.2, given the mechanisms are consistent across objects in a scene, we need a way to enforce this behaviour when our SCM is modelled in a parametric sense. We propose the following mechanism consistency regularisations (ref. definition 5.5), based on a Jacobian of the considered SCM.

Definition 5.5 (Mechanism consistency)**.** *The causal relation among features for a particular component is given by $\mathcal{G}^p$ which is captured by $d_s \times d_s$ Jacobian matrix given $\boldsymbol{J}^{pp}_{g_z^{-1}}$, we define mechanism consistency as:*

$$\mathcal{L}_{\text{MC}} = \mathbb{E}_{\mathbf{z}\in\mathcal{Z}} \left[\sum_{p=1}^{|V|} \|\boldsymbol{J}^{pp}_{g_z^{-1}}(\mathbf{z}) - \boldsymbol{J}^{00}_{g_z^{-1}}(\mathbf{z})\|_f \right] \tag{5.4}$$

Interpretation: Figure 5.6 illustrates this behaviours graphically. The regularisation basically encourages all the block-diagonals with dimension d_z of a Jacobian $\boldsymbol{J}_{g_z^{-1}}$ to follow the same sparsity structure. This basically forces the parametric SCM to use the same set of parameters for all objects.

Interestingly, additive decoder architectures, such as those employed in SA, PSA, and VISA (see Chapter 4), naturally satisfy mechanism consistency by design. In these models, the output is composed by summing the contributions of individual object-centric representations, with each object specific decoder sharing the same parameters. As a result, the corresponding Jacobians exhibit block-diagonal structures with identical sparsity patterns, aligning precisely with the mechanism consistency regularization described in Definition 5.5. However, the converse direction is particularly compelling: if imposing mechanism consistency as a soft inductive bias (via the regularizer $\mathcal{L}_{\text{MC}}$), could this encourage any general models to implicitly learn additive or spatially disentangled representations, addressed the main issues of performance and computational complexity with additive decoder.

Specifically, enforcing similar Jacobian structures across object slots can be interpreted as guiding the model toward discovering a shared causal mechanism or data-generating process, which is central to object-centric generalization and modular reasoning. By penalizing deviations in the local input-output Jacobians across different slots, we encourage functional symmetry and parameter reuse, one of the main property of additive, compositional models. Exploring this as an emergent property, rather than an architectural constraint, opens up exciting future research directions. For instance, such a regularization could allow flexible models (e.g., Transformers, Diffusion with cross-slot attention) to gradually approximate additive decoders to a certain degree.

DAG regularisation: Lemma 5.1 establishes conditions for compositional interactions to ensure the resulting joint composition graph is a DAG. To enforce DAG structure in SEM g_z, we apply a continuous DAG penalty over the Jacobian $\boldsymbol{J}$ (we later show, $\boldsymbol{J} \sim_{\text{DAG}} \boldsymbol{I} - \boldsymbol{C}$). We leverage the findings of Brouwer and Haemers (2011) and Nazaret et al. (2023), which demonstrate that minimizing the largest eigenvalue $\lambda(\boldsymbol{J})$ ensures DAG soundness. Equation 5.5 outlines an augmented Lagrangian scheme with hyperparameters α and ρ, which are progressively increased during training.

$$h(\boldsymbol{J}) = |\lambda(\boldsymbol{J})|; \qquad \mathcal{L}_{\text{DAG}} = \|\boldsymbol{J}\|_f^2 + \rho h(\boldsymbol{J})^2 + \alpha h(\boldsymbol{J}) \tag{5.5}$$

Compositional Consistency: Similar to (Wiedemer et al., 2023), to ensure compositional generalisability, we use compositional consistency, which basically points towards a fixed point of ϕ, given by $\mathbf{s} = \phi(\mathbf{s})$. Which is achieved by maximising the posterior log probabilities for all possible random compositions.

Conjecture 5.1 (Compositional consistency helps in generalisation)**.** *Given the full support over $\mathcal{S}$ by observational data $\mathcal{X}$, maximising the posterior $q(\phi(\mathbf{s}))\forall \mathbf{s} \in \mathcal{S}$, while preserving high likelihood over $p(\mathbf{x})$ results in compositional generalisation of considered LCM. Which means for any random composition $\mathbf{s} \in \mathcal{S}$, $(f_c \circ f_d \circ g_z)(\mathbf{s})$ is optimised to be in-distribution of $p(\mathbf{x})$.*

$$\mathcal{L}_{\text{CC}} = -\mathbb{E}_{\mathbf{s}\in\mathcal{S}}\left[\log q(\phi(\mathbf{s}))\right] \tag{5.6}$$

Interpretation: As illustrated in Figure 5.7, $\mathcal{S}'$ is the inferred latent space of observational data capturing the possible interactions of objects $\mathbf{s}^1$ and $\mathbf{s}^2$ as reflected in observed data, while $\mathcal{S}$ demonstrates the space captured by all possible random compositions while preserving data *fidelity*.

Conjecture 5.2 (Identifiability of LCM)**.** *For any two models $\mathcal{M} = \langle \mathcal{X}, g_z, f_d, f_c, \mathcal{Y}\rangle$ and $\mathcal{M}' = \langle \mathcal{X}, g_z', f_d', f_c', \mathcal{Y}\rangle$, the following properties ensure their identifiability ($\mathcal{M} \sim \mathcal{M}'$), meaning $p_{\mathcal{M}}(\mathbf{x}, y) = p_{\mathcal{M}'}(\mathbf{x}, y)$:*

P.1: Identical observational spaces: $\mathcal{X}, \mathcal{Y}$ for both models,

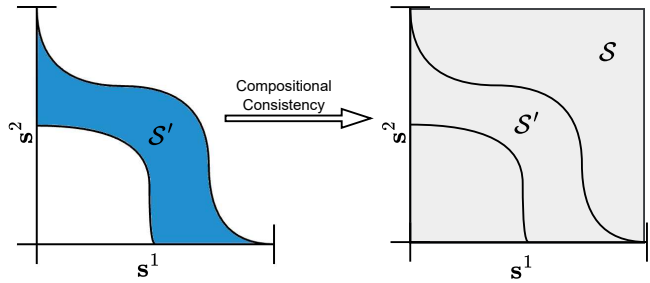


Figure 5.7: The figure demonstrates the generalisation of latent space as a result of optimising compositional consistency.

P.2: Latent distribution structure : $p_{\mathcal{M}}(\mathbf{z})$ *and* $p_{\mathcal{M}'}(\mathbf{z})$ *follow the structure defined by Equation 5.2,*

P.3: Weak injectivity: f_d *maps each* $\mathbf{z} \in \mathcal{Z}$ *uniquely to a ball of radius* $\delta > 0$ *in* $\mathcal{X}$*,*

P.4: Scalar latent variables: g_z *and* g'_z *are sets of scalar endogenous* $\mathbf{z}_i^v \in \mathbb{R}$ *and exogenous* $\mathbf{s}_i^v \in \mathbb{R}$ *variables,*

P.5: Regularizations: Constraints on compositional consistency (Equation 5.6), DAG structure (Equation 5.5), and mechanism consistency (Equation 5.4) are optimized.

Interpretation: The conjecture ensures identifiability by constraining LCMs: P.1 and P.2 ensures the same observation $(\mathcal{X}, \mathcal{Y})$ and model classes are considered across LCMs, P.3 ensures that the decoder f_d guarantees that every latent state $\mathbf{z}$ uniquely determines a region in the observational space $\mathcal{X}$, crucial for identifiability (Kivva et al., 2022), P.4 ensures that considered variables are scalar in nature, and finally, P.5 ensures all the regularisations are optimised, where compositional consistency and invariant mechanisms are essential for generalisation, making sure that support for object representation is full (Wiedemer et al., 2023) and encourages similar latent representations along with functions in both LCMs, DAG consistency enforces a valid latent graph structure by penalizing cycles, ensuring a sound SEM. Together, these properties ensure the LCM results in learning true compositional model.

Conjecture 5.3 ($J_{g_z^{-1}} \sim_{DAG} (I - C)$)**.** *In case when LCM is identifiable, we can show that Jacobian* $J_{g_z^{-1}}$ *is structurally equivalent to* $(I - C)$.

Interpretation: When an LCM is identifiable, the Jacobian of SEM is faithful to the compositional graph involved in data generation. This means we can use $I - C$ for invariant mechanism and DAG regularisations.

5.3 PUTTING THINGS IN CONTEXT

In this section, we link our theory to existing OCL and CRL works, highlighting their relations.

5.3.1 *Relation to Spatial Mixtures*

Considering a mixing coefficient π over a categorical environment variable y, we can restructure Equation 5.3 as a mixture of exponentials (Equation 5.7):

$$p(\mathbf{z}) \propto \sum_{p=1}^{|V|} \pi^p \exp\left(\bar{\phi}(\mathbf{z}^p, \mathbf{s}^p; y^p)\right) \quad \cdot \left[\prod_{q \in \mathcal{K}} \prod_{(i,j) \in |\mathcal{Z}^p| \times |\mathcal{Z}^q|} \exp\left(\lambda_{ij}^{pq} \phi_1(\mathbf{s}_i^p, \mathbf{z}_j^q)\right)\right] \tag{5.7}$$

Previous works such as Greff et al. (2017), Kori et al. (2024c), Yuan et al. (2024), and Burgess et al. (2019) have shown that, from a probabilistic perspective, the latent distribution in unsupervised object-discovery methods can be interpreted as spatial mixtures. Kori et al. (2024c) and Krimmel et al. (2024) demonstrate that more recent methods along slot-attention line of work can reduce the latent distribution to Gaussian Mixture Models (GMMs) or Von Mises-Fisher (vMF) distributions, depending on the chosen distance function. In our case, the model represents a generalized approach to spatial mixtures. When we assume that latent factors are statistically independent both within and across environments (i.e., $\lambda_{ij}^{pq} = 0$ for all (i, j, p, q)), the latent distribution $p(\mathbf{z})$ simplifies to a GMM, as shown in Equation 5.2. This distribution can be further extended to capture compositional graph structures and LCM, making it suitable for modelling more complex dependencies in the latent space.

WHAT DOES THIS MEAN? The latent distribution in Equation 5.7 generalizes traditional *spatial mixture models* by incorporating *pairwise dependencies* between object features. This demonstrates that providing a probabilistic interpretation of scene graphs can be achieved by considering pairwise interactions, where the term $\lambda_{ij}^{pq}\phi_1(\mathbf{s}_i^p, \mathbf{z}_j^q)$ can be interpreted as edges and functions in a scene graph, capturing both object- and attribute-level interactions.

5.3.2 *Relation to Invariance Theory*

The proposed formulation of invariant mechanisms can be understood through the lens of invariance principles in CRL, as introduced by Yao et al. (2024a). In the context of sample-level invariances (Von Kügelgen et al., 2021; Yao et al., 2023), consider two observed images $(\mathbf{x}^1, \mathbf{x}^2)$ that share underlying structural similarities due to transformations such as augmentations, changes in viewpoint, or pose shifts. By encoding these images into latent representations $\mathbf{z}^1 = g(\mathbf{x}^1)$ and $\mathbf{z}^2 = g(\mathbf{x}^2)$, the invariant CRL frameworks identifies a subset of invariant latent variables $\mathbf{c}^1, \mathbf{c}^2 \subseteq \mathcal{Z}$ that remain stable across both samples, such that for any $z \notin \mathbf{c}_1, \mathbf{c}_2$, we have $\frac{\partial h_1(\mathbf{c}_1)}{\partial z} = \frac{\partial h_2(\mathbf{c}_2)}{\partial z} = 0$. However, similar to observational grouping in Morioka and Hyvärinen (2023), we consider grouping of image $\mathbf{x}$ components by grouping variable y. Each partition corresponds to an object or component in the image, and by minimising mechanism consistency (Equation 5.4), ensures learning invariant object- and attribute-level interactions across scenes.

WHAT DOES THIS MEAN? The proposed formalisation can be reduced to invariant representation learning as in Yao et al. (2024a) by appropriately grouping observed data via categorical conditioning variable y. In essence, the approach unifies CRL and scene graph modelling by grounding scene understanding in invariant causal structures, by capturing object- and attribute-level interactions.



(a) **(b)** **(c)**

Figure 5.8: (a) "Two glasses, one with milk and the other with wine, both fully filled.", **(b)** "A laptop on top of a table.", and **(c)** "A table placed on top of a laptop." These examples illustrate generated samples from GPT-4o with highlighted predicates, demonstrating issues with VLMs and the need for our unified perspective.

5.4 WHY DOES IT MATTER?

As VLMs and foundation models continue to evolve, the need for structured, interpretable, and compositional representations becomes increasingly evident in achieving human-aligned, safe, fair, robust, and generalisable representations. While these models excel at capturing broad patterns from large datasets, their ability to generalise to novel, compositional, and causal reasoning tasks remains limited. Figure 5.8 highlights such challenges. In Figure 5.8(a), the mechanism of filling a glass should be consistent, independent of the specific instance (glass of milk vs glass of wine), yet the model fails to learn this invariant mechanism and instead learns instance-specific mechanisms. This demonstrates the necessity of the proposed perspective on invariant mechanisms that maintain causal consistency in a compositional setting.

Figures 5.8(b) and (c) further illustrate compositional inconsistencies. Both scenarios—*a laptop on a table* and *a table on a laptop*—are physically plausible and causally consistent. However, the model hallucinates in the latter case, failing to generalise in that scenario, highlighting the lack of compositional generalisability. This is crucial for ensuring out-of-distribution (OOD) generalisation capabilities, and accurate modelling of rare yet realistic events (Wiedemer et al., 2023). Building on these observations, we

outline the theoretical and practical implications of a probabilistic SGM perspective that unifies both CRL and OCL.

Transferability: The proposed causal abstraction provides a direction for developing *System-2* agents by bridging perception and abstract reasoning. This perspective suggests that by grounding dataset-level objects in scenes (Lake et al., 2017; Schölkopf et al., 2021; Kori et al., 2024b), models could potentially generalise across unseen scenes through invariant mechanisms. Such grounded representations facilitates task transferability and human-like reasoning (Tenenbaum et al., 2011), providing new avenues for integrating perception-driven neural networks with symbolic reasoning for planning and decision-making (Bengio et al., 2013a; Mao et al., 2019a; Proietti and Toni, 2023).

Robustness: This perspective aligns with applying causal inference principles, such as intervention and counterfactual reasoning, to scene modelling, as partially explored by Mansouri et al. (2022) and Mansouri et al. (2023). Models leveraging invariant mechanisms may capture object features even under partial observability, reducing the risk of hallucinations. Exploring this robustness is particularly promising for critical applications, such as autonomous driving (Schölkopf et al., 2021) and medical diagnosis (Castro et al., 2020).

Generalisation: Enforcing mechanism consistency across image partitions could lead to representations that are both causal and invariant across objects, improving OOD generalisation (Schölkopf et al., 2021). Additionally, compositional consistency enables model to generalise to novel environments in a causally consistent manner.

Counterfactual Reasoning and Controllability: Extending counterfactual reasoning in OCL through causal abstraction holds significant potential. Existing generative models (Kingma and Welling, 2013; Goodfellow et al., 2014; Mirza and Osindero, 2014; Karras, 2019; Song and Ermon, 2019) lack fine-grained control, often resulting in hallucinations or poor OOD generalisation (Reizinger and Krishnan, 2024). By formalising compositional graphs in the latent space, this perspective points towards new methods for controllable and interpretable generation, which extends to applications like image editing and embodied AI systems.

Despite these promising implications, several challenges remain to provide promising future direction:

Identifiability in Complex Scenes. Ensuring the identifiability of object-centric causal variables in complex, the current formalism provides an informal proof under explicitly made assumptions; rigorously establishing identifiability for general setting will be crucial for many use cases.

Mechanism Invariances. The proposed approach enforces invariances across image partitions, but extending these invariances to handle temporal dependencies is an interesting direction to model the physical properties of the system. For instance, considering the scene in Figure 5.1, if a torus is observed *pushing the cube* within a temporal sequence, this interaction reveals important properties such as the mass of objects and provides insight into friction within the environment. Such temporal extensions can improve the model's ability to infer *dynamical causal relationships*, making it more suitable for applications in physical reasoning, robotic manipulation, and scene understanding.

Systematic Evaluation. Given all the existing OCL datasets do not provide access to compositional or causal graphs that are used in generating particular scenes, it would be an important direction to define relevant metrics to evaluate these properties additionally, creating a detailed environment to generate synthetic/realistic scenes along with compositional structure would be very beneficial to the field in general. Some of the existing environments (Greff et al., 2022; Authors, 2024) can be modified for this purpose.

5.5 Conclusion & Discussion

The proposed a unifying perspective that integrates OCL and CRL through a probabilistic view SGM with *causal abstraction*. By conceptualizing object interactions as invariant mechanisms embedded within object-level graphs, we outlined a structured approach that captures both semantic and spatial dependencies, extending existing paradigms such as IF, NICA, and CRL. Here, we provide a perspective substantiated through rigorous conceptual arguments and demonstrative scenarios compelling reasoning for how this approach could mitigate key challenges in *foundation models*, including *hal-*

lucinations, lack of structured reasoning, and poor generalization. Specifically, by embedding causal reasoning within object-centricity, our perspective suggests that future AI systems can: 1) Enhance controllability and reliability in structured scene generation. 2) Improve generalization by aligning learned representations with real-world structured environments. 3) Enable compositional/counterfactual reasoning, allowing robust inference over novel interactions.

While our claims remain theoretical, we argue that formalizing composition as a causal abstraction presents a promising direction for developing AI systems that are interpretable, compositional, and robust. This perspective has broad implications for autonomous systems, robotics, and medical AI, where structured and generalizable scene understanding is critical, moving beyond pattern recognition to structured, human-aligned intelligence.

This page is intentionally left blank.

6

ARGUMENTATIVE VISUAL REASONING

"An argument is not a fight. It's a search for truth."

– Bo Bennett

Deep learning models are powerful image classifiers but their opacity hinders their trustworthiness. Explanation methods for capturing the reasoning process within these classifiers faithfully and in a clear manner are scarce, due to their sheer complexity and size. We provide a solution for this problem by defining a novel method for explaining the outputs of image classifiers with debates between two agents, each arguing for a particular class. We obtain these debates as concrete instances of *Free Argumentative eXchanges* (FAXs), a novel argumentation-based multi-agent framework allowing agents to internalise opinions by other agents differently than originally stated. We define two metrics (*consensus* and *persuasion rate*) to assess the usefulness of FAXs as argumentative explanations for image classifiers. We then conduct a number of empirical experiments showing that FAXs perform well along these metrics as well as being more faithful to the image classifiers than conventional, non-argumentative explanation methods.

6.1 MOTIVATION

With the increasing complexity and widespread deployment of deep learning models in our daily lives, the interpretation and explanation of these models' decisions have become a central focus in recent eXplainable Artificial Intelligence (XAI) literature (Ribeiro et al., 2016; Lundberg and Lee, 2017; Shrikumar et al., 2017). Many existing approaches for explaining image classification models heavily rely on heatmaps and segments to

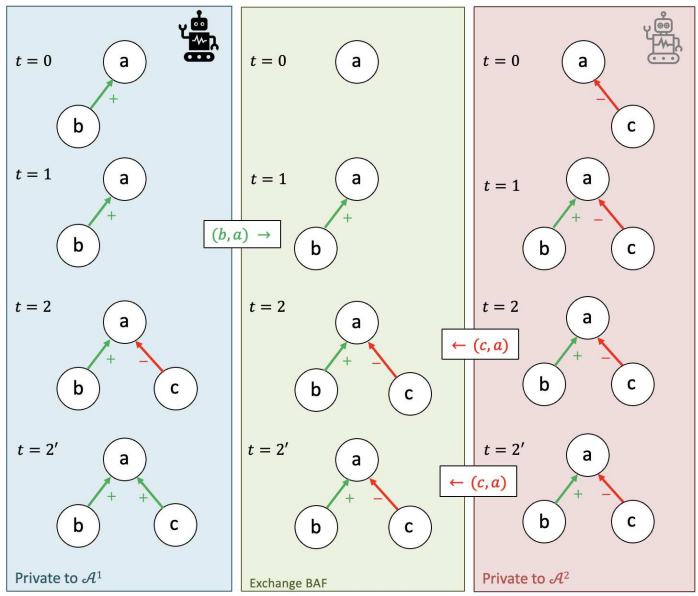


Figure 6.1: Illustration of FAXs (see Example 6.1).

localize regions of interest in input images that contribute to the model's output (Selvaraju et al., 2017; Chattopadhyay et al., 2018; Sattarzadeh et al., 2021), typically offering static input-output-based explanations while lacking deeper insights into the underlying model being explained. The literature has repeatedly highlighted the need for deeper and more dynamic explanations (Madumal et al., 2019; Miller, 2019; Lakkaraju et al., 2022), highlighting the input-output relationships of the model but also delving into its internal mechanisms and elucidating the model's reasoning. Also, there is an ongoing debate about the necessity of interactive explanations (Cawsey, 1991; Miller, 2019) and explanations that are contrastive and selected (Miller, 2019).

Dialogue-based explanations have been advocated as being useful in understanding the inner working of deep learning models (Lakkaraju et al., 2022). (Wang and Vasconcelos, 2019) argues that explanations are especially important when the model has high uncertainty about the output when the prediction oscillates between different classes resulting in different interpretations of the model behaviour. (Rago et al., 2023) proposes argumentative

exchanges to explain models via interactions amongst agents. Motivated by these varied lines of work, the main focus of this work is to extract explanations for image classifiers as debates between two artificial agents, arguing, in the spirit of bipolar argumentation (Cayrol and Lagasquie-Schiex, 2005), for and against the classifiers' outputs for given inputs.

Example 6.1 (Debate Intuition). *As an abstract illustration, consider two agents $\mathcal{A}^1$ and $\mathcal{A}^2$ as outlined in Figure 6.1 (black and grey, respectively). Each agent has an initial perception (at timestep $t = 0$) of their environment as a bipolar argumentation framework (BAF) about a topic of interest (a in the figure) which we consider as private. As agents start debating, they share their knowledge (see the exchange BAF figure 6.1) and may expand their private BAFs (e.g. $\mathcal{A}^2$ learns that b supports a at timestep $t = 1$ and agent $\mathcal{A}^1$ learns that c attacks a at timestep $t = 2$). As in human debates, as agents expand their knowledge they may see things differently from the other agents (e.g. at the alternative timestep $t = 2'$ $\mathcal{A}^1$ sees $\mathcal{A}^2$s's attack from c to a as a support).*

Overall in this chapter, we make the following contributions:

- we define a novel form of *free argumentative exchanges* (FAXs) to characterise explanations amongst agents as illustrated in Figure 6.1; differently from (Rago et al., 2023), these exchanges allow for agents to disagree on what constitutes an attack or support amongst arguments exchanged during debates (and are thus *free*); this technical novelty empowers the use of FAXs for explanation of image classifiers;
- we instantiate FAXs so that they can serve as the basis for explaining the outputs of image classifiers;
- we provide an implementation of the instantiated FAXs, by adapting the methodology of (Kori et al., 2024a) to allow agents to generate their own arguments;
- we evaluate our methodology and implementation quantitatively and qualitatively, with two types of image classifiers on two datasets; for the quantitative evaluation, we use two novel metrics to assess the argumentative quality of the generated debates, and, for comparison with baselines, two existing metrics (adapted to our setting) for ascertaining the faithfulness of explanations to the explained classifiers.

6.2 FAX:SET-UP

Consider a dataset $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^{h \times w \times c}$ represents a vector space with dimensions $h \times w$ corresponding to an image and c channels ($c \geq 1$). The label space $\mathcal{Y} = 1, \ldots, N$ consists of $N \geq 2$ classes. Following conventions in image classification (He et al., 2016; Huang et al., 2017), we consider an image classifier trained on $\mathcal{D}$, comprising a feature extractor $f : \mathcal{X} \to \mathcal{Z}$ and a feature classifier $g : \mathcal{Z} \to \mathcal{Y}$. The feature extractor f maps the observational space $\mathcal{X}$ to a continuous latent space $\mathcal{Z} \subseteq \mathbb{R}^{m \times d}$, where each element of $\mathcal{Z}$ represents a set of latent features with m elements, each of length d. Given an input $x \in \mathcal{X}$, the image classifier orders the classes in $\mathcal{Y}$ based on $g(f(x))$, predicting the top-class y as the output (with an abuse of notation we often write $g(f(x)) = y$). Consistent with prior works (Van Den Oord, Vinyals, et al., 2017; Kori et al., 2024a), we assume the existence of a set $\mathcal{Z}^i \subset \mathcal{Z}$ of features associated with each class $i \in \mathcal{Y}$. These are referred to as the *class-i-specific features*. Up to Section 6.5, we defer discussion on how these class-specific features are obtained.

6.3 FAX: AGENTS AND EXPLANATIONS

Agents are represented as *private triples for explananda*, a notion adapted from (Rago et al., 2023), as follows.

Definition 6.1 (Agent Explanandum)**.** *An agent $\mathcal{A}^i$ is a* private triple for an explanandum *e, amounting to $(\mathbb{I}^i, \mathbf{B}^i, \sigma^i)$ where:*

- $\mathbb{I}^i = \mathbb{I}^i_+ \cup \mathbb{I}^i_-$ *is an evaluation range, referred to as $\mathcal{A}^i$'s* private evaluation range*, where $\mathbb{I}^i_+$ (referred to as* positive evaluations*) and $\mathbb{I}^i_-$ (referred to as* negative evaluations*) are disjoint and for any $v_+ \in \mathbb{I}^i_+$ and $v_- \in \mathbb{I}^i_-$, $v_+ > v_-$;*
- $\mathbf{B}^i = \langle Args^i, \mathbf{A}^i, \mathbf{S}^i \rangle$ *is a BAF for e, referred to as $\mathcal{A}^i$'s* private BAF*;*
- σ^i *is an evaluation method, referred to as $\mathcal{A}^i$'s* private evaluation method*, such that, for any BAF $\mathbf{B} = \langle Args, \mathbf{A}, \mathbf{S} \rangle$ and, for any $\alpha \in Args$, $\sigma^i(\mathbf{B}, \alpha) \in \mathbb{I}^i$.*

Here, we use BAFs instead of quantitative BAFs (Baroni et al., 2018) as in (Rago et al., 2023), and rely upon evaluation ranges split into two, rather than three, partitions as in (Rago et al., 2023) (so we disregard the neutral partition in (Rago et al., 2023)). The threshold between the two partitions is seen as the point where an agent "changes their mind" on the explanandum, and may correspond to a classifier's decision boundary, as we will see later. We assume, for the remainder of this section, that any evaluation method is dialectically monotonic. We exemplify our notion of an agent in a simple, generic setting below (see Section 6.4 for instantiations for image classification).

Example 6.2 (Explanandum Example)**.** *An agent $\mathcal{A}^1$ is a private triple for an explanandum a, amounting to $(\mathbb{I}^1, \mathbf{B}^1, \sigma^1)$ where: $\mathbb{I}^1 = [0,1]$ with $\mathbb{I}^1_- = [0, 0.6[$ and $\mathbb{I}^1_+ = [0.6, 1]$; $\mathbf{B}^1 = \langle Args^1, \mathbf{A}^1, \mathbf{S}^1 \rangle$ such that $Args^1 = \{a, b\}$ with arguments a:* we should eat at this pizzeria *and b:* it is highly recommended, *$\mathbf{S}^1(a) = \{(b, a)\}$ and $\mathbf{A}^1(a) = \emptyset$; and σ^1 is some dialectically monotonic semantics, which in this case could give, for example $\sigma^1(\mathbf{B}^1, a) = 0.75$ and $\sigma^1(\mathbf{B}^1, b) = 0.5$ (given the asymmetric set-up of a's attackers and supporters). Here we can see that $\mathcal{A}^1$'s positive reasoning for the explanandum overcomes the (absent) negative reasoning against it, resulting in its strength being above the threshold of* 0.6 *and thus gives a positive evaluation.*

We define a novel form of *argumentation exchanges* amongst agents, which will serve, later, as explanations:[1]

Definition 6.2 (FAX)**.** *A* free argumentative exchange (FAX) for an explanandum e amongst agents $\mathcal{A}$, *where $|\mathcal{A}| = m \geq 2$ and each agent in $\mathcal{A}$ is a private triple for e, is a tuple: $\langle \mathbf{B}^x_0, \ldots, \mathbf{B}^x_n, \mathcal{A}_0, \ldots, \mathcal{A}_n, \mathcal{M} \rangle \quad (n \geq 0)$ where, for all timesteps $t \in [n]$, $\mathbf{B}^x_t$ (referred to as the* exchange BAF at step *t) is a BAF for e and $\mathcal{A}_t$ is a set of k agents, all private triples for e, such that:*

- $\mathbf{B}^x_0 = \langle Args^x_0, \mathbf{A}^x_0, \mathbf{S}^x_0 \rangle$ *is a BAF such that:*
  - $Args^x_0 = \{e\}$; $\mathbf{A}^x_0 = \emptyset$; $\mathbf{S}^x_0 = \emptyset$;
- $\mathcal{A}_0 = \mathcal{A}$;

and, at timestep $t \in]n]$, letting $\mathbf{B}^x_* = \langle Args^x_*, \mathbf{A}^x_*, \mathbf{S}^x_* \rangle = \mathbf{B}^x_t \setminus \mathbf{B}^x_{t-1}$:

1 Throughout, we denote with $[k]$ the set $\{0, 1, \ldots, k\}$, with $]k]\{1, \ldots, k\}$, etc.

- $\mathbf{B}^x_t \sqsupseteq \mathbf{B}^x_{t-1}$, *where* $\forall (a,b) \in \mathbf{A}^x_*$, $\exists j \in]m]$ *such that* $(a,b) \in \mathbf{A}^j_{t-1}$ *and* $\forall (c,d) \in \mathbf{S}^x_*$, $\exists k \in]m]$ *such that* $(c,d) \in \mathbf{S}^k_{t-1}$;
- $\mathcal{A}_t$ *is a set of private triples* $(\mathbb{I}^i, \mathbf{B}^i_t, \sigma^i)$ *for* e*, one for each* $i \in]m]$*, where* $\mathbf{B}^i_t \sqsupseteq \mathbf{B}^i_{t-1}$ *and* $Args^x_* \subseteq Args^i_t$ *and* $\mathbf{A}^x_* \cup \mathbf{S}^x_* \subseteq \mathbf{A}^i_t \cup \mathbf{S}^i_t$;

$\mathcal{M}$ *is the* contributor mapping*, such that, for* $\mathbf{B}^x_n = \langle Args^x_n, \mathbf{A}^x_n, \mathbf{S}^x_n \rangle$*, for every* $(\alpha, \beta) \in \mathbf{A}^x_n \cup \mathbf{S}^x_n$*:* $\mathcal{M}((\alpha,\beta)) = (i,t)$ *with* $i \in]m]$ *and* $t \in]n]$.

FAXs thus allow agents to add attacks or supports from their private to the exchange BAF, which are then incorporated to all other agents' frameworks as some form of relation. Note that it is only the BAFs which change with the timestep, not, for example, the evaluation ranges or methods. Intuitively, the contributor mapping returns, for each attack/support pair in the final exchange BAF, the agent who contributes the pair as well as the timestep at which the pair was contributed. Note that, by defining $\mathcal{M}$ as a mapping, we impose that there is a single contributor and timestep for each attack/support pair in the final exchange BAF, and therefore pairs cannot be introduced multiple times in FAXs. Note that FAXs are variants of argumentative exchanges (AXs) in (Rago et al., 2023): whereas in AXs agents are equipped with quantitative BAFs, in FAXs they are equipped with BAFs; and, whereas in AXs agents are assumed to share a *lingua franca* of arguments' attackers and supporters (so that if an argument attacks or supports another for an agent it does so for all others), in FAXs attackers/supporters for an agent may be supporters/attackers for another, witnessing the 'free" nature of FAXs. Next, we illustrate (in the earlier simple, generic setting) why this may be useful.

Example 6.3. *Continuing from Example 6.2, a second agent* $\mathcal{A}^2$ *is a private triple for* a*, amounting to* $(\mathbb{I}^2, \mathbf{B}^2, \sigma^2)$ *where:* $\mathbb{I}^2 = \mathbb{I}^1$; $\mathbf{B}^2 = \langle Args^2, \mathbf{A}^2, \mathbf{S}^2 \rangle$ *such that* $Args^2 = \{a, c\}$ *with argument* c*:* there is pineapple on the pizza, $\mathbf{A}^2(a) = \{(c,a)\}$ *and* $\mathbf{S}^2(a) = \emptyset$*; and* σ^2 *is some dialectically monotonic semantics, which in this case could give, e.g.,* $\sigma^2(\mathbf{B}^2, a) = 0.25$ *and* $\sigma^2(\mathbf{B}^2, c) = 0.5$ *(again given the asymmetric set-up of* a*'s attackers and supporters). Here* $\mathcal{A}^2$*'s negative reasoning for the explanandum overcomes the (absent) positive reasoning against it, resulting in its strength being below the threshold of* 0.6 *and thus gives a negative evaluation. A FAX for a amongst agents* $\mathcal{A} = \{\mathcal{A}^1, \mathcal{A}^2\}$ *is then a tuple* $\langle \mathbf{B}^x_0, \mathbf{B}^x_1, \mathbf{B}^x_2, \mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \mathcal{M} \rangle$ *such that:*

- $\mathbf{B}_1^x \setminus \mathbf{B}_0^x = \langle\{b\}, \emptyset, \{(b,a)\}\rangle$, *where* $\mathcal{M}((b,a)) = (1,1)$, *and thus* $\mathcal{A}_1 \neq \mathcal{A}_0$, *where* $\mathcal{A}_1^1 = \mathcal{A}_0^1$ *and* $\mathcal{A}_1^2 \neq \mathcal{A}_0^2$ *with* $\mathbf{B}_1^2 \setminus \mathbf{B}_0^2 = \langle\{b\}, \emptyset, \{(b,a)\}\rangle$;
- $\mathbf{B}_2^x \setminus \mathbf{B}_1^x = \langle\{c\}, \{(c,a)\}, \emptyset\rangle$, *where* $\mathcal{M}((c,a)) = (2,2)$, *and thus* $\mathcal{A}_2 \neq \mathcal{A}_1$, *where* $\mathcal{A}_2^2 = \mathcal{A}_1^2$ *and* $\mathcal{A}_2^1 \neq \mathcal{A}_1^1$ *with* $\mathbf{B}_2^1 \setminus \mathbf{B}_1^1 = \langle\{c\}, \emptyset, \{(c,a)\}\rangle$.

Here, support (b,a) *provided at timestep* 2 *by* $\mathcal{A}^1$ *is learnt by* $\mathcal{A}^2$ *(as is conventional in AXs (Rago et al., 2023)) as a support, indicating that the agents agree on the relation. Then, at timestep* 2, $\mathcal{A}^2$ *provides the attack* (c,a), *e.g. because pineapple on a pizza is anathema in Italy. However,* $\mathcal{A}^1$ *learns the relation* (c,a) *as a support, indicating that they considered* c *to be providing reasoning for* a, *e.g. because* $\mathcal{A}^1$ *likes pineapple on pizza. This shows how FAXs, differently to AXs, allow for differences in the way agents interpret relations. Given that* σ^1 *and* σ^2 *are dialectically monotonic, we know that* $\sigma^1(\mathbf{B}_0^1, \alpha) = \sigma^1(\mathbf{B}_1^1, \alpha) \leq \sigma^1(\mathbf{B}_2^1, \alpha)$ *and* $\sigma^2(\mathbf{B}_0^2, \alpha) \leq \sigma^2(\mathbf{B}_1^2, \alpha) = \sigma^2(\mathbf{B}_2^2, \alpha)$, *respectively.*

We can define a notion to restrict FAXs so that agents therein can be deemed to share a lingua franca, as follows.

Definition 6.3 (Lingua Franca)**.** *Given a FAX* F *for* e *amongsts* $\mathcal{A}$, *with* $F = \langle \mathbf{B}_0^x, \ldots, \mathbf{B}_n^x, \mathcal{A}_0, \ldots, \mathcal{A}_n, \mathcal{M}\rangle$, $\mathcal{A}_n^i \in \mathcal{A}_n$ *such that* $\mathcal{A}_n^i = (\mathbb{I}^i, \mathbf{B}_n^i, \sigma^i)$ *and* $\mathbf{B}_n^i = \langle Args_n^i, \mathbf{A}_n^i, \mathbf{S}_n^i\rangle$, *we say that* F *has an* effective lingua franca *iff* $(\bigcup_{\mathcal{A}_n^i \in \mathcal{A}_n} \mathbf{A}_n^i) \cap (\bigcup_{\mathcal{A}_n^i \in \mathcal{A}_n} \mathbf{S}_n^i) = \emptyset$.

It can be seen that the FAX in Example 6.3 does not have a lingua franca since the agents disagree on the relation (c,a). Typically, FAXs begin because there is a *conflict* between agents, amounting to a different *stance* on the explanandum.

Definition 6.4 (Stance)**.** *Given an agent* $\mathcal{A}^i$, *i.e. a private triple* $(\mathbb{I}^i, \mathbf{B}^i, \sigma^i)$ *(for some explanandum* e*) with* $\mathbf{B}^i = \langle Args^i, \mathbf{A}^i, \mathbf{S}^i\rangle$, *for any* $\alpha \in Args^i$, *let* $\mathcal{A}^i$*'s* stance on α *be defined as* $\Sigma^i(\mathbf{B}^i, \alpha) = +$ *(*positive stance*) iff* $\sigma^i(\mathbf{B}^i, \alpha) \in \mathbb{I}_+^i$, *and* $\Sigma^i(\mathbf{B}^i, \alpha) = -$ *(*negative stance*) otherwise. Then, a set* $\mathcal{A}$ *of agents/ private triples for explanandum* e *is* in conflict wrt e *iff there are two or more agents in* $\mathcal{A}$ *with different stances on* e.

Note that this notion of stance is adapted from (Rago et al., 2023), ignoring the neutral stance therein. We see FAXs as means to lead to resolution of initial conflicts, by allowing agents to argue while identifying and filling

any gaps in their beliefs (represented by their private BAFs). We adapt the following from (Rago et al., 2023), to characterise FAXs successfully leading to resolution (or not), from an initial conflict.

Definition 6.5 (Resolution). *Let $\mathcal{A}$ be a set of agents/private triples for explanandum e. Let $\mathcal{A}$ be in conflict wrt e. Let $F = \langle \mathbf{B}_0^x, \ldots, \mathbf{B}_n^x, \mathcal{A}_0, \ldots, \mathcal{A}_n, \mathcal{M} \rangle$ be a FAX for e amongst $\mathcal{A}$. Then:*

- *F is* unresolved at timestep *t, for $t \in]n]$, iff $\mathcal{A}_t$ is in conflict wrt e, and* resolved at *t otherwise;*
- *F is* unresolved *iff it is unresolved at every timestep $t \in]n]$;*
- *F is* resolved *iff it is resolved at timestep n.*

Example 6.4. *Continuing from Example 6.3, we know that $\sigma^1(\mathbf{B}_0^1, \alpha) \in \mathbb{I}_+^1$ and $\sigma^2(\mathbf{B}_0^2, \alpha) \in \mathbb{I}_-^2$, and so $\Sigma^1(\mathbf{B}_0^1, \alpha) = +$ $\Sigma^2(\mathbf{B}_0^2, \alpha) = -$, meaning $\mathcal{A}$ is in conflict wrt α. At timestep 1, let us assume that, although $\sigma^2(\mathbf{B}_0^2, \alpha) \leq \sigma^2(\mathbf{B}_1^2, \alpha)$, it remains the case that $\sigma^2(\mathbf{B}_1^2, \alpha) \in \mathbb{I}_-^2$. Since $\sigma^1(\mathbf{B}_0^1, \alpha) = \sigma^1(\mathbf{B}_1^1, \alpha) \in \mathbb{I}_+^1$, we can see that the FAX is unresolved at timestep 1. Then, since $\sigma^1(\mathbf{B}_1^1, \alpha) \leq \sigma^1(\mathbf{B}_2^1, \alpha)$, thus $\sigma^1(\mathbf{B}_2^1, \alpha) \in \mathbb{I}_+^1$ and $\sigma^2(\mathbf{B}_1^2, \alpha) = \sigma^2(\mathbf{B}_2^2, \alpha) \in \mathbb{I}_-^1$, we know that $\Sigma^1(\mathbf{B}_2^1, \alpha) = +$, $\Sigma^2(\mathbf{B}_2^2, \alpha) = -$ and thus the FAX is unresolved. Meanwhile, if $\mathcal{A}^1$ had interpreted (γ, α) as an attack, let us say at an alternate $t = 2'$, since σ^1 is dialectically monotonic we know that it would have been the case that $\sigma^1(\mathbf{B}_1^1, \alpha) \geq \sigma^1(\mathbf{B}_2^1, \alpha)$, and the FAX may have been resolved if $\sigma^1(\mathbf{B}_{2'}^1, \alpha) \in \mathbb{I}_-^1$, thus giving $\Sigma^1(\mathbf{B}_{2'}^1, \alpha) = \Sigma^2(\mathbf{B}_{2'}^2, \alpha) = -$.*

When FAXs are used for conflict resolution, exchange BAFs therein can be seen as *explanations*, in that they unearth the reasoning behind the resolution or otherwise of the conflict amongst the agents, with evidence that the explanandum is "correct" or not (when the FAX is resolved, depending on the final stance of all agents) or why it cannot be deemed "correct" or otherwise (when the FAX is unresolved). For illustration, in the first FAX ($t = 2$) in Example 6.4, the agents do not share the same stance on the explanandum due to their differing interpretations of (γ, α), whereas in the second FAX ($t = 2'$), they agree on both this relation being an attack and on their final stances on the explanandum.

When using FAX for explaining image classification (from Section 6.4), we will restrict attention to special forms of FAXs, as follows.

Definition 6.6 (Strictly Interleaved FAX)**.** *A* strictly interleaved FAX *for e amongst $\mathcal{A}$ is a FAX*
$\langle \mathbf{B}^x_0, \ldots, \mathbf{B}^x_n, \mathcal{A}_0, \ldots, \mathcal{A}_n, \mathcal{M} \rangle$ for e amongst $\mathcal{A}$ such that

- *$\mathcal{A} = \{\mathcal{A}^1, \mathcal{A}^2\}$ (i.e. with $|\mathcal{A}| = 2$);*
- *for each timestep $t \in]n]$ there exists exactly one (α, β) such that $\mathcal{M}((\alpha, \beta)) = (k, t)$ and if t is odd then $k = 1$; else $k = 2$.*

If n is even, then the FAX is equal opportunity.

The FAX in Example 6.3 is equal opportunity strictly interleaved. Intuitively, in a strictly interleaved FAX agents take turns, contributing one attack/support at a time and making the same number of contributions, to have the same chances at persuading the other.

6.4 FAX EXPLANATIONS FOR IMAGE CLASSIFICATION

We see explanations for image classification as (exchange BAFs of) equal opportunity strictly interleaved FAXs amongst:

- $\mathcal{A}^1$ (the *proponent*, arguing for the class in $\mathcal{Y}$ predicted by the underlying image classifier for input image $x \in \mathcal{X}$ of interest), and
- $\mathcal{A}^2$ (the *opponent*, arguing against the predicted class).

For simplicity, we restrict attention to the top two classes in the ordering determined by the image classifier g, for the input image, so that $\mathcal{A}^2$ argues against the predicted (top) class by arguing for the second best in the ordering. In the remainder of the paper we assume, without loss of generality, that class 1 is the predicted class for x and class 2 is the second best. In line with the machine learning literature, we also refer to class 1 as y.

In our approach to explaining image classification using FAXs, each agent argues for a particular class, and thus class-specific features play the role of arguments exchanged between agents. The arguments put forward by the proponent and opponent can be seen as playing the role of positive and negative, respectively, feature attributions as in some XAI literature (Ribeiro

et al., 2016; Lundberg and Lee, 2017; Selvaraju et al., 2017; Shrikumar et al., 2017). However, intuitively, the exchange BAF in the FAX can also convey the reasoning of the classifier. We will provide an evaluation of FAXs as explanations of image classifiers in Section 6.7. Here, we focus on instantiating the generic FAXs for our purposes.

Concretely, we assume that the two agents explaining the output of an underlying image classifier by virtue of a FAX can leverage on *class-specific classifiers* $q^1 : 2^{\mathcal{Z}^1 \cup \mathcal{Z}^2} \to [0,1]$ and $q^2 : 2^{\mathcal{Z}^1 \cup \mathcal{Z}^2} \to [0,1]$, allowing the agents to evaluate sets of their own and other agents' arguments (in other words, amounting to their private evaluation methods). Until Section 6.5, we ignore how these class-specific classifiers can be learnt, alongside the class-specific features, so that they are faithful to the original image classifier being explained (see Section 6.8.4), which is crucial to guarantee that FAXs as explanations are also faithful. We also refer to the class-specific classifiers as *private classifiers* (with $\mathcal{Z}^1$ and $\mathcal{Z}^2$ also referred to as *private features*). Finally, given that we aim at explaining the output of the image classifier, we use (x, y) as the *explanandum* e.

We now instantiate the general Definition 6.1, describing agents, to capture proponent and opponent for image classification.

Definition 6.7 (Image Classification BAF)**.** *Let $i \in \{1,2\}$. Then, the* initial image classification agent *is a private triple $(\mathbb{I}^i, \mathbf{B}^i, \sigma^i)$ for (x, y) such that:*

- $\mathbb{I}^i = [0,1]$ *is the* agent's evaluation range, *with $\mathbb{I}^i_- = [0, \tau)$ and $\mathbb{I}^i_+ = [\tau, 1]$, for some* threshold $\tau \in [0,1]$;
- $\mathbf{B}^i = \langle Args^i, \mathbf{A}^i, \mathbf{S}^i \rangle$ *is a BAF where if $i = 1$ then*
  - $Args^i \subseteq \{(x,y)\} \cup \mathcal{Z}^1$ *such that $(x,y) \in Args^i$,*
  - $\mathbf{A}^i = \emptyset$,
  - $\mathbf{S}^i \subseteq \mathcal{Z}^1 \times \{(x,y)\}$;

  and if $i = 2$ then
  - $Args^i \subseteq \{(x,y)\} \cup \mathcal{Z}^2$ *such that $(x,y) \in Args^i$,*
  - $\mathbf{A}^i \subseteq \mathcal{Z}^2 \times \{(x,y)\}$,
  - $\mathbf{S}^i = \emptyset$;
- $\sigma^i : Args^i \to \mathbb{I}^i$ *is such that:*

– $\sigma^i(\mathbf{B}^i,(x,y)) = q^i(\mathbf{A}^i((x,y)) \cup \mathbf{S}^i((x,y)))$;
– *for* $z \in Args^i \setminus \{(x,y)\}$, $\sigma^i(\mathbf{B}^i,z) = q^i(\{z\})$.

Corollary 6.1 (Conflict). *If* $\sigma^1(\mathbf{B}^1,(x,y)) \neq \sigma^2(\mathbf{B}^2,(x,y))$, *i.e.* $q^1(\mathbf{S}^1((x,y))) \neq q^2(\mathbf{A}^2((x,y)))$, *then* $\exists\tau \in [0,1]$ *such that* $\mathcal{A}$ *is in conflict.*

Here, we use specialised notions compared to those in Definition 6.1. Specifically, the private BAFs are "shallow" and acyclic, with all attacks and supports from private features (of either agent) to the explanandum. The evaluation ranges are divided into positive and negative evaluations by a threshold τ, which, as Lemma 6.1 demonstrates, can be guaranteed to provide a dividing line between the two classes if the agents' class-specific features have an effect on the private classifiers' outputs. The evaluation method is defined in terms of the class-specific classifier of the agent: the evaluation of the explanandum is given by the classifier applied to its attackers and supporters (which are class-specific features), while the evaluation of a class-specific feature is given by the private classifier applied to this feature only.

To obtain explanations for image classifiers, we will use FAXs where agents update their private BAFs guided by their private classifiers, using the following "learning strategy".

Definition 6.8 (Monotonicity). *Let* $\langle \mathbf{B}^x_0,\ldots,\mathbf{B}^x_n,\mathcal{A}_0,\ldots,\mathcal{A}_n,\mathcal{M}\rangle$ *be an equal opportunity strictly interleaved FAX for explanandum* (x,y) *amongst initial image classification agents* $\mathcal{A} = \{\mathcal{A}^1,\mathcal{A}^2\}$. *Then, for* $i,j \in \{1,2\}$ *with* $i \neq j$, $\mathcal{A}^i$ *adopts a* dialectically monotonic learning strategy *iff at timestep* $t \in]n]$, *if* $\exists(z,(x,y))$ *such that* $\mathcal{M}((z,(x,y))) = (i,t)$, *then* $\mathbf{B}^i_t = \mathbf{B}^i_{t-1}$; *otherwise* $\mathcal{M}((z,(x,y))) = (j,t)$ *and* $\mathbf{B}^i_t = \langle Args^i_t, \mathbf{A}^i_t, \mathbf{S}^i_t\rangle$ *is such that:*

- $(z,(x,y)) \in \mathbf{A}^i_t \setminus \mathbf{A}^i_{t-1}$ *iff* $q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y)) \cup \{z\}) - q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y))) < 0$;
- $(z,(x,y)) \in \mathbf{S}^i_t \setminus \mathbf{S}^i_{t-1}$ *iff* $q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y)) \cup \{z\}) - q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y))) \geq 0$;

and σ^i *is such that:*

- $\sigma^i(\mathbf{B}^i_t,(x,y)) = q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y)) \cup \{z\})$;

- $\forall z' \in Args^i_t \cap Args^i_{t-1}$, $\sigma^i(\mathbf{B}^i_t, z') = \sigma^i(\mathbf{B}^i_{t-1}, z')$; *and*
- $\sigma^i(\mathbf{B}^i_t, z) = |q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y)) \cup \{z\}) - q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y)))|$.

In the remainder, we refer to equal opportunity strictly interleaved FAXs where both agents adopt a dialectically monotonic learning strategy simply as FAXs. Note that, in FAXs, the initial private BAF of each agent contains all arguments representing the agent's private features. Then, any arguments learned from the other agent's contributions are such that their addition to the agent's private BAF results in dialectically monotonic behaviour (due to the agent's characterisation of the relation as an attacker or supporter), in the spirit of (Rago et al., 2022). This naturally leads to:

Proposition 6.1 (Dialectically Monotonicity)**.** *In any FAX for an explanandum (x,y) amongst agents $\mathcal{A} = \{\mathcal{A}^1, \mathcal{A}^2\}$, for any $\mathcal{A}^i \in \mathcal{A}$, σ^i is dialectically monotonic.*

Proof. From Definition 6.7, for any $\mathcal{A}^i \in \mathcal{A}$ and any timestep t, $\mathbf{B}^i_t$ is flat, i.e. it consists of the explanandum (x,y) and its attackers or supporters. It can then be seen from Definition 6.8 that the strength of any argument $a \in Args^i \setminus \{(x,y)\}$ does not change with the timestep (its attackers and supporters remain empty throughout), i.e. for any timesteps $t > 0$, it is necessarily the case that $\sigma^i(\mathbf{B}^i_t, a) = \sigma^i(\mathbf{B}^i_{t-1}, a)$. Thus, the last bullet of the definition of dialectical monotonicity is satisfied. Meanwhile, the first bullet can only apply to arguments added which attack or support (x,y), since this is the only argument which is attacked or supported, by Definition 6.7. We can see from Definition 6.8 that for any t such that there exists some $(z,(x,y)) \in \mathbf{A}^i_t \setminus \mathbf{A}^i_{t-1}$, it is necessarily the case that $q^i(\mathbf{A}^i_t((x,y)) \cup \mathbf{S}^i_t((x,y))) < q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y)))$, and thus, by Definition 6.7, $\sigma^i(\mathbf{B}^i_t, (x,y)) < \sigma^i(\mathbf{B}^i_{t-1}, (x,y))$. Again from Definition 6.8, we can see that for any t such that there exists some $(z,(x,y)) \in \mathbf{S}^i_t \setminus \mathbf{S}^i_{t-1}$, it is necessarily the case that $q^i(\mathbf{A}^i_t((x,y)) \cup \mathbf{S}^i_t((x,y))) \geq q^i(\mathbf{A}^i_{t-1}((x,y)) \cup \mathbf{S}^i_{t-1}((x,y)))$, and therefore, by Definition 6.7, $\sigma^i(\mathbf{B}^i_t, (x,y)) \geq \sigma^i(\mathbf{B}^i_{t-1}, (x,y))$. Thus, the property is satisfied in both cases for the first bullet, and dialectical monotonicity is satisfied overall. □

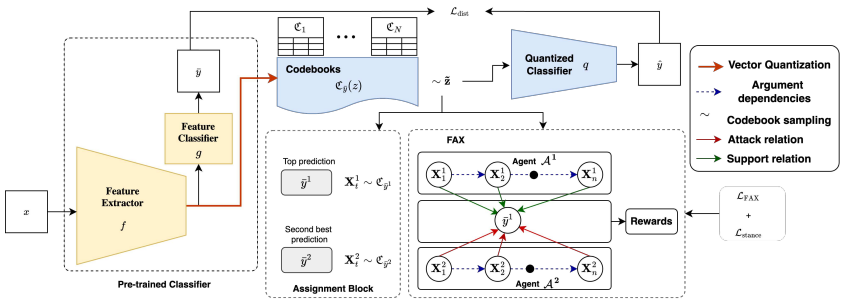


Figure 6.2: Overview of our implementation (argument dependencies are temporal).

This result sanctions that our choices in instantiating the semantics for FAXs leads to explanations that are dialectically monotonic, which has been identified as an important property by argumentation practitioners (Amgoud and Ben-Naim, 2018; Baroni et al., 2019; Potyka, 2021) and found to be intuitive by humans.

6.5 FAX: IMPLEMENTATION

In this section, we detail our methodology (overviewed in Figure 6.2) for obtaining and evaluating empirically FAXs with data for image classification. Specifically, we detail how class-specific (private) features can be obtained (Section 6.5.1); and how class-specific agents (including their private classifiers and their policies for contributing to FAXs) are learnt and deployed (Section 6.5.2). Note that, while FAXs are amongst two agents only (for the top- and second-best predicted classes by the classifier), given that different inputs will result in different predictions we need to develop, at training time, all agents, with their private features and classifiers.

6.5.1 *Class-specific discrete features*

We obtain these by simultaneously training, similarly to (Kori et al., 2024a):

- *N codebooks* $\mathfrak{C}_1, \ldots, \mathfrak{C}_N$ (one per class in $\mathcal{Y}$); for each $i \in \{1, \ldots, N\}$, $\mathfrak{C}_i \in \mathbb{R}^{\tilde{n} \times d}$ corresponds to $\mathcal{Z}^i$;[2] for $z = f(x)$, if $\bar{y}$ is the top-class predicted by $g(z)$, we use $\mathfrak{C}_{\bar{y}}(z)$ to stand for the specific discrete features in $\mathfrak{C}_{\bar{y}}$ corresponding to z (we use $\tilde{z}$ to stand for $\mathfrak{C}_{\bar{y}}(z)$ if clear);
- a *quantized classifier* $q : \mathfrak{C} \to \mathcal{Y}$, for $\mathfrak{C} = \mathfrak{C}_1 \cup \ldots \cup \mathfrak{C}_N$, distilling the knowledge of the feature classifier g so that $q(\tilde{z})$ approximates $g(z)$.

Intuitively, a codebook is a collection of averaged patterns or "concept representations" that summarize the key features of data (Van Den Oord, Vinyals, et al., 2017).

TRAINING. To get the codebooks, we draw inspiration from (Van Den Oord, Vinyals, et al., 2017). Intuitively, for each $(x, y) \in \mathcal{D}$, with $z = f(x)$, for $\bar{y} = \texttt{argmax}(g(z))$ (i.e. $\bar{y}$ is the top-class predicted by the feature classifier g for z), we aim at deterministically mapping the elements of z to the nearest elements of $\mathfrak{C}_{\bar{y}}$ using some convex distance function δ, as follows:

$$\tilde{z} = \{\arg\min_{\tilde{k} \in \mathfrak{C}_{\bar{y}}} \delta(z_k, \tilde{k}) | z_k \in z\} \tag{6.1}$$

To learn this in an end-to-end fashion, we use the Gumble sampling procedure from (Jang et al., 2016), resulting in $\tilde{z}$ as a projection of the continuous features in z onto every element of codebook $\mathfrak{C}_{\bar{y}}$, corresponding to pairwise similarity scores between z and all $\tilde{n}$ codebook features $\mathfrak{C}_{\bar{y}}$. The resulting quantization objective for training is described as follows (see (Jang et al., 2016) for details):

$$\mathcal{L}_{\text{quant}} = \sum \texttt{softmax}(\tilde{z}) \, (\texttt{logsoftmax}(\tilde{z})) \tag{6.2}$$

To faithfully learn the quantized classifier q, we adopt the following *distillation* loss during training, where $\bar{y}$ is the predicted class (by g) for input x and CE correspond to cross-entropy loss:

$$\mathcal{L}_{\text{dist}} = \mathcal{L}_{\text{quant}} + \text{CE}\,(q(\tilde{z})), \bar{y}) \tag{6.3}$$

2 We assume for simplicity that $M^i = \tilde{n}$ for all $i \in \{1, \ldots, N\}$, namely all agents have the same number of class-specific features.

By using this loss, we strive towards a faithful q to the classifier.

6.5.2 *Class-specific agents*

We obtain N agents, each of which is responsible for arguing wrt a particular class in $\mathcal{Y}$. For any given input x, an instance of FAX is obtained between the two agents whose class is the top-class predicted by $g(f(x))$ (for the proponent) and the second-best class (for the opponent). Specifically, if $g(f(x)) = [\bar{y}^1, \ldots, \bar{y}^N]$, then the proponent is $\mathcal{A}^{\bar{y}^1}$ and the opponent is $\mathcal{A}^{\bar{y}^2}$, as depicted in the assignment block in Figure 6.2 (the assignment block selects the agents based on the estimated top two classes).

We model each agent $\mathcal{A}^i$ (arguing for class $\bar{y}^i$) as a *sequence model* ζ^i. In line with (Kori et al., 2024a), we use gated recurrent units (GRUs) for realizing these sequence models. The sequence model ζ^i operates on a hidden state vector (h^i_{t-1}), which can be treated as a proxy interpretation for information drawn from the exchange BAF in the FAX till the current timestep (t). The sequence model uses a private modulator network $\mathcal{M}^i$ which encodes arguments (in $Args^i_t$ in the agent's private BAF $\mathbf{B}^i_t$) to update the hidden state representation:

$$h^i_t = \zeta^i(h^i_{t-1}, \mathcal{M}^i(Args^i_t)) \tag{6.4}$$

The output (hidden state vector) of ζ^i is then used to determine a *policy function* Π^i, for determining how agents contribute attacks and supports (in the agent's private BAF $\mathbf{B}^i_t$) to FAXs, by determining which argument these attacks (if the agent is the opponent $\mathcal{A}^j$) or supports (if the agent is the proponent $\mathcal{A}^i$) are drawn from:

$$Args^i_t \sim \Pi^i(h^i_{t-1} \mid Args^i_{t-1} \cup Args^j_{t-1}, \bar{y}^i) \tag{6.5}$$

Note that these arguments are from $\mathfrak{C}_{\bar{y}^i}$, corresponding to private features in $\mathcal{Z}^{\bar{y}^i}$. The output (hidden state vector) of ζ^i is also used to obtain the *private classifier* q^i, as a multi-layer perceptron with hidden state vectors as inputs, associating them to class confidence in turn used to assign values to sets of private features as in Section 6.4.

TRAINING. To obtain each agent's sequence model in an end-to-end fashion, we adapt the REINFORCE learning algorithm from (Mnih et al., 2014). Specifically, we see the next argument prediction/selection as a reinforcement learning task, with agents' rewards as follows.

Definition 6.9 (Reward). *Let* $\langle \mathbf{B}_0^x, \ldots, \mathbf{B}_n^x, \mathcal{A}_0, \ldots, \mathcal{A}_n, \mathcal{M} \rangle$ *be a FAX for explanandum* (x, y) *amongst agents* $\mathcal{A} = \{\mathcal{A}^1, \mathcal{A}^2\}$. *Then, for* $i \in \{1, 2\}$ *and* $t \in \{1, \ldots, n\}$, $\mathcal{A}^i$*'s* reward *at timestep* t *is* $r_t^i = \Sigma^i(\mathbf{B}_0^i, e)\sigma^i(\mathbf{B}_t^i, e)$.[3]

Thus, reward is a continuous-valued function modelled using the agent's "confidence" towards the explanandum, and reflecting the contributed arguments to date and the agent's original stance towards the explanandum. Note that this stance is always positive for the first agent and negative for the second, given our choice of τ in Section 6.4. Note also that rewards are between $[-1, 1]$. We combine the agent's reward with REINFORCE gradients:

$$\mathcal{L}_{\text{FAX}}^i = -\sum_t \log \Pi_{\theta^i}^i(h_t^i \mid Args_{t-1}^i \cup Args_{t-1}^j, \bar{y}^i)(r_t^i - b_t^i) \tag{6.6}$$

where θ^i are the parameters of policy Π^i (and thus modulator $\mathcal{M}^i$) being learnt and b_t^i is a baseline value estimated by $\mathcal{A}^i$ at timstep t, which is mainly used to reduce the variance in the agent's behaviour during the exploration stage in (reinforcement) learning. Minimisation of this loss can also be viewed as maximisation of log-likelihood of the policy distribution (Mnih et al., 2014).

Finally, to encourage agents to argue for a particular class ($\bar{y}^1$ for the proponent and $\bar{y}^2$ for the opponent) we use the *stance loss* $\mathcal{L}_{\text{stance}}^i = \text{CE}(\Sigma^i(\mathbf{B}_t^i, (x, \bar{y}^i)), \bar{y}^i)$ to obtain the combined loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{dist}} + \left(\mathcal{L}_{\text{FAX}}^1 + \mathcal{L}_{\text{stance}}^1\right) + \left(\mathcal{L}_{\text{FAX}}^2 + \mathcal{L}_{\text{stance}}^2\right).$$

DEPLOYMENT. After training, we deploy the learnt agents for generating FAXs. To determine the number of timesteps in each FAX, we adopt the following strategy. We analyse cosine similarity between arguments to evaluate the information contributed in each timestep and we model the gain

3 Here, we treat the agent's initial stance as a (positive/negative) sign for the explanandum's current strength in the private BAF.

in information as the average dissimilarity of the contributed argument wrt all the previous arguments. We terminate the FAX if the mean dissimilarity is less then a small amount (which is a parameter) or when the FAX is resolved.

6.6 FAX: EVALUATION METRICS

In this section we lay out our approach to evaluating the realization of our FAXs for explaining image classifiers. We use evaluation metrics for assessing *faithfulness* and *argumentative quality* of our FAXs. The metrics are measured ona test set $\mathcal{T} \subseteq \mathcal{X} \times \mathcal{Y}$, providing ground-truths (correct classifications) for a number of inputs (we will use concrete instances of $\mathcal{T}$ in our experiments in Section 6.7).

We define the metrics using the same notation $\mathfrak{C}(z)$ as in Section 6.5 as well as notations $\mathfrak{C}(h)$ to represent the codebook corresponding to hidden state representation h (in some sequence model) and $q^i(h)$ to represent the values assigned by q^i to the private features/arguments corresponding to h (in ζ^i).

The faithfulness metrics are adapted from the literature, and (Kori et al., 2024a) in particular. The first metric measures *correctness* of q and of the codebooks by measuring accuracy wrt the ground-truth in $\mathcal{T}$:

$$| \{(x, y) \in \mathcal{T} \mid q(\mathfrak{C}(z)) = y\} | / | \mathcal{T} |$$

The second metric measures *completeness* of q on the BAFs resulting from FAXs, by measuring the accuracy of q on the codebooks corresponding to the hidden state representations of the arguments in these BAFs, wrt the classifier's predictions on $\mathcal{T}$:[4]

$$\left| \{(x, y) \in \mathcal{T} \mid q\left(\mathfrak{C}\left(\cup_{i=1}^{N} h_n^i\right)\right) = (g \circ f)(x)\} \right| / | \mathcal{T} |$$

This metric gives an indication of the faithfulness to the original classifier of q on the output FAXs (as the loss function used during training only strives towards faithfulness of q on the input to FAXs).

4 With an abuse of notation we use n to indicate the length of every FAX obtained from datapoints in $\mathcal{T}$, even though different FAXs will typically have different lengths.

Table 6.1: Accuracies of the trained classifiers

	Fair	Biased	Random
AFHQ-ResNet-18	0.95	0.39	0.31
AFHQ-DenseNet-121	0.96	0.47	0.32
FFHQ-ResNet-18	0.88	0.58	0.47
FFHQ-DenseNet-121	0.92	0.61	0.48

Table 6.2: Faithfulness properties of all considered methods on the DenseNet121 classifiers.

Methods → Dataset ↓	Correctness					Completeness				
	GradCAM	DeepLIFT	DeepSHAP	LIME	FAX	GradCAM	DeepLIFT	GradSHAP	LIME	FAX
FFHQ-Random	0.55	0.58	0.54	0.50	0.53	0.40	0.41	0.38	0.36	0.97
FFHQ-Biased	0.35	0.37	0.39	0.50	0.91	0.34	0.36	0.40	0.48	1.00
FFHQ-Fair	0.77	0.77	0.73	0.58	0.96	0.74	0.74	0.68	0.59	0.96
AFHQ-Random	0.30	0.32	0.28	0.25	0.55	0.35	0.36	0.33	0.31	0.72
AFHQ-Biased	0.27	0.32	0.37	0.48	0.71	0.35	0.38	0.40	0.47	0.91
AFHQ-Fair	0.75	0.71	0.71	0.52	0.78	0.70	0.66	0.67	0.54	0.99

The argumentative quality metrics are tailored to our (implementation of) FAXs. The third metric measures *consensus* amongst the (two) agents, in terms of the number of resolved FAXs:

$$\left|\{(x,y) \in \mathcal{T} \mid q^i(h_n^i) = q^j(h_n^j), j \neq i\}\right| \; / \; |\mathcal{T}|$$

The fourth (and final) metric (*pro persuasion rate*) measures consensus again, but towards the proponent agent:

$$\left|\{(x,y) \in \mathcal{T} \mid q^1(h_0^1) = q^j(h_n^j), j \neq 1\}\right| \; / \; |\mathcal{T}|$$

Given that agents are trained to disagree (see Definition 6.9) both argumentative metrics can be seen as estimates of the goodness of the learnt class-specific features: high values indicate that information is leaked across different features (arguments).

6.7 FAX: results

We analyse FAXs on the high resolution animal and human faces (AFHQ (Choi et al., 2020), FFHQ (Karras, 2019)) datasets, with two well known

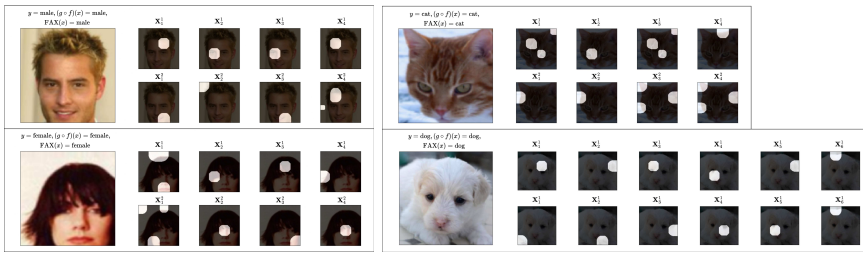


Figure 6.3: Arguments in FAXs (the proponent starts with the top left argument, the opponent follows with the argument below it, etc.), for classifiers trained on FFHQ (left column) and AFHQ (right column), on fair (top row) and biased (bottom row) settings.

architectures ResNet-18 (He et al., 2016) and DenseNet121 (Huang et al., 2017) as image classifiers $g \circ f$. We consider three settings: (i) *fair*, where the classifier is trained with correct labels; (ii) *biased*, where the classifier is trained with biased labels, obtained by randomly switching the labels for 10% of the datasets, and (iii) *random*: where we use randomly initialised weights for the classifiers rather than training them.

QUALITATIVE RESULTS Figure 6.3 shows some FAXs visualised as in (Kori et al., 2024a), using the approach in (Bau et al., 2017). In all these figures, the first image is the input, while the first and second rows correspond respectively to $\mathcal{A}^1$'s and $\mathcal{A}^2$'s arguments. We can see that FAXs focus on different but semantically meaningful regions on the input images, for both agents. In the biased setting, as previously described, we expect some leakage of information across different class-specific features/arguments, as observed in the figure.

QUANTITATIVE RESULTS Table 6.1 gives the classifiers' accuracy on test sets. For the DenseNet121 classifiers, Table 6.2 gives the faithfulness results in comparison with standard baselines (i.e. GradCAM (Selvaraju et al., 2017), DeepLIFT (Shrikumar et al., 2017), DeepSHAP (Lundberg and Lee, 2017) and LIME (Ribeiro et al., 2016)) and Table 6.3 measures the argumentative metrics in the three settings (for FAXs only, as these metrics are not applicable to baseline methods). Further, Table 6.4 measure all metrics for the ResNet-18 classifiers. We observe that *completeness* is high in all settings, while *correctness* is high for the fair and biased settings only; this

Table 6.3: Argumentative metrics for DenseNet-121 classifiers.

	Fair	Biased	Random
AFHQ-Consensus	0.24	0.44	0.26
AFHQ-Pro persuasion rate	0.27	0.41	0.77
FFHQ-Consensus	0.42	0.54	0.09
FFHQ-Pro persuasion rate	0.33	0.50	0.38

Table 6.4: All metrics for ResNet-18 classifiers.

	Fair	Biased	Random
AFHQ-Correctness	0.77	0.73	0.29
AFHQ-Completeness	0.99	0.93	0.68
AFHQ-Consensus	0.57	0.89	0.13
AFHQ-Pro persuasion rate	0.48	0.56	0.16
FFHQ-Correctness	0.65	0.45	0.51
FFHQ-Completeness	0.99	0.99	0.91
FFHQ-Consensus	0.31	0.90	0.45
FFHQ-Pro persuasion rate	0.24	0.53	0.71

is due to completeness reflecting FAXs behaviour wrt continuous classifier, while correctness is measured wrt ground truth. As for the last two metrics, the experiments show higher values in the biased than fair settings, as expected, given that we expect the leak of features across codebooks due to incorrect label assignment in the former, which results in higher consensus and easier persuasion. We can observe mixed behaviours with random classifiers, which results in high value in the case of overlapping features during initialisation and low in the other case. Overall, the experiments show that for more "uncertain" models (random), the agents cannot reach as much consensus as for "certain" models (fair and biased) and the persuasion rate is lower for "uncertain" models. This analysis is empowered by the argumentative nature of our explanatory framework.

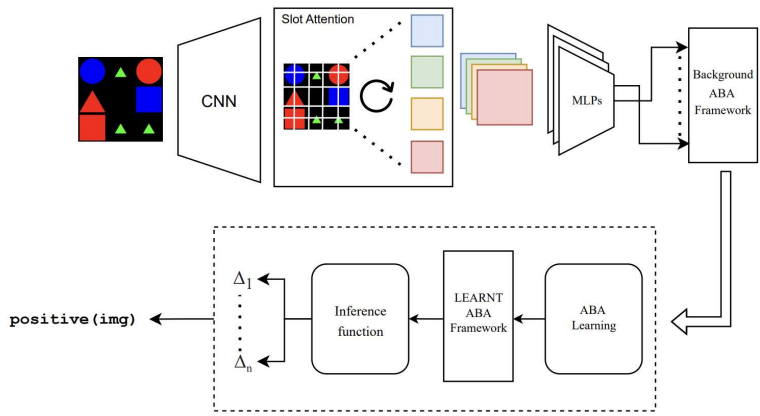


Figure 6.4: Our NAL architecture: the input image (top left) is processed by slot attention to obtain objects (coloured squares) mapped by multi-layer perceptrons (MLPs) into facts fed into a Background ABA framework (which may also contain other knowledge); then ABA learning generates a Learnt ABA framework, which may admit several 'extensions'; inference therewith gives a classification.

We have defined explanations for image classification as (free) argumentative exchanges between two agents, aiming to demystify trained image classifiers based on the argument contribution strategies by the agents. Differently from standard feature attribution methods generating heatmaps over responsible regions in images, our method generates more fine-grained composition of sub-regions, incrementally. Our work opens many opportunities for future work. We plan to investigate whether FAXs can uncover shortcuts in classifiers. Further, it would be valuable to collaborate with domain experts to attribute semantic meaning to arguments, potentially aiding alignment between human understanding and the latent knowledge of models.

6.8 OBJECT-CENTRIC ARGUMENTATIVE LEARNING

Here, we introduce a novel Neural Argumentative Learning (NAL) architecture that integrates Assumption-Based Argumentation (ABA) with deep learning for image analysis. Our architecture consists of neural and

symbolic components. The former segments and encodes images into facts using object-centric learning, while the latter applies ABA learning to develop ABA frameworks enabling predictions with images. Experiments on synthetic data show that the NAL architecture can be competitive with a state-of-the-art alternative.

To generate ABA frameworks, our NAL architecture uses ABA Learning (De Angelis et al., 2023), a method that uses argumentation in a logic-based learning fashion to generate ABA frameworks which, with their accepted arguments, cover given positive examples and do not cover given negative examples. The architecture also uses slot attention (Locatello et al., 2020b) to support a granular understanding of input images in terms of the objects they contain. Overall, our NAL architecture enables the extraction of meaningful properties and relationships between objects within the images, facilitating accurate classification with interpretable argumentation frameworks. Overall, we make the following contributions: 1) we tailor slot-attention to generate factual background knowledge suitable to be injected in ABA Learning; 2) we combine slot-attention and ABA Learning into a pipeline architecture for neuro-argumentative learning; 3) we assess the performances of our method experimentally on (synthetic) image datasets, showing that it can be competitive against a baseline (Neuro-Symbolic Concept Learner (NS-CL) (Mao et al., 2019b)).

6.8.1 *NAL: Architecture*

We will now explain the NAL architecture shown in Figure 6.4, by detailing the neural and symbolic components and their training, as well as inference post-learning. The NAL architecture accepts a dataset $D \subseteq X \times Y \times L$ of N labelled images, where $L = \{c_1, c_2\}$ and $Y = \{0,1\}^{K \times (P+1)}$ for K the total number of objects/slots that may occur in images and P the total number of *properties* that each of these objects may have (we consider an extra property for characterising the absence of objects). As a simple example, for images such as the one in Figure 6.4, $K = 10$ (as there are a maximum of 9 objects in each such image plus the background) and $P = 8$ (3 for the shapes, 3 for the colours, 2 for sizes). We assume that D is not noisy. L is a set of two alternative classes. Y is a metadata consisting of one-hot encodings, each

representing all objects and their corresponding properties in an image. Note that the neural component of the architecture disregards the label in L, while the symbolic component disregards the image itself, using instead its abstraction drawn from the neural component.

For this the input is first converted into slots using SA (Locatello et al., 2020b), then, each slot $\hat{z}$ is passed through the MLPs to extract the properties of each object. To identify both continuous and categorical properties, we use MLPs of two types: classification MLPs, which use a softmax activation in the final layer to predict the most likely attribute for a given slot and regression MLPs, to predict the location of objects and determine whether each given slot has attended to a real object in the image. The results of these predictions are then concatenated to form the final prediction $\hat{y}$ for the input image, with the corresponding ground truth $y \in Y$. This component is trained with weak-supervision and is optimised by minimising the loss function:

$$\text{MSE}(x, \hat{x}) + \alpha \min_{\tau \in S_K} \sum_{j=1}^{P} \text{BCE}(y_j, \tau(\hat{y})_j). \tag{6.7}$$

The first is the mean square error (MSE) between the input image x and the reconstructed image $\hat{x}$, ensuring the reconstruction quality of the model. The second is the binary cross entropy (BCE) between the ground truth label y and the predicted label $\hat{y}$, where y_j corresponds to a particular property in vector y and $\tau(\hat{y})$ is the prediction for a permutation of the K objects, drawn from S_K, which denotes the set of all such permutations of K objects. Intuitively, we compareeach of the permutations with the ground-truth representation (with a specific order). Given the equivariance property of slot attention, we need to first align the slots before applying the BCE loss. To circumvent this, we used the Hungarian matching algorithm. Finally we balance both loss terms with the hyperparameter α.

6.8.2 *NAL: Symbolic Component*

This receives the output predictions from the neural component and transforms them into facts for the Background ABA Framework taken in input by the ASP-ABALearn algorithm. It also uses the input labels in L to obtain appropriate positive and negative examples in $(\mathcal{E}^+, \mathcal{E}^-)$. This is accom-

plished by aggregating the slots and performing K-means clustering. The number of clusters corresponds to the desired number of examples, and an image is chosen from each cluster as a representative positive or negative example. We also check the confidence of each prediction and prune off any image below a certain threshold.

The slot predictions are then passed to concept embedding functions which use a dictionary (for the K objects and the P properties) to convert the raw predictions to ABA facts. For each image and object, an identifier is given in the form of an atom `image(img_i)` and a constant `object_i` respectively. Then, for each slot prediction, we take the argmax to identify the properties in the dictionary which are attributed to the object. We encode this as a fact, e.g. `blue(object_i)`. Once all images and objects are encoded into the Background ABA Framework, we generate the ASP-ABALearn command `aba_asp('filename.aba', e_pos,e_neg)`. This specifies which images are positive/negative, using `e_pos`, `e_neg` as the encodings of $\mathcal{E}^+, \mathcal{E}^-$, as discussed earlier. The ABA-ASPLearn algorithm is then run to produce a Learnt ABA Framework.

6.8.3 *NAL: Inference*

At inference time, we run a slightly different pipeline to obtain a final classification for each unseen input image. Specifically:

1. We pass the image through the neural component to obtain predictions of the objects and their properties therein. These are subsequently converted into facts as during training.

2. We then create an ABA Framework which contains these facts, the rules learnt via ASP-ABALearn and any extra background knowledge.

3. The stable extensions of this ABA framework are then computed (using a straightforward mapping into ASP, and then using an ASP solver. specifically we use Clingo (Gebser et al., 2019)).

4. Depending on the ABA framework, we may obtain more than one stable extension. The prediction boils down to checking whether the atom sanctioning that the input image belongs to concept c_1 is a

SHAPES Dataset ASP Rules
`s1(A) :- image(A), in(A,B), square(B), blue(B).`
`s2(A) :- image(A), in(A,B), triangle(B), small(B), green(B).`
`s3(A) :- image(A), in(A,B), in(A,C), triangle(B), blue(B), circle(C), red(C), large(C).`
`s4(A) :- image(A), in(A,B), in(A,C), circle(B), red(B), square(C), blue(C), above(B,C).`
`s5(A) :- image(A), in(A,B), in(A,C), triangle(B), red(B), circle(C), green(C), left(B,C).`
`s6(A) :- not exception(A), image(A).`
`exception(A) :- image(A), in(A,B), circle(B), blue(B).`

Figure 6.5: ASP rules used to generate the 6 classes for the SHAPES dataset. We generated 3K images for each rule half of which was the negative instance of the rules. We then took 500 positive and negative splits for each rule as testing data. In total, we had 12K images for training and 6K for testing.

member of all the stable extensions (i.e., it is a *cautious* consequence of the Learnt ABA framework).

6.8.4 *NAL: Experimental evaluation*

We conducted experiments on the NAL architecture to answer the following questions: **Q1**: How well can the Neural Component identify/predict object properties? **Q2**: How well can the NAL architecture learn ABA frameworks which describe the latent rules in images so that they be used for classification? and **Q3**: How well does our NAL architecture scale w.r.t. the number of examples used in the Symbolic Component and the complexity of the latent rules? To address these questions, we defined various binary classification tasks using our SHAPES-inspired dataset. This dataset was generated by a tool that processed ASP rules to create images conforming to them. We defined 6 rules, as detailed in Table 6.5, to form the dataset. Each binary classification task aimed to distinguish between positive and negative instances for each class. This dataset served as a baseline to evaluate the viability of using argumentation to reason in an object-centric way. We also defined a multi-class classification task on the CLEVR dataset using CLEVR-Hans3 (Stammer et al., 2021) which splits CLEVR images into 3 classes based on the following concepts c1: `Large (Gray) Cube and Large Cylinder`, c2: `Small metal Cube and Small (metal) Sphere`, c3: `Large blue Sphere and Small yellow Sphere`. The goal of this task was for NAL to generate Learnt ABA frameworks that could differentiate these classes.

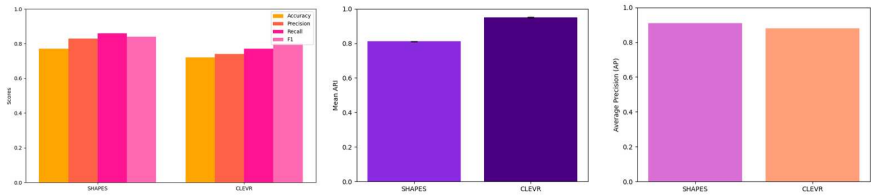


Figure 6.6: (Left) Standard machine learning evaluation metrics of attribute classification on datasets SHAPES and CLEVR **(Center)** Mean Adjusted Rand Index on datasets **(Right)** Average Precision on datasets.

SETUP We trained the NAL architecture in two stages as described in above section. The Neural Component was trained using the full datasets for 1000 epochs with hyperparameter $\alpha = 0.35$. The Symbolic Component used 10 positive and 10 negative examples from the datasets to obtain the Learnt ABA framework. Regarding the CLEVR-Hans3 task (which was multi-classed), we executed the Symbolic Component twice, the first run to distinguish c3 images from c1 and c2 and the second run to distinguish between c1 and c2, thus producing two frameworks. We compare the results with a ResNet (He et al., 2016) and the NS-CL.

[Q1] OBJECT PREDICTION Figure 6.6 shows that the Neural Component effectively segmented images into their constituent objects and accurately predicted each object's property. We evaluated its localisation and segmentation capabilities using the Adjusted Rand Index, which measures similarity between clusters (with clusters representing objects and data points representing pixels), and the Average Precision metric. The scores ranged from 0.80 to 0.95, indicating consistent performance regardless of the number of objects present in each dataset. We also observed high scores in standard machine learning metrics, indicating that the MLPs accurately predicted each object's properties. The F1 score exceeded 70%, suggesting that the model effectively minimises false positives and negatives, ensuring the symbolic component contains facts that accurately represent the image. Results on the CLEVR dataset were slightly worse than those on the SHAPES dataset, likely due to CLEVR objects having a larger number of properties. This suggests that the number of properties can affect prediction performance.

[Q2] CLASSIFICATION From Table 6.5 we observe that the Learnt ABA frameworks performed well on the binary classification tasks for SHAPES, achieving near-perfect scores on most tasks. Among the metrics, we saw that recall was consistently the lowest. This is likely due to errors propagated from the Neural Component, which resulted in facts that did not accurately represent the images, causing some instances to be misclassified.
However, we saw the opposite when predicting classes s4 and s5, with significantly lower precision. We believe that this is due to the fact that in those tasks ASP-ABALearn learnt rules capturing (some of) the exceptions rather than the full concepts. For example, s1 images were defined in the dataset (see Table 6.5) as those with blue squares. However, the learnt framework defined `s_1` as images with squares that are not red and not green (see `c_alpha2` in Figure 6.7). Despite being semantically equivalent, this reasoning made interpretation difficult and impacted the generation of more complex rules. Consequently, the F1 scores for these tasks dropped significantly due to the inability to capture all rule exceptions, leading to many false positives and thus lower precision. We encountered a similar outcome with the learnt ABA frameworks generated for the CLEVR-Han3 tasks. These only partially captured the rules defining each class. For instance, the framework for c1 stated that these images contain cubes that are not small. However, this rule failed to distinguish some c2 images, which could also contain cubes that are not small. As a result, many c2 images were incorrectly classified as c1, as shown in the confusion matrix (see Figure 6.11). Despite this issue, the learnt ABA frameworks were able to capture most of the facts for classifying c3 images, leading to an F1 score of 0.68. This outperformed the ResNet baseline by 10%, though it was slightly worse than NS-CL, which scored in the 80% range. The superior performance of NS-CL could be attributed to its use of a set transformer for classification, rather than relying solely on symbolic reasoning.

[Q3] SCALABILITY During our experiment, we found that the symbolic component, specifically ASP-ABALearn, faced some scalability issues as the execution time grew significantly as we increased the number of examples. This could be due to a larger search space generated when looking for ABA frameworks whose extensions cover all positive examples and none of

```
% Learnt Rules
s_1(A) :- in(A,B), square(B), alpha_2(B,A).
c_alpha_2(A,B) :- image(B), red(A).
c_alpha_2(A,B) :- image(B), green(A).
```

Figure 6.7: Rules in the ABA framework generated by our NAL architecture for SHAPES (class s1). Here, alpha_2 is an assumption, with $\overline{alpha\_2(A,B)}$=c_alpha_2(A,B).

```
% Learnt Rules
c_1(A) :- in(A,B), cube(B), alpha_2(B,A).
c_alpha_2(A,B) :- small(A), image(B).
```

Figure 6.8: Rules in the ABA framework generated by our NAL architecture for SHAPES (class s5). Here, alpha_2 is an assumption, with $\overline{alpha\_2(A,B)}$=c_alpha_2(A,B).

```
% Learnt Rules
c_3(A) :- in(A,B), sphere(B), alpha_2(B,A).
c_alpha_2(A,B) :- brown(A), image(B).
c_alpha_2(A,B) :- green(A), image(B).
c_alpha_2(A,B) :- cyan(A), image(B).
c_alpha_2(A,B) :- red(A), image(B).
c_alpha_2(A,B) :- large(A), image(B).
c_alpha_2(A,B) :- blue(A), image(B).
c_alpha_2(A,B) :- gray(A), image(B).
```

Figure 6.9: Rules in the ABA framework generated by our NAL architecture for CLEVR, differentiating class c3 from c1 and c2. Here, alpha_2 is an assumption, with $\overline{alpha\_2(A,B)}$=c_alpha_2(A,B).

Predicted class	Accuracy	Precision	Recall	F1-Score
s1	99.0 ± 0.0	100.0 ± 0.0	97.5 ± 0.0	99.0 ± 0.2
s2	96.0 ± 0.0	99.0 ± 0.0	93.5 ± 0.0	96.0 ± 0.2
s3	98.0 ± 0.0	100.0 ± 0.0	97.5 ± 0.2	98.0 ± 0.0
s4	75.0 ± 3.0	61.0 ± 5.2	98.5 ± 0.4	75.0 ± 0.2
s5	86.0 ± 4.1	77.0 ± 3.8	96.5 ± 0.2	84.0 ± 0.2
s6	99.0 ± 0.0	98.0 ± 0.0	100.0 ± 0.0	99.0 ± 0.0

Table 6.5: Standard evaluation metrics denoting how well NAL can distinguish between positive and negative instances of rules present in SHAPES images

	Accuracy	Precision	Recall	F1-Score
NAL	69.1 ± 0.3	70.0 ± 0.1	69.1 ± 0.2	68.0 ± 0.2
ResNet	65.2 ± 0.3	66.2 ± 0.4	65.2 ± 0.2	61.0 ± 0.2
NS-CL	84.7 ± 0.1	86.1 ± 0.2	84.7 ± 0.2	84.0 ± 0.2

Figure 6.10: Standard evaluation metrics for the performance of NAL, ResNet, and NS-CL on the CLEVR-Hans3 task.

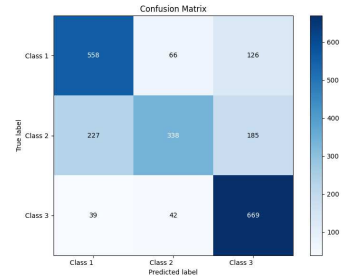


Figure 6.11: Confusion Matrix of NAL on the CLEVR-Hans3 task.

the negative examples. It may happen that this search is unsuccessful and ASP-ABALearn halts with failure.

Our experiments also showed that the non-determinism present when generalising rules led to variability in both the quality of the results and the system's execution time. This variability was amplified by the size of the Background ABA framework. Additionally, we observed that longer execution task times negatively impacted the quality of the results, as heavily nested rules caused the frameworks to overfit to the example set.

6.9 COSA: END-TO-END VISUAL REASONING

Here, we provide an application of CoSA algorithm as detailed in Chapter 3 from a reasoning perspective, which is alternative to NAL more from an end-to-end training objective. The reasoning tasks we consider consist of classifying images into rule-based classes $\mathbf{y} \in \mathcal{Y}_1$, while providing a *rationale* $\mathbf{p}$ behind the predicted class. For example, class A might be something like "large cube and large cylinder", and a rationale for a prediction emerges directly from the slots binding to an input cube and cylinder. Refer to Appendix A.3 for more details.

To achieve this, we define a set of learned functions $\mathfrak{S}^3 := \{\mathfrak{S}^3_1, \ldots, \mathfrak{S}^3_M\}$, which map refined slot representations $\mathbf{s}^{t=T}$ to M *rationales* for each object *type* in the grounded slot dictionary $\mathfrak{S}$. This expands the dictionary to three elements: $\mathfrak{S} := \{(\mathfrak{S}^1, \mathfrak{S}^2, \mathfrak{S}^3)\}$. The reasoning task predictor Φ_r combines K rationales extracted from K slot representations of an input, which we denote by $\mathbf{p}_i = \mathfrak{S}^3_i(\mathbf{s}^T_i) \in \mathbb{R}^{|\mathcal{Y}_2|}$, and maps them to the rule-based class

labels. The optimization objective for this reasoning task is a variational lower bound on the conditional log-likelihood, as in Proposition 6.2.

Proposition 6.2 (ELBO for Reasoning Tasks)**.** *Under a categorical distribution over our discrete latent variables* $\tilde{\mathbf{z}}$*, and the object-level prior distributions* $p(\mathbf{s}_i^0) = \mathcal{N}\left(\mathbf{s}_i^0; \boldsymbol{\mu}_i, \sigma_i^2\right)$ *contained in* $\mathfrak{S}^2$*, the variational lower bound on the conditional log-likelihood of* $\mathbf{y}$ *given* $\mathbf{x}$ *is given by:*

$$\log p(\mathbf{y} \mid \mathbf{x}) \geq \mathbb{E}_{\tilde{\mathbf{z}} \sim q(\tilde{\mathbf{z}}|\mathbf{x}), \mathbf{s}^0 \sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \left[\log p(\mathbf{y} \mid \mathbf{s}, \mathbf{x})\right] - D_{\mathrm{KL}}\left(q(\tilde{\mathbf{z}} \mid \mathbf{x}) \parallel p(\tilde{\mathbf{z}})\right). \tag{6.8}$$

noting that conditioning $\mathbf{y}$ *on* $\{\mathbf{s}, \mathbf{x}\}$ *is equivalent to conditioning on the predicted rationales* $\mathbf{p}$*, since the latter are deterministic functions of the former.*

The proof is similar to proposition 3.1 and is given in Appendix A.2 for completeness.

As briefly outlined in the reasoning module description, the visual reasoning task involves classifying images into rule-based classes, while providing a rationale behind the predicted class. To evaluate and compare our models, we use the F1-score, accuracy, and the Hungarian Matching Coefficient (HMC), applied to benchmark datasets: CLEVR-Hans3, CLEVR-Hans7 (Stammer et al., 2021), FloatingMNIST-2, and FloatingMNIST-3. CLEVR-Hans includes both a confounded validation set and a non-confounded test set, enabling us to verify the generalizability of our model. FloatingMNIST (FMNIST) is a variant of the MNIST (Deng, 2012) dataset with three distinct reasoning tasks, showcasing the model's adaptability across domains. Further details on the datasets are given in Appendix A.3.

To facilitate comparisons, we train and evaluate several classifiers, including: (i) the baseline classifier; (ii) a classifier with a vanilla SA bottleneck; (iii) Block-Slot; and (iv) our CoSA classifiers. The main comparative results are presented in Table 6.6, and additional results for FloatingMNIST3 and CLEVR-Hans datasets are available in the Appendix (Tables A.13-A.16). We conduct task adaptability experiments to assess the reusability of grounded representations in our learned GSD. To that end, we create multiple variants of the FloatingMNIST dataset, introducing addition, subtraction, and mixed tasks. Initially, we train the model with one specific objective, which we consider as the source dataset and assess its capacity to adapt to other

Table 6.6: Accuracy and Hungarian matching coefficient (HMC) for reasoning tasks on addition and subtraction variant of FMNIST2 dataset. Here, the first and third pair of columns correspond to the models trained and tested on the FMNIST2-Add and FMNIST2-Sub datasets, respectively, while the second and fourth pair correspond to few-shot (k=100) adaptability results across datasets.

Method	FMNIST2-Add$_{source}$		FMNIST2-Sub$_{target}$		FMNIST2-Sub$_{source}$		FMNIST2-Add$_{target}$	
	Acc ↑	HMC ↓	Acc ↑	F1 ↑	Acc ↑	HMC ↓	Acc ↑	F1 ↑
CNN	97.62	-	10.35	10.05	98.16	-	12.35	9.50
SA	97.33	0.14	11.06	09.40	97.41	0.13	08.28	7.83
Block-Slot	98.11	0.12	09.71	09.10	97.42	0.14	09.61	8.36
CoSA	**98.12**	**0.10**	**60.24**	**50.16**	**98.64**	**0.12**	**63.29**	**58.29**

target datasets by fine-tuning the task predictor layer through k-shot training. The adaptability results for FloatingMNIST-2 are provided in Table 6.6, and results for FloatingMNIST-3 are detailed in the Appendix in Table A.16. Results for mixed objectives are also discussed in Appendix A.11. Fig. A.22-A.24 depict the few-shot adaptation capabilities over multiple k iterations on the FloatingMNIST2, FloatingMNIST3, and CLEVR-Hans datasets. Overall we observe that GSD helps in (i) better capturing the rational for the predicted class, as illustrated by HMC measure; and (ii) learning reusable properties leading to better generalisability as demonstrated by accuracy and F1 measure in k-shot adaptability tasks; detailed in Table 6.6, A.13- A.15.

6.10 Conclusions & Future Work

We have proposed a bunch of explainability methods incorporating argumentative reasoning, firstly we proposed novel agent interaction based explainability method and a neuro-argumentative learning method for image classification. To address the issue of human understandability of concepts, we extended it to satisfy object representations with a multi-stage approach, and finally extended it to end-to-end training approach. The proposed framework follows a faithful and human-understandable reasoning process. We empirically demonstrate that our approach can be effective via experiments in all three different models considered. with the resulting architecture on datasets of synthetic images.

This page is intentionally left blank.

7

CONCLUSION

"The future depends on what you do today."

– Mahatma Gandhi

This chapter concludes the thesis by summarizing key contributions, highlighting the implications of the proposed methods, and outlining potential directions for future work within the evolving landscape of artificial intelligence. At its core, this thesis has explored the foundations and frontiers of object-centric learning as a promising pathway toward human-like visual reasoning. Motivated by the limitations of conventional deep learning—particularly its lack of interpretability, poor generalization, and inability to model causal structure—we introduced a series of models and theoretical insights that reconceptualize perception as the discovery and organization of discrete, meaningful entities: objects.

Through the development of novel models such as CoSA (Chapter 3), PSA (Chapter 4), and VISA (Chapter 4), this work demonstrated that object-centric representations can be learned in a grounded, identifiable, and viewpoint-invariant fashion, even under partial observability. These models depart from conventional black-box feature extractors by offering structured, modular, and interpretable representations that align more closely with how humans perceive and reason about the world. From a practical standpoint, these representations offer tangible advantages—reusability across tasks and transparent reasoning capabilities. These properties are especially crucial in high-stakes domains such as healthcare, finance, and public policy, where decision-making must be both reliable and understandable. Another main insight of this work lies in the unification of object-centric learning for causal reasoning and argumentative reasoning. This fusion enables AI systems to move beyond surface-level pattern recognition and into the realm of reasoning—supporting interventions, counterfactual thinking, and the discovery of abstract causal dependencies. This synthesis moves

us closer to AI systems that understand not just what is seen, but also why things occur and how they might change under hypothetical scenarios. Alternative to this line of reasoning, we introduced FAX—a novel framework for argumentative and agent-to-agent communicative reasoning. Rather than relying on static saliency maps or opaque confidence scores, FAX treats explanation as an interactive, dialectical process. This opens up new possibilities for AI systems that can justify their decisions, engage with conflicting perspectives, and offer reasoned, structured dialogues with humans and other agents.

To summarise from the perspective of the objectives outlined in Chapter 1, Chapter 3 presents a novel algorithm based on the newly introduced concept of the GSD. This addresses **Objective 1**—the notions of binding and grounding—by mapping slots to elements within the grounded dictionary. This concept is further developed in Chapter 6 to support a reasoning-focused objectives. In Chapter 4, we address **Objectives 2 & 3**—namely, the identifiability of object representations and identifiability under spatial ambiguities. This is achieved by adapting SA to incorporate object-level priors, resulting in the PSA and VISA algorithms. These methods leverage structure in the latent space to enable identifiability without relying on additional auxiliary data. Finally, Chapters 5 and 6 apply the developed theoretical framework to reasoning tasks, addressing **Objectives 4 & 5**. In particular, Chapter 5 introduces the notion of causal compositions, which leads to the concept of compositional graphs and latent compositional models—paralleling causal graphs and latent causal models, enabling compositional reasoning. In chapter 6 we introduce FAX and NAL frameworks, that relies on OCL and computational argumentation for generating human understandable reasoning process.

Each of the proposed method in this work, including CoSA, PSA, VISA, FAX, and NAL, was designed not in isolation, but as part of a broader framework that ties together perception, causality, explainability, and probabilistic scene modelling. To contextualize the broader relevance of these ideas, it is essential to reflect on their implications. What kind of systems become possible when we can learn disentangled, grounded, and identifiable representations without supervision? How does causal reasoning evolve when the latent variables correspond to physical objects with meaningful

interactions? What does it mean for AI to explain itself—not through post-hoc rationalizations, but through argumentative interactions akin to human discourse? These questions lie at the heart of trustworthy, intelligent, and socially-aligned AI. Here, we outline the most significant implications of this thesis, each addressing a key dimension of this evolving paradigm:

1. Toward Structured, Object-Centric Representations: Here, we advocate for object-centric representations over flat, monolithic embeddings, this work suggests that structured perception—where individual entities are disentangled and encoded as separate "slots"—provides a natural basis for reasoning, abstraction, and generalization. Much like human cognition, which segments scenes into manipulable and conceptual units (objects, agents, tools), OCL enables us to computationally achieve the same for objects. As a future work, it would be interesting to explore the application of these representations from an embodied AI viewpoint, encouraging re-use of concepts across environments.

2. Identifiability as the Key to Generalization and Interpretability: One of the key contributions in this thesis is the introduction and enforcement of identifiability in object-centric learning. Identifiability ensures that the latent variables representing objects remain consistent and semantically grounded across different contexts, transformations, or inputs. By promoting identifiable slots, models can anchor abstract concepts to persistent entities, making them both more generalizable and more interpretable. This property is critical for debugging AI systems, transferring knowledge across domains, and aligning internal representations with symbolic or human-understandable categories. In-depth analysis of this property across multiple downstream tasks requires a study on its own, which we leave to explore in the future.

3. Multi-View Scenarios to handle Partial Visibility: In settings where an agent observes a dynamic scene from multiple partial views (e.g., a robot navigating a room or a drone mapping terrain), consistent object representation is essential for effective action and planning. The proposed VISA framework shows that this invariance can be learned

without supervision, leveraging multi-view correspondences and spatial reasoning. This contribution significantly expands the applicability of object-centric learning to embodied AI, enabling agents that can understand and act in 3D environments with persistence and memory across views. Even though we make some specific assumptions including *weak injectivity* of the mixing functions and viewpoint sufficiency, we do provide evidence for why these assumptions are reasonable. Addressing this constraint in future work could strengthen identifiability guarantees. Additionally, the impact of partial visibility results on real-world applications requires further investigation.

4. Unifying Causal and Object-Centric Representation Learning: While OCL enables structural abstraction, CRL brings explanatory depth —highlighting the dependencies and dynamics that underlie observed data. By unifying these two paradigms, this work allows AI systems to not only recognize and track objects, but to reason about how those objects influence one another and how their attributes evolve under intervention. This opens the door to future work on agents that can perform counterfactual reasoning, plan actions based on cause-effect understanding, and make robust decisions under uncertainty. As future work, it would be worthwhile to explore the use of the proposed theories to connect current AI systems with broader theories of cognition, where causal inference is considered foundational to intelligence.

5. FAX -Toward Transparent and Argumentative AI: The proposed FAX framework takes a step toward explainability modelled as agent-to-agent communication. Instead of static visualizations or feature attributions, FAX treats explanation as a process—one where models articulate their internal reasoning, justify decisions, and challenge conflicting viewpoints. Inspired by human argumentative dialogue, this approach emphasizes interaction, contrast, and justification, mimicking how people reason together. This way of viewing explanations as dialogue may be useful in multi-modality scenario to help in understanding the reasoning process of current large scale VLMs: we leave this topic to future work.

6. Ethical and Societal Impact through Interpretable and Grounded AI: The interpretability, identifiability, and structured reasoning promoted in this thesis have far-reaching consequences for the ethics and governance of AI systems. As AI is increasingly deployed in high-stakes domains—such as healthcare, finance, law, and autonomous vehicles—the need for verifiable and transparent decision-making becomes urgent. Models trained under proposed frameworks can produce outputs that are more auditable, less prone to hidden bias, and easier to explain to non-experts. Moreover, by grounding representations in physically meaningful entities, this work paves the way to future work towards integrating symbolic logic, rule-based constraints, and human values directly into AI systems, with the potential of contributing to safe and human-aligned AI.

Together, these contributions form a key step toward a new generation of artificial systems—ones that perceive, think, and explain more like humans. While significant challenges remain, particularly in scaling these ideas to real-world complexity and integrating them with embodied agents, the research presented here lays a solid theoretical and empirical foundation for progress in that direction.

REFERENCES

Ahuja, K., Wang, Y., Mahajan, D., and Bengio, Y. (2022). "Interventional Causal Representation Learning". In: *arXiv preprint arXiv:2209.11924* (cited on pp. 42, 43).

Amgoud, L. and Ben-Naim, J. (2018). "Weighted Bipolar Argumentation Graphs: Axioms and Semantics". In: *IJCAI*, pp. 5194–5198 (cited on p. 140).

Atkinson, K., Baroni, P., Giacomin, M., Hunter, A., Prakken, H., Reed, C., Simari, G., Thimm, M., and Villata, S. (2017). "Towards artificial argumentation". In: *AI magazine* 38.3, pp. 25–36 (cited on pp. 34, 49).

Authors, G. (2024). *Genesis: A Universal and Generative Physics Engine for Robotics and Beyond* (cited on p. 125).

Bardes, A., Ponce, J., and LeCun, Y. (2021). "Vicreg: Variance-invariance-covariance regularization for self-supervised learning". In: *arXiv preprint arXiv:2105.04906* (cited on p. 200).

Baroni, P., Rago, A., and Toni, F. (2018). "How Many Properties Do We Need for Gradual Argumentation?" In: *AAAI*, pp. 1736–1743 (cited on p. 132).

Baroni, P., Rago, A., and Toni, F. (2019). "From fine-grained properties to broad principles for gradual argumentation: A principled spectrum". In: *Int. J. Approx. Reason.* 105, pp. 252–286 (cited on pp. 50, 140).

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (cited on pp. 31, 71).

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. (2017). "Network Dissection: Quantifying Interpretability of Deep Visual Representations". In: *CVPR*, pp. 3319–3327 (cited on p. 146).

Behrens, T. E., Muller, T. H., Whittington, J. C., Mark, S., Baram, A. B., Stachenfeld, K. L., and Kurth-Nelson, Z. (2018). "What is a cognitive map? Organizing knowledge for flexible behavior". In: *Neuron* 100.2, pp. 490–509 (cited on pp. 30, 71).

Bengio, Y., Courville, A., and Vincent, P. (2013a). "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828 (cited on pp. 30, 31, 57, 71, 124).

Bengio, Y., Yao, L., Alain, G., and Vincent, P. (2013b). "Generalized denoising auto-encoders as generative models". In: *Advances in neural information processing systems* 26 (cited on p. 105).

Besnard, P., Garcia, A., Hunter, A., Modgil, S., Prakken, H., Simari, G., and Toni, F. (2014). "Introduction to structured argumentation". In: *Argument & Computation* 5.1, pp. 1–4 (cited on p. 49).

Biederman, I. (1987). "Recognition-by-components: A theory of human image understanding". Vol. 94. 2, pp. 115–147 (cited on pp. 32, 39).

Bishop, C. M. (2006). "Pattern Recognition and Machine Learning". Springer: New York (cited on p. 77).

Brady, J., Zimmermann, R. S., Sharma, Y., Schölkopf, B., Kügelgen, J. von, and Brendel, W. (2023). "Provably Learning Object-Centric Representations". In: *arXiv preprint arXiv:2305.14229* (cited on pp. 48, 49, 70–72, 85, 86, 99, 107, 225, 233, 236).

Brehmer, J., De Haan, P., Lippe, P., and Cohen, T. S. (2022). "Weakly supervised causal representation learning". In: *Advances in Neural Information Processing Systems* 35, pp. 38319–38331 (cited on pp. 42, 43, 112).

Brouwer, A. E. and Haemers, W. H. (2011). "Spectra of graphs". Springer Science & Business Media (cited on p. 119).

Buchholz, S., Besserve, M., and Schölkopf, B. (2022). "Function Classes for Identifiable Nonlinear Independent Component Analysis". In: *Advances in Neural Information Processing Systems* (cited on p. 42).

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. (2019). "Monet: Unsupervised scene decomposition and representation". In: *arXiv preprint arXiv:1901.11390* (cited on pp. 45, 70, 86, 107, 121, 189).

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660 (cited on pp. 210, 269).

Castro, D. C., Walker, I., and Glocker, B. (2020). "Causality matters in medical imaging". In: *Nature Communications* 11.1, p. 3673 (cited on p. 124).

Cawsey, A. (1991). "Generating Interactive Explanations". In: *AAAI*, pp. 86–91 (cited on p. 129).

Cayrol, C. and Lagasquie-Schiex, M. (2005). "On the Acceptability of Arguments in Bipolar Argumentation Frameworks". In: *ECSQARU*, pp. 378–389 (cited on pp. 49, 130).

Chang, M., Griffiths, T. L., and Levine, S. (2022). "Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation". In: *arXiv preprint arXiv:2207.00787* (cited on pp. 57, 64, 65, 70, 189).

Chattopadhyay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks". In: *WACV*, pp. 839–847 (cited on p. 129).

Chen, C., Deng, F., and Ahn, S. (2021). "Roots: Object-centric representation and rendering of 3d scenes". In: *Journal of Machine Learning Research* 22.259, pp. 1–36 (cited on pp. 46, 70, 107).

Choi, Y., Uh, Y., Yoo, J., and Ha, J. (2020). "StarGAN v2: Diverse Image Synthesis for Multiple Domains". In: *CVPR*, pp. 8185–8194 (cited on p. 145).

Clark, A. (2013). "Whatever next? Predictive brains, situated agents, and the future of cognitive science". In: *Behavioral and Brain Sciences* 36.3, pp. 181–204 (cited on p. 40).

Crawford, E. and Pineau, J. (2020). "Exploiting spatial invariance for scalable unsupervised object tracking". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 3684–3692 (cited on p. 47).

Čyras, K., Rago, A., Albini, E., Baroni, P., and Toni, F. (2021). "Argumentative XAI: a survey". In: *arXiv preprint arXiv:2105.11266* (cited on p. 49).

Daunhawer, I., Bizeul, A., Palumbo, E., Marx, A., and Vogt, J. E. (2023). "Identifiability results for multimodal contrastive learning". In: *arXiv preprint arXiv:2303.09166* (cited on p. 46).

De Angelis, E., Proietti, M., and Toni, F. (2023). "ABA Learning via ASP". In: *Proceedings 39th International Conference on Logic Programming, ICLP 2023, Imperial College London, UK, 9th July 2023 - 15th July 2023*. Vol. 385, pp. 1–8 (cited on pp. 52, 149).

Deng, F., Zhi, Z., Lee, D., and Ahn, S. (2021). "Generative scene graph networks". In: *International Conference on Learning Representations* (cited on p. 109).

Deng, L. (2012). "The mnist database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine* 29.6, pp. 141–142 (cited on pp. 157, 194).

Dhir, A., Sedgwick, R., Kori, A., Glocker, B., and Wilk, M. van der (2024). "Continuous Bayesian Model Selection for Multivariate Causal Discovery". In: *arXiv preprint arXiv:2411.10154* (cited on p. 37).

Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K., and Shanahan, M. (2016). "Deep unsupervised clustering with gaussian mixture variational autoencoders". In: *arXiv preprint arXiv:1611.02648* (cited on pp. 43, 80).

Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al. (2021). "Cogview: Mastering text-to-image generation via transformers". In: *Advances in Neural Information Processing Systems* 34 (cited on p. 44).

Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T. B., and Vanhoucke, V. (2022). "Google scanned objects: A high-quality dataset of 3d scanned household items". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2553–2560 (cited on p. 98).

Dung, P. M. (1995). "On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games". In: *Artif. Intell.* 77.2, pp. 321–358 (cited on p. 49).

Dung, P. M., Kowalski, R. A., and Toni, F. (2009). "Argumentation in AI". In: chap. Assumption-Based Argumentation (cited on p. 51).

Eastwood, C. and Williams, C. K. I. (2018). "A framework for the quantitative evaluation of disentangled representations". In: *International Conference on Learning Representations* (cited on p. 42).

Elsayed, G., Mahendran, A., Steenkiste, S. van, Greff, K., Mozer, M. C., and Kipf, T. (2022). "Savi++: Towards end-to-end object-centric learning from real-world videos". In: *Advances in Neural Information Processing Systems* 35, pp. 28940–28954 (cited on pp. 45, 70, 189).

Emami, P., He, P., Ranka, S., and Rangarajan, A. (2022). "Slot order matters for compositional scene understanding". In: *arXiv preprint arXiv:2206.01370* (cited on pp. 45, 70, 116, 189).

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. (2019). "Genesis: Generative scene inference and sampling with object-centric latent representations". In: *arXiv preprint arXiv:1907.13052* (cited on pp. 45, 57, 70, 108, 189).

Engelcke, M., Parker Jones, O., and Posner, I. (2021). "Genesis-v2: Inferring unordered object representations without iterative refinement". In: *Advances in Neural Information Processing Systems* 34, pp. 8085–8094 (cited on pp. 45, 57, 70, 107, 189).

Epstein, R. A., Patai, E. Z., Julian, J. B., and Spiers, H. J. (2017). "The cognitive map in humans: spatial navigation and beyond". In: *Nature neuroscience* 20.11, pp. 1504–1513 (cited on pp. 30, 39).

Eslami, S. A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. (2018a). "Neural scene representation and rendering". In: *Science* 360.6394, pp. 1204–1210 (cited on p. 108).

Eslami, S. A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. (2018b). "Neural scene representation and rendering". In: *Science* 360.6394, pp. 1204–1210 (cited on p. 193).

Esser, P., Rombach, R., and Ommer, B. (2021). "Taming transformers for high-resolution image synthesis". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883 (cited on p. 44).

Falck, F., Zhang, H., Willetts, M., Nicholson, G., Yau, C., and Holmes, C. C. (2021). "Multi-facet clustering variational autoencoders". In: *Advances in Neural Information Processing Systems* 34, pp. 8676–8690 (cited on p. 72).

Fodor, J. (1975). "The language of thought" (cited on p. 108).

Friston, K. (2010). "The free-energy principle: A unified brain theory?" In: *Nature Reviews Neuroscience* 11.2, pp. 127–138 (cited on p. 40).

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., O'Doherty, J., and Pezzulo, G. (2017). "Active inference: A process theory". In: *Neural Computation* 29.1, pp. 1–49 (cited on p. 32).

Gao, G., Liu, W., Chen, A., Geiger, A., and Schölkopf, B. (2024). "Graphdreamer: Compositional 3d scene synthesis from scene graphs". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21295–21304 (cited on pp. 108, 109).

Gebser, M., Kaminski, R., Kaufmann, B., and Schaub, T. (2019). "Multi-shot ASP solving with clingo". In: *Theory Pract. Log. Program.* 19.1, pp. 27–82 (cited on p. 151).

Gerstenberg, T., Goodman, N. D., Lagnado, D. A., and Tenenbaum, J. B. (2021). "A counterfactual simulation model of causal judgments for physical events." In: *Psychological review* 128.5, p. 936 (cited on pp. 31, 71).

Getoor, L., Friedman, N., Koller, D., and Pfeffer, A. (2001). "Learning probabilistic relational models". In: *Relational data mining*, pp. 307–335 (cited on p. 109).

Getoor, L. and Taskar, B. (2007). "Introduction to statistical relational learning". MIT press (cited on p. 109).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (cited on p. 124).

Gopnik, A., Glymour, C., Sobel, D. M., Schulz, L. E., Kushnir, T., and Danks, D. (2004). "A theory of causal learning in children: causal maps and Bayes nets." In: *Psychological review* 111.1, p. 3 (cited on pp. 31, 71).

Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D. J., Gnanapragasam, D., Golemo, F., Herrmann, C., et al. (2022). "Kubric: A scalable dataset generator". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761 (cited on pp. 98, 125, 246).

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. (2019). "Multi-object representation learning with iterative variational inference". In: *International Conference on Machine Learning*. PMLR, pp. 2424–2433 (cited on pp. 45, 62, 70, 93, 107).

Greff, K., Van Steenkiste, S., and Schmidhuber, J. (2017). "Neural expectation maximization". In: *Advances in Neural Information Processing Systems* 30 (cited on p. 121).

Greff, K., Van Steenkiste, S., and Schmidhuber, J. (2020). "On the binding problem in artificial neural networks". In: *arXiv preprint arXiv:2012.05208* (cited on pp. 31, 45, 57, 71, 107).

Gregory, R. L. (1980). "Perceptions as hypotheses". In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 290.1038, pp. 181–197 (cited on p. 39).

Gresele, L., Rubenstein, P. K., Mehrjou, A., Locatello, F., and Schölkopf, B. (2020). "The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica". In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 217–227 (cited on p. 46).

Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., and Ling, M. (2019). "Scene graph generation with external knowledge and image reconstruction". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1969–1978 (cited on p. 108).

Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., and Guo, B. (2021). "Vector quantized diffusion model for text-to-image synthesis". In: *arXiv preprint arXiv:2111.14822* (cited on p. 44).

Harnad, S. (1990). "The symbol grounding problem". In: *Physica D: Nonlinear Phenomena* 42.1-3, pp. 335–346 (cited on pp. 31, 56).

Hartley, R. and Zisserman, A. (2003). "Multiple view geometry in computer vision". Cambridge university press (cited on p. 97).

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep Residual Learning for Image Recognition". In: *CVPR*, pp. 770–778 (cited on pp. 131, 146, 153).

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in neural information processing systems* 30 (cited on pp. 65, 85).

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations* (cited on p. 42).

Higgins, I., Racanière, S., and Rezende, D. (2022). "Symmetry-Based Representations for Artificial and Biological General Intelligence". In: *Frontiers in Computational Neuroscience* 16 (cited on p. 42).

Hinton, G. (1979). "Some demonstrations of the effects of structural descriptions in mental imagery". In: *Cognitive Science* 3.3, pp. 231–250 (cited on pp. 30, 106).

Hinton, G. (2022). "How to represent part-whole hierarchies in a neural network". In: *Neural Computation*, pp. 1–40 (cited on pp. 30, 33, 57).

Hinton, G. E., Sabour, S., and Frosst, N. (2018). "Matrix capsules with EM routing". In: *International conference on learning representations* (cited on p. 45).

Hoffman, M. D. and Johnson, M. J. (2016). "Elbo surgery: yet another way to carve up the variational evidence lower bound". In: *Workshop in Advances in Approximate Bayesian Inference, NIPS*. Vol. 1. 2 (cited on pp. 79, 94, 227, 253).

Huang, G., Liu, Z., Maaten, L. van der, and Weinberger, K. Q. (2017). "Densely Connected Convolutional Networks". In: *CVPR*, pp. 2261–2269 (cited on pp. 131, 146).

Hyvarinen, A. and Morioka, H. (2016). "Unsupervised feature extraction by time-contrastive learning and nonlinear ica". In: *Advances in neural information processing systems* 29 (cited on p. 42).

Hyvarinen, A., Sasaki, H., and Turner, R. (2019a). "Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning". In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Vol. 89, pp. 859–868 (cited on p. 43).

Hyvarinen, A., Sasaki, H., and Turner, R. (2019b). "Nonlinear ICA using auxiliary variables and generalized contrastive learning". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 859–868 (cited on pp. 42, 82).

Hyvärinen, A., Khemakhem, I., and Monti, R. (2023). "Identifiability of latent-variable and structural-equation models: from linear to nonlinear" (cited on p. 32).

Hyvärinen, A. and Oja, E. (2000). "Independent component analysis: algorithms and applications". In: *Neural networks* 13.4-5, pp. 411–430 (cited on pp. 42, 71, 107).

Hyvärinen, A. and Pajunen, P. (1999). "Nonlinear independent component analysis: Existence and uniqueness results". In: *Neural networks* 12.3, pp. 429–439 (cited on pp. 42, 71, 107).

Jacob, A. R., Kori, A., Angelis, E. D., Glocker, B., Proietti, M., and Toni, F. (2023). "Object-Centric Neuro-Argumentative Learning". In: *The Fourth International Joint Conference on Learning and Reasoning (IJCLR 2024)* (cited on p. 35).

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). "Spatial transformer networks". In: *Advances in neural information processing systems* 28 (cited on p. 91).

Jang, E., Gu, S., and Poole, B. (2016). "Categorical reparameterization with gumbel-softmax". In: *arXiv preprint arXiv:1611.01144* (cited on pp. 44, 45, 141, 199).

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. (2017). "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910 (cited on pp. 11, 64, 85, 191).

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M., and Fei-Fei, L. (2015). "Image retrieval using scene graphs". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678 (cited on p. 108).

Kabra, R., Burgess, C., Matthey, L., Kaufman, R. L., Greff, K., Reynolds, M., and Lerchner, A. (2019). "Multi-Object Datasets". https://github.com/deepmind/multi-object-datasets/ (cited on pp. 11, 64, 85, 193).

Kakogeorgiou, I., Gidaris, S., Karantzalos, K., and Komodakis, N. (2024). "SPOT: Self-Training with Patch-Order Permutation for Object-Centric Learning with Autoregressive Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22776–22786 (cited on p. 88).

Kant, I. (1908). "Critique of pure reason. 1781". In: *Modern Classical Philosophers, Cambridge, MA: Houghton Mifflin*, pp. 370–456 (cited on p. 108).

Karras, T. (2019). "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *arXiv preprint arXiv:1812.04948* (cited on pp. 124, 145).

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119 (cited on pp. 11, 64, 194).

Kazemi, S. M. and Poole, D. (2018). "RelNN: A deep neural model for relational learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1 (cited on p. 109).

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. (2020a). "Variational autoencoders and nonlinear ica: A unifying framework". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2207–2217 (cited on pp. 42, 43, 82, 91, 106, 226, 237).

Khemakhem, I., Monti, R., Kingma, D., and Hyvarinen, A. (2020b). "ICE-BeeM: Identifiable Conditional Energy-Based Deep Models Based on Nonlinear ICA".

In: *Advances in Neural Information Processing Systems*. Vol. 33 (cited on pp. 43, 226, 237).

Khemakhem, I., Monti, R., Kingma, D., and Hyvarinen, A. (2020c). "Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica". In: *Advances in Neural Information Processing Systems* 33, pp. 12768–12778 (cited on pp. 43, 106, 188, 237).

Kim, H. and Mnih, A. (2018a). "Disentangling by Factorising". In: *Proceedings of the 35th International Conference on Machine Learning* (cited on pp. 42, 107).

Kim, H. and Mnih, A. (2018b). "Disentangling by factorising". In: *International Conference on Machine Learning*. PMLR, pp. 2649–2658 (cited on p. 193).

Kim, K., Yoon, K., Jeon, J., In, Y., Moon, J., Kim, D., and Park, C. (2024). "LLM4SGG: Large Language Models for Weakly Supervised Scene Graph Generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28306–28316 (cited on p. 108).

Kingma, D. P. and Welling, M. (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (cited on pp. 30, 40, 42, 81, 90, 106, 124).

Kipf, T., Elsayed, G. F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., and Greff, K. (2021). "Conditional object-centric learning from video". In: *arXiv preprint arXiv:2111.12594* (cited on pp. 45, 47, 70, 189).

Kivva, B., Rajendran, G., Ravikumar, P., and Aragam, B. (2022). "Identifiability of deep generative models without auxiliary information". In: *Advances in Neural Information Processing Systems* 35, pp. 15687–15701 (cited on pp. 43, 46, 72, 82, 84, 115, 120, 226, 232–235, 256, 257, 260, 262, 263).

Knill, D. C. and Pouget, A. (2004). "The Bayesian brain: The role of uncertainty in neural coding and computation". In: *Trends in Neurosciences* 27.12, pp. 712–719 (cited on p. 39).

Kori, A., Balsells-Rodas, C., Glocker, B., Li, Y., and Locatello, F. (2025a). "Causal Representation Learning and Inference via Mixture-Based Priors". In: *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy* (cited on p. 36).

Kori, A., Glocker, B., Schölkopf, B., and Locatello, F. (2025b). "Unifying Causal and Object-centric Representation Learning allows Causal Composition". In: *ICLR*

2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy (cited on p. 35).

Kori, A., Glocker, B., and Toni, F. (2023a). "GLANCE: Global to Local Architecture-Neutral Concept-based Explanations". In: *XAI workshop, Part of Advances in Neural Information Processing Systems 36 (NeurIPS 2023)* (cited on p. 36).

Kori, A., Glocker, B., and Toni, F. (2024a). "Explaining Image Classifiers with Visual Debates". In: *DS*, pp. 200–214 (cited on pp. 36, 130, 131, 140, 142, 144, 146).

Kori, A., Locatello, F., Ribeiro, F. D. S., Toni, F., and Glocker, B. (2024b). "Grounded Object-Centric Learning". In: *The Twelfth International Conference on Learning Representations (ICLR 2024)* (cited on pp. 34, 35, 70, 80, 107, 116, 124, 269).

Kori, A., Locatello, F., Toni, F., Glocker, B., and Ribeiro, F. D. S. (2024c). "Identifiable object centric representations via probabilistic slot attention". In: *Part of Advances in Neural Information Processing Systems 37 (NeurIPS 2024)* (cited on pp. 35, 91, 94, 98, 99, 107, 116, 121, 245, 252, 253, 257, 260).

Kori, A., Rago, A., and Toni, F. (2025c). "Free Argumentative Exchanges for Explaining Image Classifiers". In: *The 24th International Conference on Autonomous Agents and Multi-Agent Systems* (cited on p. 35).

Kori, A., Sanchez, P., Vilouras, K., Glocker, B., and Tsaftaris, S. A. (2023b). "A Causal Ordering Prior for Unsupervised Representation Learning" (cited on p. 36).

Kori, A., Toni, F., and Glocker, B. (2025d). "Identifiable Object Representations under Spatial Ambiguities". In: *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy* (cited on p. 35).

Kosiorek, A., Kim, H., Teh, Y. W., and Posner, I. (2018). "Sequential attend, infer, repeat: Generative modelling of moving objects". In: *Advances in Neural Information Processing Systems* 31 (cited on pp. 47, 108).

Kosiorek, A. R. (2019). "Learning Object-Centric Representations". PhD thesis. University of Oxford (United Kingdom) (cited on p. 39).

Krimmel, M., Achterhold, J., and Stueckler, J. (2024). "Attention Normalization Impacts Cardinality Generalization in Slot Attention". In: *arXiv preprint arXiv:2407.04170* (cited on p. 121).

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). "Visual genome: Connecting language

and vision using crowdsourced dense image annotations". In: *International journal of computer vision* 123, pp. 32–73 (cited on p. 108).

Kügelgen, J. von, Ustyuzhaninov, I., Gehler, P., Bethge, M., and Schölkopf, B. (2020). "Towards causal generative scene models via competition of experts". In: *arXiv preprint arXiv:2004.12906* (cited on pp. 108, 109).

Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. (2015). "Deep convolutional inverse graphics network". In: *Advances in neural information processing systems* 28 (cited on p. 30).

Lachapelle, S., López, P. R., Sharma, Y., Everett, K., Priol, R. L., Lacoste, A., and Lacoste-Julien, S. (2024a). "Nonparametric Partial Disentanglement via Mechanism Sparsity: Sparse Actions, Interventions and Sparse Temporal Dependencies". In: *arXiv preprint arXiv:2401.04890* (cited on p. 42).

Lachapelle, S., Mahajan, D., Mitliagkas, I., and Lacoste-Julien, S. (2023). "Additive decoders for latent variables identification and cartesian-product extrapolation". In: *arXiv preprint arXiv:2307.02598* (cited on p. 247).

Lachapelle, S., Mahajan, D., Mitliagkas, I., and Lacoste-Julien, S. (2024b). "Additive decoders for latent variables identification and cartesian-product extrapolation". In: *Advances in Neural Information Processing Systems* 36 (cited on pp. 71, 82, 88, 107, 225).

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). "Building machines that learn and think like people". In: *Behavioral and brain sciences* 40, e253 (cited on pp. 30, 31, 71, 106, 124).

Lakkaraju, H., Slack, D., Chen, Y., Tan, C., and Singh, S. (2022). "Rethinking Explainability as a Dialogue: A Practitioner's Perspective". In: *CoRR* abs/2202.01875 (cited on p. 129).

Li, K., Wu, Z., Peng, K.-C., Ernst, J., and Fu, Y. (2018). "Tell me where to look: Guided attention inference network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9215–9223 (cited on p. 88).

Li, N., Eastwood, C., and Fisher, R. (2020). "Learning object-centric representations of multi-object scenes from multiple views". In: *Advances in Neural Information Processing Systems* 33, pp. 5656–5666 (cited on pp. 46, 47, 70, 98, 99).

Li, N., Raza, M. A., Hu, W., Sun, Z., and Fisher, R. (2021). "Object-centric representation learning with generative spatial-temporal factorization". In: *Advances in neural information processing systems* 34, pp. 10772–10783 (cited on pp. 47, 70).

Li, R., Zhang, S., and He, X. (2022). "Sgtr: End-to-end scene graph generation with transformer". In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19486–19496 (cited on p. 108).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, pp. 740–755 (cited on pp. 11, 64, 194).

Lin, Z., Wu, Y.-F., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. (2020). "Space: Unsupervised object-oriented scene representation via spatial attention and decomposition". In: *arXiv preprint arXiv:2001.02407* (cited on pp. 45, 70, 189).

Lippe, P., Magliacane, S., Löwe, S., Asano, Y. M., Cohen, T., and Gavves, S. (2022). "Citris: Causal identifiability from temporal intervened sequences". In: *International Conference on Machine Learning*. PMLR, pp. 13557–13603 (cited on p. 43).

Liu, X., Sanchez, P., Thermos, S., O'Neil, A. Q., and Tsaftaris, S. A. (2022). "Learning disentangled representations in the imaging domain". In: *Medical Image Analysis* 80 (cited on p. 42).

Liu, Y., Jia, B., Chen, Y., and Huang, S. (2025). "SlotLifter: Slot-guided Feature Lifting for Learning Object-centric Radiance Fields". In: *European Conference on Computer Vision*. Springer, pp. 270–288 (cited on pp. 46, 70).

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *international conference on machine learning*. PMLR, pp. 4114–4124 (cited on pp. 42, 71, 80, 107).

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. (2020a). "Weakly-supervised disentanglement without compromises". In: *International Conference on Machine Learning*. PMLR, pp. 6348–6359 (cited on pp. 42, 43).

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. (2020b). "Object-centric learning with slot attention". In: *Advances in Neural Information Processing Systems* 33, pp. 11525–11538 (cited on pp. 30, 45, 47, 57, 58, 62, 64, 70, 73, 74, 86, 92, 99, 107, 108, 116, 149, 150, 189, 269, 270).

Löwe, S., Lippe, P., Locatello, F., and Welling, M. (2024). "Rotating features for object discovery". In: *Advances in Neural Information Processing Systems* 36 (cited on pp. 30, 70, 86, 107).

Lundberg, S. M. and Lee, S. (2017). "A Unified Approach to Interpreting Model Predictions". In: *NIPS*, pp. 4765–4774 (cited on pp. 128, 137, 146).

Luo, R., Yu, H.-X., and Wu, J. (2024). "Unsupervised Discovery of Object-Centric Neural Fields". In: *arXiv preprint arXiv:2402.07376* (cited on pp. 46, 47, 70).

Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). "The concrete distribution: A continuous relaxation of discrete random variables". In: *arXiv preprint arXiv:1611.00712* (cited on pp. 44, 45, 199).

Madumal, P., Miller, T., Sonenberg, L., and Vetere, F. (2019). "A Grounded Interaction Protocol for Explainable Artificial Intelligence". In: *AAMAS*, pp. 1033–1041 (cited on p. 129).

Mansouri, A., Hartford, J., Ahuja, K., and Bengio, Y. (2022). "Object-centric causal representation learning". In: *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations* (cited on p. 124).

Mansouri, A., Hartford, J., Zhang, Y., and Bengio, Y. (2023). "Object-centric architectures enable efficient causal representation learning". In: *arXiv preprint arXiv:2310.19054* (cited on p. 124).

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. (2019a). "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision". In: *arXiv preprint arXiv:1904.12584* (cited on p. 124).

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. (2019b). "The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision". In: *ICLR* (cited on p. 149).

Marcus, G. F. (2003). "The algebraic mind: Integrating connectionism and cognitive science". MIT press (cited on pp. 31, 71).

Marino, J., Yue, Y., and Mandt, S. (2018). "Iterative amortized inference". In: *International Conference on Machine Learning*. PMLR, pp. 3403–3412 (cited on p. 45).

Mathieu, E., Rainforth, T., Siddharth, N., and Teh, Y. W. (2019). "Disentangling Disentanglement in Variational Autoencoders". In: *Proceedings of the 36th International Conference on Machine Learning* (cited on p. 42).

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2021). "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM* 65.1, pp. 99–106 (cited on p. 70).

Miller, T. (2019). "Explanation in artificial intelligence: Insights from the social sciences". In: *Artif. Intell.* 267, pp. 1–38 (cited on p. 129).

Mirza, M. and Osindero, S. (2014). "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (cited on p. 124).

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. (2014). "Recurrent Models of Visual Attention". In: *NIPS*, pp. 2204–2212 (cited on p. 143).

Morioka, H. and Hyvärinen, A. (2023). "Causal representation learning made identifiable by grouping of observational variables". In: *arXiv preprint arXiv:2310.15709* (cited on pp. 111, 116, 122).

Mozafari, M. (2020). "Bitmoji faces" (cited on pp. 11, 64, 194).

Nazaret, A., Hong, J., Azizi, E., and Blei, D. (2023). "Stable differentiable causal discovery". In: *arXiv preprint arXiv:2311.10263* (cited on p. 119).

Neal, R. M. (1996). "Bayesian Learning for Neural Networks". Springer-Verlag (cited on p. 80).

Parr, T. and Friston, K. J. (2018). "The anatomy of inference: Generative models and brain structure". In: *Frontiers in Computational Neuroscience* 12, p. 90 (cited on p. 40).

Pearl, J. (2009). "Causality". 2nd ed. Cambridge University Press (cited on p. 112).

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. (2016). "A benchmark dataset and evaluation methodology for video object segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732 (cited on p. 214).

Potyka, N. (2021). "Interpreting Neural Networks as Quantitative Argumentation Frameworks". In: *AAAI*, pp. 6463–6470 (cited on p. 140).

Proietti, M. and Toni, F. (2023). "A Roadmap for Neuro-argumentative Learning". In: *NeSy* (cited on p. 124).

Rago, A., Baroni, P., and Toni, F. (2022). "Explaining Causal Models with Argumentation: the Case of Bi-variate Reinforcement". In: *KR* (cited on p. 139).

Rago, A., Li, H., and Toni, F. (2023). "Interactive Explanations by Conflict Resolution via Argumentative Exchanges". In: *KR*, pp. 582–592 (cited on pp. 50, 51, 129–135).

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). "Zero-shot text-to-image generation". In: *International Conference on Machine Learning*. PMLR, pp. 8821–8831 (cited on p. 44).

Reizinger, P. and Krishnan, R. (2024). "Exploring A Bayesian View On Compositional and Counterfactual Generalization". In: *NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward* (cited on p. 124).

Reizinger, P., Sharma, Y., Bethge, M., Schölkopf, B., Huszár, F., and Brendel, W. (2023). "Multivariable Causal Discovery with General Nonlinear Relationships". In: *UAI 2022 Workshop on Causal Representation Learning* (cited on p. 117).

Revonsuo, A. and Newman, J. (1999). "Binding and consciousness". In: *Consciousness and cognition* 8.2 (cited on pp. 31, 57).

Ribeiro, F. D. S., Leontidis, G., and Kollias, S. (2020). "Capsule routing via variational bayes". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34, pp. 3749–3756 (cited on p. 45).

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *SIGKDD*, pp. 1135–1144 (cited on pp. 128, 136, 146).

Rock, I. (1973). "Orientation and form". In: *(No Title)* (cited on pp. 30, 106).

Sabour, S., Frosst, N., and Hinton, G. E. (2017). "Dynamic routing between capsules". In: *Advances in neural information processing systems* 30 (cited on p. 45).

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). "Causal protein-signaling networks derived from multiparameter single-cell data". In: *Science* 308.5721, pp. 523–529 (cited on p. 115).

Santhirasekaram, A., Kori, A., Winkler, M., Rockall, A., and Glocker, B. (2022). "Vector Quantisation for Robust Segmentation". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part IV*. Springer, pp. 663–672 (cited on pp. 36, 44).

Santhirasekaram, A., Kori, A., Winkler, M., Rockall, A., Toni, F., and Glocker, B. (2023). "Robust Hierarchical Symbolic Explanations in Hyperbolic Space for

Image Classification". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 561–570 (cited on pp. 36, 44).

Sattarzadeh, S., Sudhakar, M., Plataniotis, K. N., Jang, J., Jeong, Y., and Kim, H. (2021). "Integrated Grad-Cam: Sensitivity-Aware Visual Explanation of Deep Convolutional Networks Via Integrated Gradient-Based Scoring". In: *ICASSP*, pp. 1775–1779 (cited on p. 129).

Schölkopf, B. and Kügelgen, J. von (2022). "From statistical to causal learning". In: *Proceedings of the International Congress of Mathematicians* (cited on pp. 30, 71, 106).

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021). "Toward Causal Representation Learning". In: *Proceedings of the IEEE* 109 (cited on pp. 106, 124).

Seitzer, M., Horn, M., Zadaianchuk, A., Zietlow, D., Xiao, T., Simon-Gabriel, C.-J., He, T., Zhang, Z., Schölkopf, B., Brox, T., et al. (2022). "Bridging the gap to real-world object-centric learning". In: *arXiv preprint arXiv:2209.14860* (cited on pp. 57, 64, 65, 70, 72, 86, 88, 189, 201, 210, 269, 270).

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *ICCV*, pp. 618–626 (cited on pp. 129, 137, 146).

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). "Learning Important Features Through Propagating Activation Differences". In: *ICML*, pp. 3145–3153 (cited on pp. 128, 137, 146).

Singh, G., Deng, F., and Ahn, S. (2021). "Illiterate dall-e learns to compose". In: *arXiv preprint arXiv:2110.11405* (cited on pp. 45, 57, 66, 70, 72, 87, 108, 189, 216).

Singh, G., Kim, Y., and Ahn, S. (2022a). "Neural Block-Slot Representations". In: *arXiv preprint arXiv:2211.01177* (cited on pp. 45, 64).

Singh, G., Wu, Y.-F., and Ahn, S. (2022b). "Simple unsupervised object-centric learning for complex and naturalistic videos". In: *Advances in Neural Information Processing Systems* 35, pp. 18181–18196 (cited on p. 72).

Song, Y. and Ermon, S. (2019). "Generative modeling by estimating gradients of the data distribution". In: *Advances in neural information processing systems* 32 (cited on p. 124).

Srivastava, N., Mansimov, E., and Salakhudinov, R. (2015). "Unsupervised learning of video representations using lstms". In: *International conference on machine learning*. PMLR, pp. 843–852 (cited on p. 194).

Stammer, W., Schramowski, P., and Kersting, K. (2021). "Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3619–3629 (cited on pp. 11, 152, 157, 192, 193).

Takida, Y., Shibuya, T., Liao, W., Lai, C.-H., Ohmura, J., Uesaka, T., Murata, N., Takahashi, S., Kumakura, T., and Mitsufuji, Y. (2022). "SQ-VAE: Variational Bayes on Discrete Representation with Self-annealed Stochastic Quantization". In: *arXiv preprint arXiv:2205.07547* (cited on pp. 200, 223).

Tangemann, M., Schneider, S., Von Kügelgen, J., Locatello, F., Gehler, P., Brox, T., Kümmerer, M., Bethge, M., and Schölkopf, B. (2021). "Unsupervised object learning via common fate". In: *arXiv preprint arXiv:2110.06562* (cited on p. 108).

Tarlé, L. D. de, Bonzon, E., and Maudet, N. (2022). "Multiagent Dynamics of Gradual Argumentation Semantics". In: *AAMAS*, pp. 363–371 (cited on p. 51).

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., and Goodman, N. D. (2011). "How to grow a mind: Statistics, structure, and abstraction". In: *science* 331.6022, pp. 1279–1285 (cited on pp. 30, 108, 124).

Teney, D., Liu, L., and Den Hengel, A. van (2017). "Graph-structured representations for visual question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9 (cited on p. 108).

Thomas, D., Jiang, J., Kori, A., Russo, A., Winkler, S., Sale, S., McMillan, J., Belardinelli, F., and Rago, A. (2025). "Explainable Reinforcement Learning for Formula One Race Strategy". In: *arXiv preprint arXiv:2501.04068* (cited on p. 37).

Tomczak, J. and Welling, M. (2018). "VAE with a VampPrior". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1214–1223 (cited on pp. 79, 200).

Träuble, F., Goyal, A., Rahaman, N., Mozer, M., Kawaguchi, K., Bengio, Y., and Schölkopf, B. (2022). "Discrete key-value bottleneck". In: *arXiv preprint arXiv:2207.11240* (cited on pp. 44, 200).

Treisman, A. (1999). "Solutions to the binding problem: progress through controversy and convergence". In: *Neuron* 24.1, pp. 105–125 (cited on pp. 58, 61).

Van Den Oord, A., Vinyals, O., et al. (2017). "Neural discrete representation learning". In: *Advances in neural information processing systems* 30 (cited on pp. 44, 79, 131, 141, 199, 200).

Van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. (2018). "Relational neural expectation maximization: Unsupervised discovery of objects and their interactions". In: *arXiv preprint arXiv:1802.10353* (cited on p. 47).

Van Steenkiste, S., Kurach, K., Schmidhuber, J., and Gelly, S. (2020). "Investigating object compositionality in generative adversarial networks". In: *Neural Networks* 130, pp. 309–325 (cited on pp. 45, 70, 189).

Varici, B., Acarturk, E., Shanmugam, K., Kumar, A., and Tajer, A. (2023). "Score-based Causal Representation Learning with Interventions". In: *arXiv preprint arXiv:2301.08230* (cited on p. 43).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30 (cited on pp. 48, 75, 93).

Von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. (2021). "Self-supervised learning with data augmentations provably isolates content from style". In: *Advances in neural information processing systems* 34, pp. 16451–16467 (cited on pp. 42, 46, 112, 122).

Vroomen, J. and Keetels, M. (2010). "Perception of intersensory synchrony: a tutorial review". In: *Attention, Perception, & Psychophysics* 72.4, pp. 871–884 (cited on p. 57).

Wang, P. and Vasconcelos, N. (2019). "Deliberative Explanations: visualizing network insecurities". In: *NeurIPS*, pp. 1372–1383 (cited on p. 129).

Wang, X., Li, Y., Zhang, Q., Wu, W., Li, M. J., Ma, L., and Jiang, J. (2024). "Weakly-Supervised 3D Scene Graph Generation via Visual-Linguistic Assisted Pseudo-labeling". In: *IEEE Transactions on Multimedia* (cited on p. 108).

Wang, Y., Liu, L., and Dauwels, J. (2023a). "Slot-VAE: Object-Centric Scene Generation with Slot Attention". In: *arXiv preprint arXiv:2306.06997* (cited on pp. 45, 64, 70, 81, 92, 116, 189).

Wang, Y., Liu, L., and Dauwels, J. (2023b). "Slot-VAE: Object-Centric Scene Generation with Slot Attention". In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202, pp. 36020–36035 (cited on pp. 75, 79).

Watters, N., Matthey, L., Borgeaud, S., Kabra, R., and Lerchner, A. (2019a). "Spriteworld: A Flexible, Configurable Reinforcement Learning Environment" (cited on p. 86).

Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. (2019b). "Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes". In: *arXiv preprint arXiv:1901.07017* (cited on p. 85).

Wei, Y., Min, M. R., Vosselman, G., Li, L. E., and Yang, M. Y. (2024). "Compositional 3D Scene Synthesis with Scene Graph Guided Layout-Shape Generation". In: *arXiv preprint arXiv:2403.12848* (cited on p. 108).

Wertheimer, M. (1923). "Laws of organization in perceptual forms". In: *Psychologische Forschung* 4, pp. 301–350 (cited on p. 39).

Wiedemer, T., Brady, J., Panfilov, A., Juhos, A., Bethge, M., and Brendel, W. (2023). "Provable Compositional Generalization for Object-Centric Learning". In: *arXiv preprint arXiv:2310.05327* (cited on pp. 108, 119, 120, 123).

Willetts, M. and Paige, B. (2021). "I Don't Need u: Identifiable Non-Linear ICA Without Side Information". In: *arXiv preprint arXiv:2106.05238* (cited on pp. 43, 46, 106).

Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., and Wang, J. (2021). "Causalvae: Disentangled representation learning via neural structural causal models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9593–9602 (cited on p. 43).

Yang, X., Wang, Y., Sun, J., Zhang, X., Zhang, S., Li, Z., and Yan, J. (2022). "Nonlinear ICA Using Volume-Preserving Transformations". In: *International Conference on Learning Representations* (cited on p. 42).

Yao, D., Rancati, D., Cadei, R., Fumero, M., and Locatello, F. (2024a). "Unifying Causal Representation Learning with the Invariance Principle". In: *arXiv preprint arXiv:2409.02772* (cited on pp. 43, 106, 122).

Yao, D., Xu, D., Lachapelle, S., Magliacane, S., Taslakian, P., Martius, G., Kügelgen, J. von, and Locatello, F. (2023). "Multi-view causal representation learning with partial observability". In: *arXiv preprint arXiv:2311.04056* (cited on pp. 111, 122).

Yao, D., Xu, D., Lachapelle, S., Magliacane, S., Taslakian, P., Martius, G., Kügelgen, J. von, and Locatello, F. (2024b). "Multi-view causal representation learning with partial observability". In: (cited on p. 42).

Yao, T., Pan, Y., Li, Y., and Mei, T. (2018). "Exploring visual relationship for image captioning". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 684–699 (cited on p. 108).

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. (2021). "Vector-quantized image modeling with improved VQGAN". In: *arXiv preprint arXiv:2110.04627* (cited on p. 44).

Yuan, J., Chen, T., Shen, Z., Li, B., and Xue, X. (2024). "Unsupervised Object-Centric Learning From Multiple Unspecified Viewpoints". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (cited on pp. 70, 91, 99, 109, 121).

Yuille, A. and Kersten, D. (2006). "Vision as Bayesian inference: analysis by synthesis?" In: *Trends in cognitive sciences* 10.7, pp. 301–308 (cited on pp. 30, 39).

Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., et al. (2018). "Relational deep reinforcement learning". In: *arXiv preprint arXiv:1806.01830* (cited on p. 109).

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *CVPR* (cited on p. 32).

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. (2021). "Contrastive learning inverts the data generating process". In: *International Conference on Machine Learning*. PMLR, pp. 12979–12990 (cited on p. 42).

This page is intentionally left blank.

Part-3

Appendix

Yatra prayogāḥ sutarkāṣca, prastāvāḥ ca vicāritāḥ|
Vistārāya vijñānasya, tatraiṣaḥ pariśiṣṭakaḥ||

"Where additional experiments, reasoning, and extended propositions are explored — There begins this appendix."

A

COSA-APPENDIX

A.1 ASSUMPTIONS

Now, we list all the assumptions for modeling our slot dictionary and also discuss the how some of these assumptions are implicitly made in previous baselines (ref. Table A.1.

Assumption A.1 (Representation separation)**.** *Object level separation present in observational space can be observed in the latent space of the model (*i.e., *if* $\mathbf{x} = \bigcup_i^K \{O_i\}$*, then* $\mathbf{z} = \bigcup_i^K \{Oz_i\}$*, where* O_i *and* Oz_i *correspond to individual object representation in observational and latent space respectively).*

Remark A.1. *In practice, when features are learned with an end-to-end objective, object-level representation emerges due to iterative attention.*

Assumption A.2 (Representation overcrowding)**.** *The latent representation in* $\mathbf{z} \in \mathbb{R}^{N \times d_z}$ *can accurately recover K slots when* $K \leq N$*. In other words,* $\mathbf{z}_i \in \mathbb{R}^{d_z}$ *is an element of* $\mathbf{z}$ *that can only encode the knowledge of a single object.*

Remark A.2. *Verifying this property is difficult. However, the latent layer for CoSA can be empirically estimated to minimize this effect.*

Assumption A.3 (Object level sufficiency)**.** *We assume that there are no additional objects in the original data distributions other than the ones expressed in training data.*

Remark A.3. *The assumption on object level sufficiency may not be required in SA. Here, this is required as the aim is to learn marginal distributions for every known object in the dataset.*

Assumption A.4 (Sufficient dictionary components)**.** *We assume the slot dictionary to have a sufficient number* $\tilde{M}$ *of* $(\mathfrak{S}^1, \mathfrak{S}^2)$ *pairs to capture the emergent behavior of objectness in the entire dataset, such that* $\tilde{M} \geq K + 1$*.*

Assumption A.5 (Injectivity Assumption)**.** *We assume the resulting decoder model is injective as by construction the architecture follows the properties as expressed in Khemakhem et al., 2020c.*

Table A.1: Comparison between methods and their implicit assumptions and tasks achieved.

Method	Data	Assumption	Tasks	Grounding
Engelcke et al., 2019; Engelcke et al., 2021; Burgess et al., 2019; Wang et al., 2023a; Emami et al., 2022	Image Data	A.1, A.2	Representation, Segregation, and Generation	No
Locatello et al., 2020b; Chang et al., 2022; Seitzer et al., 2022;	Image Data	A.1, A.2	Representation, Segregation	No
Kipf et al., 2021; Elsayed et al., 2022;	Image Data and conditioning info.	A.1, A.2, A.3	Representation, Segregation	No
Lin et al., 2020; Van Steenkiste et al., 2020; Singh et al., 2021	Image Data	A.1, A.2, A.3	Representation, Segregation, and Generation (constr--ained environment or predefined prompts)	No
CoSA	Image Data	A.1, A.2, A.3, A.4	Representation, Segregation, and Generation	Yes

A.2 PROOFS

A.2.1 *Proposition 1: (Object Discovery - ELBO formulation):*

Under a categorical distribution over our discrete latent variables $\tilde{\mathbf{z}}$, and the object-level prior distributions $p(\mathbf{s}_i^0) = \mathcal{N}\left(\mathbf{s}_i^0; \boldsymbol{\mu}_i, \sigma_i^2\right)$ contained in $\mathfrak{S}^2$, we show that variational lower bound on the marginal log-likelihood of $\mathbf{x}$ can be expressed as:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\tilde{\mathbf{z}} \sim q(\tilde{\mathbf{z}}|\mathbf{x}), \mathbf{s}^0 \sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \left[\log p(\mathbf{x} \mid \mathbf{s})\right] - D_{\mathrm{KL}}\left(q\left(\tilde{\mathbf{z}} \mid \mathbf{x}\right)\right) \mid\mid p(\tilde{\mathbf{z}})\right) =: \mathrm{ELBO}(\mathbf{x}), \tag{A.1}$$

where $\mathbf{s} := \prod_{t=1}^{T} \mathcal{H}_\theta\left(\mathbf{s}^{t-1} \mid f(g(\hat{\mathbf{q}}^{t-1}, \mathbf{k}), \mathbf{v})\right)$ denotes the output of the iterative refinement procedure described in Algorithm 1 applied to the initial slot $\mathbf{s}^0$ representations.

Proof. For this proof, we consider the data distribution as $p(\mathbf{x})$, and the aim is to maximize the log-likelihood of this distribution:

$$\log p(\mathbf{x}) = \log \int_{\mathbf{s}} \int_{\tilde{\mathbf{z}}} p(\mathbf{x}, \mathbf{s}, \tilde{\mathbf{z}}) d\mathbf{s} d\tilde{\mathbf{z}}$$

Consider variational distributions $q(\tilde{\mathbf{z}} \mid \mathbf{x})$.

$$= \log \int_{\mathbf{s}} \int_{\tilde{\mathbf{z}}} p(\mathbf{x}, \mathbf{s}, \tilde{\mathbf{z}}) \frac{q(\tilde{\mathbf{z}} \mid \mathbf{x})}{q(\tilde{\mathbf{z}} \mid \mathbf{x})} d\mathbf{s} d\tilde{\mathbf{z}}$$

$$\geq \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log \frac{p(\mathbf{x} \mid \mathbf{s})p(\tilde{\mathbf{z}})}{q(\tilde{\mathbf{z}} \mid \mathbf{x})}$$

$$= \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log \frac{p(\tilde{\mathbf{z}})}{q(\tilde{\mathbf{z}} \mid \mathbf{x})} + \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log p(\mathbf{x} \mid \mathbf{s})$$

$$= \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})} \log \frac{p(\tilde{\mathbf{z}})}{q(\tilde{\mathbf{z}} \mid \mathbf{x})} + \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log p(\mathbf{x} \mid \mathbf{s})$$

$$= \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log p(\mathbf{x} \mid \mathbf{s} = g(\mathbf{s}^0)) - D_{\mathrm{KL}}\Big(q(\tilde{\mathbf{z}} \mid \mathbf{a} = (\mathcal{A} \circ \Phi_e)(\mathbf{x})) \mid\mid p(\tilde{\mathbf{z}})\Big)$$

□

A.2.2 *Proposition 2: (ELBO formulation for reasoning task):*

Under a categorical distribution over our discrete latent variables $\tilde{\mathbf{z}}$, and the object-level prior distributions $p(\mathbf{s}_i^0) = \mathcal{N}\left(\mathbf{s}_i^0; \boldsymbol{\mu}_i, \sigma_i^2\right)$ contained in $\mathfrak{S}^2$, the variational lower bound on the conditional log-likelihood of $\mathbf{y}$ given $\mathbf{x}$ is given by:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log p(\mathbf{y} \mid \mathbf{p}) - D_{\mathrm{KL}}\Big(q(\tilde{\mathbf{z}} \mid \mathbf{x}) \mid\mid p(\tilde{\mathbf{z}})\Big). \quad \text{(A.2)}$$

Proof. The proof is very similar to the proof of proposition 1 . We include this for the sake of completion. For this, we consider categorical conditional distribution as $p(\mathbf{y} \mid \mathbf{x})$, the model segregating slots into given categories. The aim is to maximize the log-likelihood of this distribution:

$$\log p(\mathbf{y} \mid \mathbf{x}) = \log \int_{\mathbf{s}} \int_{\tilde{\mathbf{z}}} p(\mathbf{y}, \mathbf{s}, \tilde{\mathbf{z}} \mid \mathbf{x}) d\mathbf{s} d\tilde{\mathbf{z}}$$

Consider variational distributions $q(\tilde{\mathbf{z}} \mid \mathbf{x})$.

$$= \log \int_{\mathbf{s}} \int_{\tilde{\mathbf{z}}} p(\mathbf{y}, \mathbf{s}, \tilde{\mathbf{z}} \mid \mathbf{x}) \frac{q(\tilde{\mathbf{z}} \mid \mathbf{x})}{q(\tilde{\mathbf{z}} \mid \mathbf{x})} d\mathbf{s} d\tilde{\mathbf{z}}$$

$$\geq \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log \frac{p(\mathbf{y} \mid \mathbf{x}, \mathbf{s}^0)p(\tilde{\mathbf{z}})}{q(\tilde{\mathbf{z}} \mid \mathbf{x})}$$

$$= \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \left[\log \frac{p(\tilde{\mathbf{z}})}{q(\tilde{\mathbf{z}} \mid \mathbf{x})}\right] + \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log p(\mathbf{y} \mid \mathbf{p})$$

$$= \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})} \left[\log \frac{p(\tilde{\mathbf{z}})}{q(\tilde{\mathbf{z}} \mid \mathbf{x})}\right] + \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log p(\mathbf{y} \mid \mathbf{p})$$

$$= \mathbb{E}_{\tilde{\mathbf{z}}\sim q(\tilde{\mathbf{z}}|\mathbf{x})}\mathbb{E}_{\mathbf{s}^0\sim p(\mathbf{s}^0|\tilde{\mathbf{z}})} \log p(\mathbf{y} \mid \mathbf{p}) - D_{\mathrm{KL}}\Big(q(\tilde{\mathbf{z}} \mid \mathbf{a} = (\mathcal{A} \circ \Phi_e)(\mathbf{x})) \mid\mid p(\tilde{\mathbf{z}})\Big)$$

□

A.3 DATASETS

In this work, we use multiple datasets for every case studies. We make use of publically available datasets that are released under MIT Licence and that are open for all research works. We also created two new variants of the MNIST dataset called FloatingMNIST2 (FMNIST2) and FloatingMNIST3 (FMNIST3) for evaluation reasoning/discriminative tasks with the proposed method. We will release the data-generation scripts along with the source code of this project under the open-source MIT License. Table A.2 describes maximum number of objects per image in the considered object discovery datasets. We details on the dataset specifications in this section:

A.3.1 *CLEVR (Johnson et al., 2017)*

CLEVR is diagnostic dataset usually used for benchmarking visual question answering tasks. Here, in our tasks we don't use the question answering part of the dataset, we make use this dataset for its object level compsitionality aspect. This dataset consists of 70000, 15000, and 15000 set of training, validation, and testing images respectively. In our tasks we resize all the

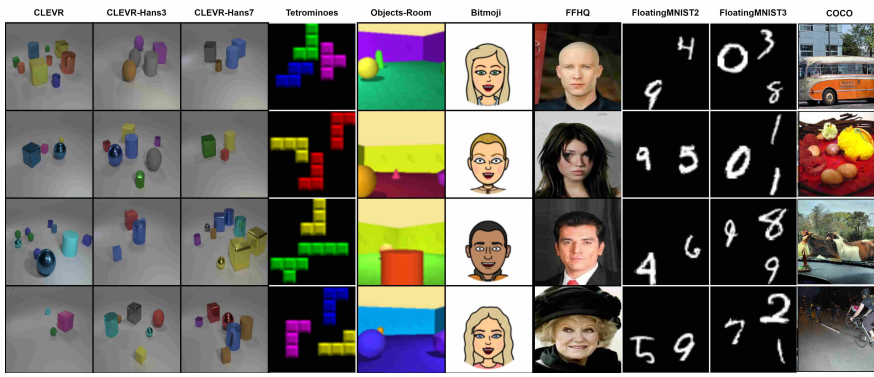


Figure A.1: This image overviews all the datasets used in this work, from left to right columns correspond to CLEVR, CLEVR-Hans3, CLEVR-Hans7, Tetrominoes, Objects-Room, Bitmoji, FFHQ, FloatingMNIST-2, FloatingMNIST-3, and COCO datasets respectively.

images to 64x64 dimension, normalizing them to bound pixel intensities to lie between [0, 1].

A.3.2 *CLEVR-Hans3 (Stammer et al., 2021)*

CLEVR-Hans is a dataset with multiple confouding factors, developed with the soul purpose of investigating reasoning possibilities within the network. The dataset was built on top of CLEVR dataset, where the CLEVR images were further categorized into multiple classes based on object attributes. Images in a particular class are encoded with confounded information with respect some attributes. CLEVR-Hans has in total of 3 classes with the following rules for class1: *Large cube and Large cylinder*, class2: *Small metal cube and Small sphere*, and class3: *Large blue sphere and small yellow sphere*. This dataset contains total of 9000, 2250, and 2250 images in training, validation, and testing set respectively (equally split between different classes). In our tasks we resize all the images to 64x64 dimension, normalizing them to bound pixel intensities to lie between [0, 1].

A.3.3 *CLEVR-Hans7 (Stammer et al., 2021)*

Similar to CLEVR-Hans3 CLEVR-Hans7 is built on top of CLEVR dataset, where the images are categorized into 7 classes based on the object attributes. Here, are the rules used for classifying images: class1: *Large cube and Large cylinder*, class2: *Small metal cube and Small sphere*, class3: *cyan object in front of 2 red objects*, class4: *image with small green, brown, and purple objects with two other small objects*, class5: *3 spheres or 3 spheres with 3 metal cylinders*, class6: *3 metal cylinders*, and class7: *large blue sphere with small yellow sphere*. This dataset contains total of 21000, 5250, and 5250 images in training, validation, and testing set respectively (equally split between different classes). In our tasks we resize all the images to 64x64 dimension, normalizing them to bound pixel intensities to lie between [0, 1].

A.3.4 *Tetrominoes (Kabra et al., 2019)*

Tetrominoes is a Tetris-like shapes dataset. This provides a test bed for non-overlapping self-supervised object discovery. Here, we use around 100000, 10000, and 10000 images for training, validation, and testing. Each image is resized to 32x32 dimension and normalized to bound the intensity values between [0,1]. Each image comprises 3 tetris shapes sampled from 17 unique shapes and orientations, where each object has one of red, green, blue, yellow, magenta, or cyan color.

A.3.5 *Objects-Room (Kabra et al., 2019)*

Objects room is a direct extention of 3Dshapes dataset Kim and Mnih, 2018b, built using MuJoCo environment Eslami et al., 2018b. While the actual dataset has 3 different variants, we make use of single variant with identical object and room variant. In this setting 4-6 objects are placed in the room and have an identical, where the colors are randomly sampled. Here, in our work we use around 100000, 10000, and 10000 images for training, validation, and testing. Each image is resized to 64x64 dimension and normalized to bound the intensity values between [0,1].

A.3.6 *Bitmoji (Mozafari, 2020)*

Bitmoji is cartoon faces dataset for bitmoji mobile application. The images in bitmoji dataset consists 5-point facial landmarks for both male and female faces. In our task we resize all the images to 64x64 dimension and normalize them to bound pixel intensities to lie between [0, 1]. This dataset consists of 4084 images, divided into sets of 3500, 292, and 292 images for training, validation, and testing respectively.

A.3.7 *FFHQ (Karras et al., 2020)*

This dataset includes high quality human faces along with the attributes corresponding to facial features. FFHQ is real world extention of bitmoji datasets. This dataset consists of approximately 200k images of 128x128 resolution with 40 different binary attributes, and the task is to categorize images based on gender (0 = male; 1=female).

A.3.8 *COCO (Lin et al., 2014)*

COCO is a large-scale object recognition dataset with scene-level caption information. It consists of everyday scenes with common objects. The dataset consists of 328k images with varied resolutions. In this work, we resize the images to 224×224 resolution and perform our analysis for scene decomposition. The dataset consists of 91 different objects with around 2.5 million masks.

A.3.9 *FloatingMNIST*

FloatingMNIST is inspired by the Moving MNIST dataset (Srivastava et al., 2015) and is used to benchmark reasoning tasks in the proposed model. FloatingMNIST is made by randomly combining MNIST digits (Deng, 2012). The dataset has three main objectives: addition, subtraction, and mixed. The addition task is to estimate the sum of all the digits present in the given image, the subtraction task is to estimate the absolute difference between

the digits (after ordering them in decreasing order), and finally, the mixed task is to perform addition if the digits are greater than five else performing subtraction.

FloatingMNIST-2: In the case of FloatingMNIST-2, we combine 2 MNIST digits in a large canvas creating the image of 64x64. We propose three variants FloatingMNIST-2-Add, FloatingMNIST-2-Sub, and FloatingMNIST-2-Mixed datasets. The task in FloatingMNIST-2-Add is to estimate the add of 2 digits present in the image. In the case of FloatingMNIST-2-Sub, the task is to estimate difference between 2 digits present in the image. While, In the case of FloatingMNIST-2-Mixed, the task is to sum the digits if both the digits are less than 5 else to estimate the absolute difference between them. We use 60000, 10000, and 10000 training, validation, and testing images respectively.

FloatingMNIST-3: Similar to FloatingMNIST-2, this dataset is made of images with 3 digits. Even this version of dataset consists of three variants of reasoning datasets, namely, FloatingMNIST-3-Add, FloatingMNIST-3-Sub, and FloatingMNIST-3-Mixed. Where the task in FloatingMNIST-3-Add and FloatingMNIST-3-Sub is to estimate sum and absolute difference beween all the digits. In the case of FloatingMNIST-3-Mixed, all digits less than 5 are added while the digits greater than 5 are substracted from the final target. We use 60000, 10000, and 10000 training, validation, and testing images respectively.

Table A.2: Maximum number of objects (including background) observed in a single image per dataset.

Datasets(↓)	Number of Objects
CLEVR	11
Tetrominoes	4
ObjectsRoom	6

A.4 METRICS

All the task specific metrics are described in this section

A.4.1 *Mean squared error (MSE)*

To compute the quality of the reconstructed image we compute pixel wise mean squared distance between the original image and the ground truth image. This metric doesn't provide any information about how good the image is decomposed into objects. As the task is unsupervised in nature, we measure MSE with few other properties to comment on the goodness of decomposition. MSE is formally described in equation A.3, where N and x^i correspond to a total number of images and a particular image in the dataset.

$$\text{MSE} = \frac{1}{N}\sum_{i}^{N} ||x^i - \Phi_d(\text{CoSA}(\Phi_e(x^i)))||_2^2 \tag{A.3}$$

A.4.2 *Overlapping index (OPI)*

OPI measures the mean overlap of a particular slot with respect to every other slot across the given dataset. For computing the overlap we first rescale the pixel intensities in the estimated mask to lie between (0, 1) by applying *min-max* normalization and later thresholded at 0.5. The normalization and thresholding will result in binary decomposed images, which can be used to compute average overlap. OPI is formally described in equation A.4, where N, S, and $x^i_{s_j}$ correspond to a total number of images, slots, and a particular slot estimated for considered image x^i respectively.

$$\text{OPI} = \frac{1}{N}\sum_{i}^{N}\frac{1}{S}\sum_{j}^{S}\frac{1}{S-j}\sum_{k}^{j}\frac{2(x^i_{s_j} \cap x^i_{s_k})}{x^i_{s_j} \cup x^i_{s_k}} \tag{A.4}$$

A.4.3 *Codebook divergence (CBD)*

We use this metric to measure the distance between the codebook embeddings, higher distance indicates the effective usage of codebook embeddings. To measure this distance we first project all the embedding vectors onto the unit hypersphere and compute the average cosine distance between them. In practice this is done by computing the inner product between the

embedding vectors, which is formally described in equation A.5, where K corresponds a total number of codebook embeddings and $\cos^{-1}$ is applied to get the angular distance between the embeddings.

$$\text{CBD} = \arccos \frac{1}{K} \sum_{i}^{K} \frac{1}{K-i} \sum_{j}^{i} \langle e_i, e_j \rangle \tag{A.5}$$

A.4.4 *Codebook perplexity (CBP)*

Perplexity measure is a property that indicates the sampling efficiency of the codebook. The perplexity is higher if all the codebook embeddings are uniformly sampled. Lower perplexity is an indicator of codebook collapse. Formally the perplexity is estimated as described in equation A.6, where N, K, and qidx_j^i correspond to a total number of images in a dataset, the total number of codebook embeddings and indicator value of j^{th} codebook vector is sampled for image x^i.

$$\text{CBP} = \exp\Big(-\sum_{j}^{K} \frac{1}{N} \sum_{i}^{N} \text{qidx}_j^i \log \frac{1}{N} \sum_{i}^{N} \text{qidx}_j^i \Big) \tag{A.6}$$

A.4.5 *Frechet inception distance (FID)*

FID is computed to measure the quality of generated images with respect to real images, it's usually measured by computing the distance between the latent representations of real and generated images. For measuring this we usually use any pre-trained model Φ_{vgg} for computing latent representation; the FID score is calculated using equation A.7, where $z^r = \frac{1}{N} \sum_i^N \Phi_{vgg}(x_i^r); \Sigma^r = \Phi_{vgg}(x^r)^T \Phi_{vgg}(x^r)$ similarly, z^g and Σ^g are defined of the generated image set.

$$\text{FID}(x^r, x^g) = ||z^r - z^g||_2^2 + \text{Trace}\Big(\Sigma^r + \Sigma^g - 2\sqrt{(\Sigma^r \Sigma^g)}\Big) \tag{A.7}$$

A.4.6 *Slot average Frechet inception distance (SFID)*

As FID measures the quality of the final reconstructed image, to measure the quality of individual slots, we compute FID score with respect original image and individual slot. The intuition behind doing this is if the slot contained object information is present in the original image, the FID for that slot would be less than the slot with random information. We compute the average FID score for all the estimated slots with respect to the original image to obtain SFID, formally described in equation A.8

$$\text{SFID}(x^r, x^g) = \frac{1}{K}\sum_{i}^{K}\text{FID}(x^r, x_i^g) \tag{A.8}$$

A.4.7 *Hungarian Matching Coefficient (HMC)*

In the case of the reasoning objective, we measure the implicit rationale provided by the model by comparing the emerging object properties with ground-truth object properties. To consider the permutation invariance property of slot properties, we perform Hungarian Matching over all the properties with respect to the ground truth properties and compute the MSE cost between paired vectors.

A.5 ALGORITHM & FORWARD PASS

Our proposed object-level representation learning uses an encoder to map an image to its latent representation, followed by computing $\mathbf{q}, \mathbf{k}, \mathbf{v}$ projection vectors. The obtained position-free encoding $\mathbf{z}$ is further passed through image-level abstraction function $\mathcal{A}$ resulting eigenvectors $\boldsymbol{V}$ and corresponding eigenvalues Λ are used to estimate principle components $\mathbf{a}$ as described in section 3.2. As the obtained principle components are with respect to a particular image, to generalize them across the dataset, GSD sampling is used, $\tilde{\mathbf{k}} \sim \mathfrak{S}^1$. Allowing us to sample slots from uniquely mapped conditioning distributions, ensuring slots to be sampled from the same conditioning distribution $\mathfrak{S}_i^2$ for all instances of an object in the case when the given image has more than one instance of the same object. Based

on the heuristic defined in section 3.2, K slots are sampled from their respective conditioning distributions, which are selected with respect to the obtained eigenvalues. Finally, the iterative attention mechanism is applied to obtain slot-level representations for the downstream tasks. Algorithm 1 describes the proposed algorithm.

A.6 QUANTIZATION

In the case of quantization, we first define a codebook $\mathfrak{S} \in \mathbb{R}^{N \times d_z}$ with N codebook embeddings, where individual embedding $\mathfrak{S}_i \in \mathbb{R}^{d_z}$. The objective in the case of quantization is to learn the categorical distribution over codebook features for a given image $\mathbf{x}$. In the case of deterministic sampling, as in the case of Euclidean and Cosine sampling, the categorical distribution is modeled with one-hot probabilities determined by the mapping of each discrete vector to the nearest codebook vector:

$$p(\mathbf{z}_j = \mathfrak{S}_k \mid \mathbf{x}) = \begin{cases} 1 & \text{for } k = \operatorname{argmin}_{\mathfrak{S}_j} \|\mathbf{z} - \mathfrak{S}_j\|^2 \quad \text{or} \quad \langle \mathbf{z}, \mathfrak{S}_j \rangle \\ 0 & \text{otherwise} \end{cases} \tag{A.9}$$

In this case, the probability estimates are usually non-differentiable. Due to this, in practice, back-propagation over this sampling block is achieved with the help of straight-through gradient approximation, enabling end-to-end training as illustrated in Van Den Oord, Vinyals, et al., 2017. Due to the deterministic nature of sampling and the uniform prior assumption, the D_{KL} term in the objective drops to be constant.

While in the case of stochastic sampling, as in Gumbel, the one-hot probabilities are approximated with the softmax distribution resulting in $p(\mathbf{z} \mid \mathbf{x}) = \texttt{softmax}(\exp((g_i + \hat{z}_i)/t))$, where g_i is sampled from a Gumbel distribution, and t is the temperature parameter as detailed in Jang et al., 2016; Maddison et al., 2016.

A.7 CODEBOOK COLLAPSE

Codebook collapse is a common challenge in training vector quantized models, in this case, only one or a selected few embeddings ($>>$ total number of codebook embeddings) of a codebook are being used repeatedly. Various engineering tricks have been proposed to address this issue previously Van Den Oord, Vinyals, et al., 2017; Bardes et al., 2021; Takida et al., 2022; Träuble et al., 2022. In this work, we first detach the inputs to the quantization block so that the object-centric representations are not affected by quantizer gradients. We also use the exponential moving average (EMA) over codebook embeddings which encourages the embedding vectors to converge at the mean of the representations. Along with EMA we also utilize random restart of dead codebook vectors with the running representational statistics.

In terms of slot distributions, we initialize the distribution on a unit hypersphere, fix the mean of the distribution, and learn the slot-specific transformation functions which transform the fixed distribution into the required distribution. This was first proposed in Tomczak and Welling, 2018 to prevent overfitting of learnable priors.

A.8 OBJECT DISCOVERY

Here, we provide more analysis on sensitivity analysis of CoSA on object discovery task. We first test our framework on all three sampling methods using Gumbel, Cosine, and Euclidian codebooks. We use the convolutional encoder and decoder architectures with 5 convolutional and transpose convolutional layers. Additionally, we provide the comparative results on slot properties across different methods.

A.8.1 *Quantitative Analysis*

Table A.3 and A.4 demonstrates the sensitivity of CoSA with respect to selected sampling criteria. Based on the results, it can be observed that CoSA performs similarly or better than its slot attention counterpart, illustrating the effectiveness of grounded representations. We measure the diversity in

the codebook embeddings and their variation in sampling by measuring CBP and CBD properties, and these properties illustrate the variability in sampling and differences in codebook representations. The resulting values for all five datasets are described in table A.5. These results demonstrate that there's no collapse in codebook representations and the sampling (ref. appendix section A.7 for details on codebook collapse). To measure the *goodness* of the generated slots, we measure OPI and SFID which are tabulated in table A.6.

Table A.3: Sensitivity analysis of CoSA on object discovery.

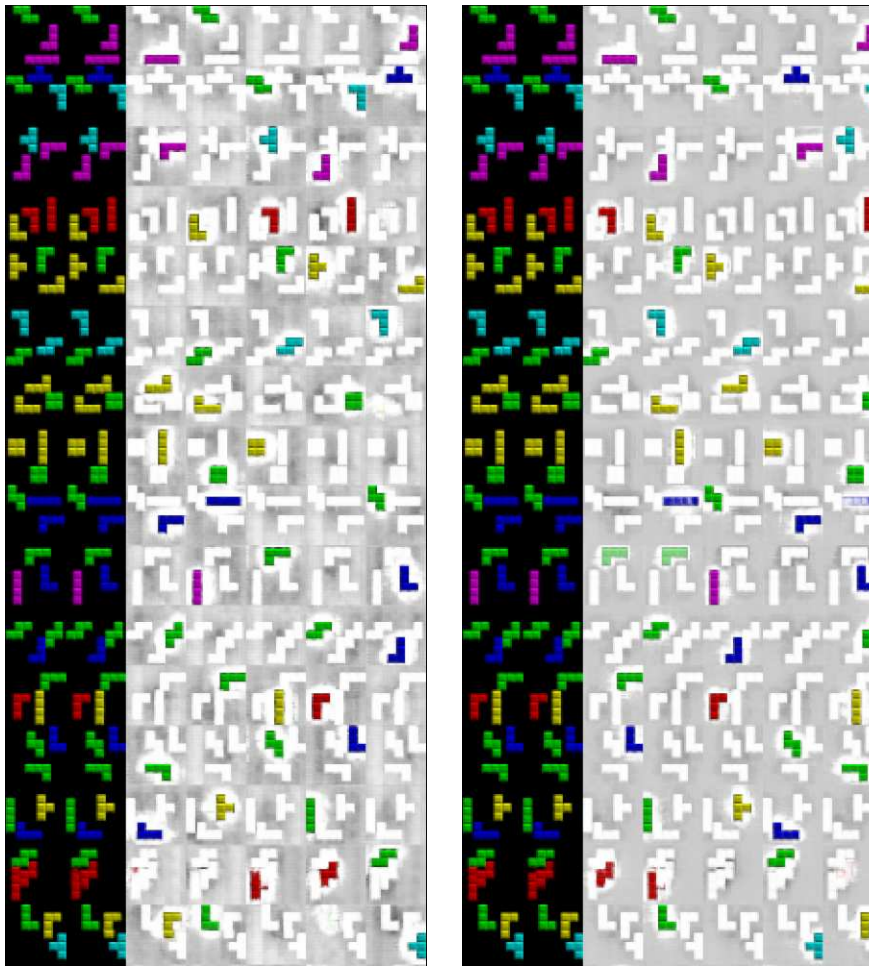
Methods(↓), Metrics(→)	CLEVR		Tetrominoes		Objects-Room		Bitmoji		FFHQ	
	MSE(↓)	FID(↓)	MSE(↓)	FID(↓)	MSE(↓)	FID(↓)	MSE(↓)	FID(↓)	MSE(↓)	FID(↓)
SA	6.37	38.18	1.49	3.81	7.57	38.12	14.62	12.78	55.14	54.95
CoSA-Gumbel	3.82	31.84	**0.21**	**0.26**	5.45	32.41	9.84	9.36	41.77	52.21
CoSA-Euclidian	4.04	33.18	1.13	1.61	6.42	34.05	**8.97**	**8.41**	**31.94**	**35.14**
CoSA-Cosine	**3.14**	**29.12**	0.42	0.41	**4.85**	**28.19**	8.17	9.28	33.37	36.34

A.8.2 *Qualitative Analysis*

Fig. A.2(a), A.6(a), A.8(a), A.10(a), and A.4(a) illustrate vanilla SA results. Fig. A.2(b), A.6(b), A.8(b), A.10(b), and A.4(b) illustrate the results of CoSA-Cosine. While, Fig. A.3(a), A.7(a), A.9(a), A.11(a), and A.5(a) illustrate the results of CoSA-Gumbel and finally, Fig. A.3(b), A.7(b), A.9(b), A.11(b), and A.5(b) illustrates CoSA-Euclidian results.

Table A.4: ARI on CLEVR6, Tetrominoes, ObjectsRoom, and COCO datasets, for SA baseline model and CoSA-cosine variant. In the case of COCO we use DINOSAUR variant of SA Seitzer et al., 2022 (ViT-S16) as a baseline and use DINO feature extractor (ViT-S16) for CoSA.

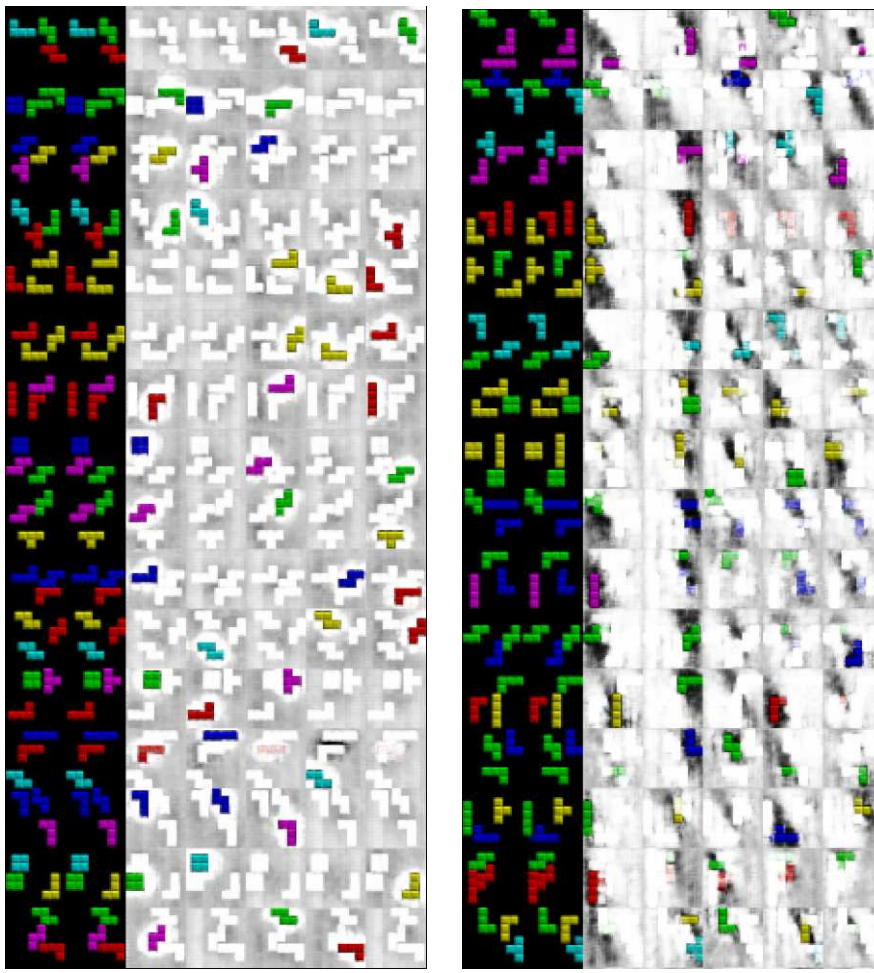
Method	Tetrominoes	CLEVR6	ObjectsRoom	COCO
SA/DINOSAUR	0.99 ± 0.005	0.93 ± 0.002	0.78 ± 0.02	0.28 ± 0.02
CoSA-Euclidian	0.99 ± 0.002	0.94 ± 0.002	0.81 ± 0.01	0.27 ± 0.02
CoSA-Gumbel	0.99 ± 0.001	0.93 ± 0.002	0.80 ± 0.01	0.30 ± 0.02
CoSA-Cosine	0.99 ± 0.001	**0.96** ± 0.002	**0.83** ± 0.002	**0.36** ± 0.01



(a) Baseline

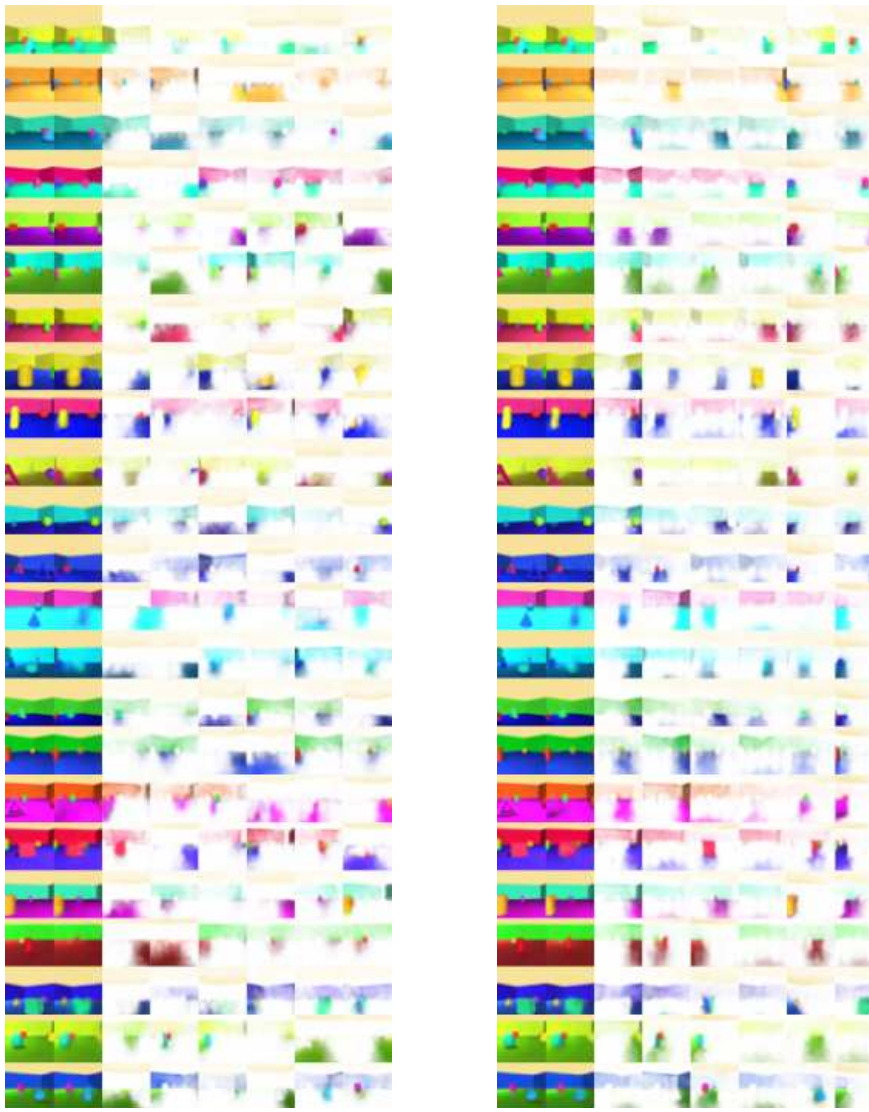
(b) CoSA-Cosine

Figure A.2: Vanilla SA and CoSA-Cosine object discovery results on tetrominoes dataset.



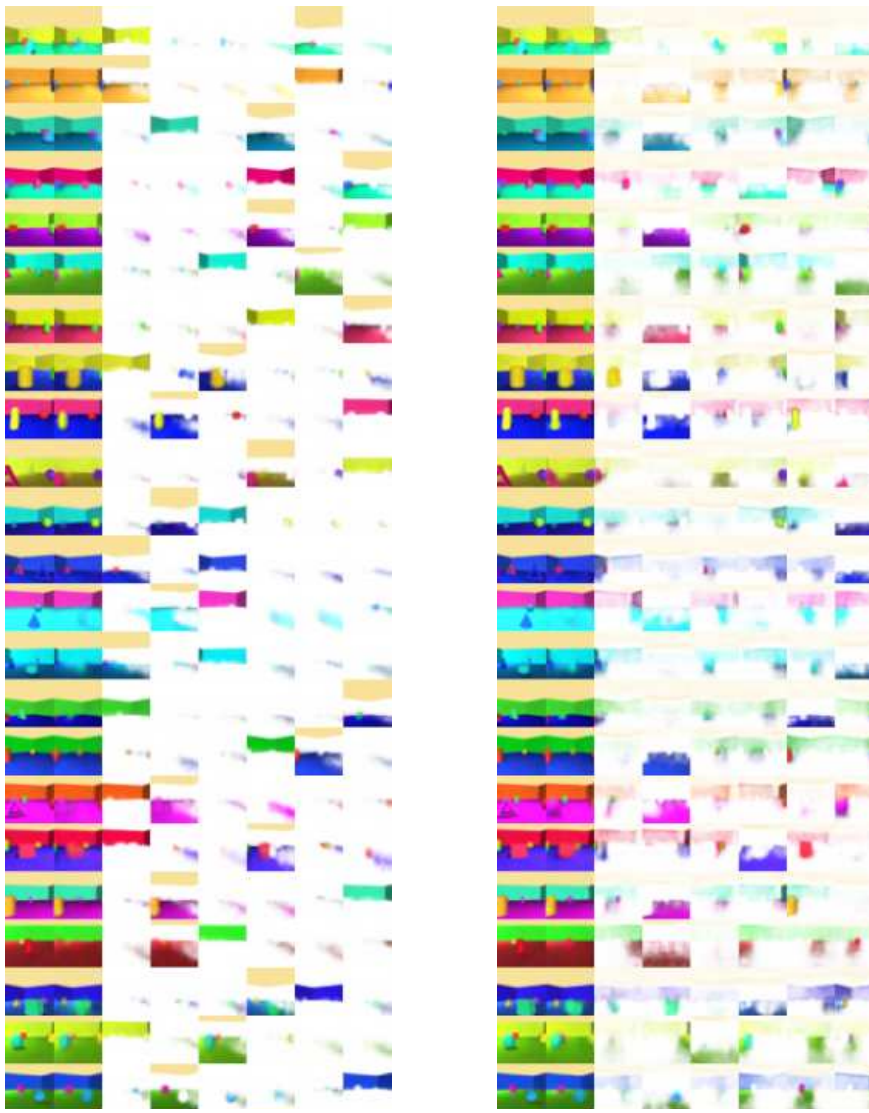
(a) CoSA-Gumbel **(b)** CoSA-Euclidian

Figure A.3: CoSA-Euclidian and CoSA-Gumbel object discovery results on tetrominoes dataset.



(a) Baseline **(b)** CoSA-Cosine

Figure A.4: Vanilla SA and CoSA-Cosine object discovery results on objects-room dataset.



(a) CoSA-Gumbel **(b)** CoSA-Euclidian

Figure A.5: CoSA-Euclidian and CoSA-Gumbel object discovery results on objects-room dataset.



(a) Baseline

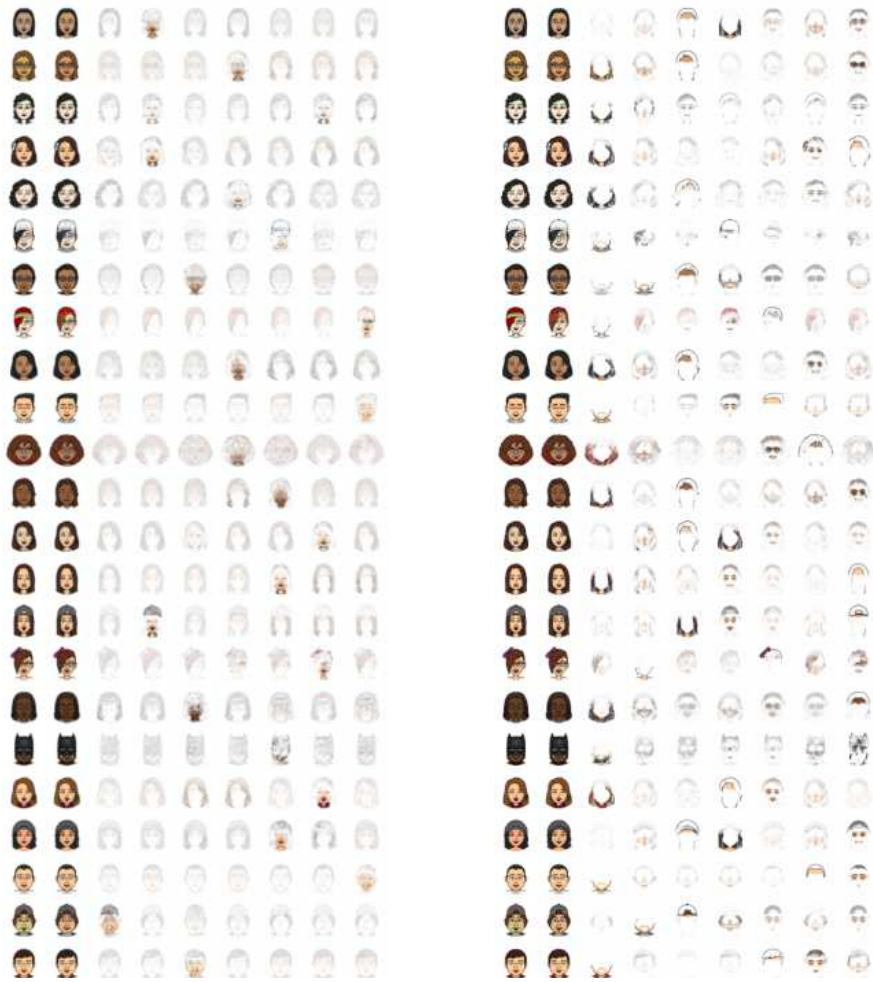
(b) CoSA-Cosine

Figure A.6: Vanilla SA and CoSA-Cosine object discovery results on CLEVR dataset.



(a) CoSA-Gumbel **(b)** CoSA-Euclidian

Figure A.7: CoSA-Euclidian and CoSA-Gumbel object discovery results on CLEVR dataset.



(a) Baseline **(b)** CoSA-Cosine

Figure A.8: Vanilla SA and CoSA-Cosine object discovery results on bitmoji dataset.



(a) CoSA-Gumbel **(b)** CoSA-Euclidian

Figure A.9: CoSA-Euclidian and CoSA-Gumbel object discovery results on bitmoji dataset.



(a) Baseline **(b)** CoSA-Cosine

Figure A.10: Vanilla SA and CoSA-Cosine object discovery results on ffhq dataset.

Table A.5: Codebook properties across different sampling methods across datasets.

Methods(↓), Metrics(→)	CLEVR		Tetrominoes		Objects-Room		Bitmoji		FFHQ	
	CBD(↑)	CBP(↑)	CBD(↑)	CBP(↑)	CBD(↑)	CBP(↑)	CBD(↑)	CBP(↑)	CBD(↑)	CBP(↑)
CoSA-Cosine	0.99	44.39	0.99	26.53	0.99	28.22	0.99	54.87	0.99	36.97
CoSA-Euclidian	0.99	32.40	0.99	28.99	0.99	24.26	0.99	55.50	0.99	24.28
CoSA-Gumbel	1.0	29.62	1.0	22.35	1.0	23.05	1.0	37.61	1.0	24.69

Table A.6: Slot properties.

Methods(↓), Metrics(→)	CLEVR		Tetrominoes		Objects-Room		Bitmoji		FFHQ	
	OPI(↓)	SFID(↓)	OPI(↓)	SFID(↓)	OPI(↓)	SFID(↓)	OPI(↓)	SFID(↓)	OPI(↓)	SFID(↓)
SA	0.43	238.7	0.48	270.3	0.42	162.9	0.41	204.1	0.27	240.9
IMPLICIT	**0.35**	220.1	0.56	249.1	0.37	128.5	0.34	197.4	0.24	244.1
CoSA-Gumbel	**0.35**	192.7	0.43	**231.2**	0.42	119.4	0.30	201.2	0.28	213.1
CoSA-Euclidian	0.40	**173.1**	**0.34**	236.7	**0.34**	137.5	**0.29**	**190.3**	0.33	232.3
CoSA-Cosine	0.45	175.8	0.53	235.2	0.39	**103.2**	0.37	213.3	**0.32**	**211.3**

A.8.3 *DINOSAUR Adaptation*

To extend slot attention to real-world images, we adopt DINOSAUR model Seitzer et al., 2022, which uses the Vision Transformer (ViT-S16) model trained with DINO training strategyCaron et al., 2021. The main objective of DINOSAUR is to reconstruct latent representation rather than reconstructing the original image. To compare our framework under similar specifications, we use DINO as a feature extractor and use CoSA for the extracted latent features to reconstruct these features and use the resulting attention maps as slots. We demonstrate the results of DINOSAUR and variants of DINO-CoSA in Figures A.12, A.13. Table 3.1 illustrates the quantitative results on COCO dataset; note that the results in the original work are based on the ViT-B16 feature extractor, while we use ViT-

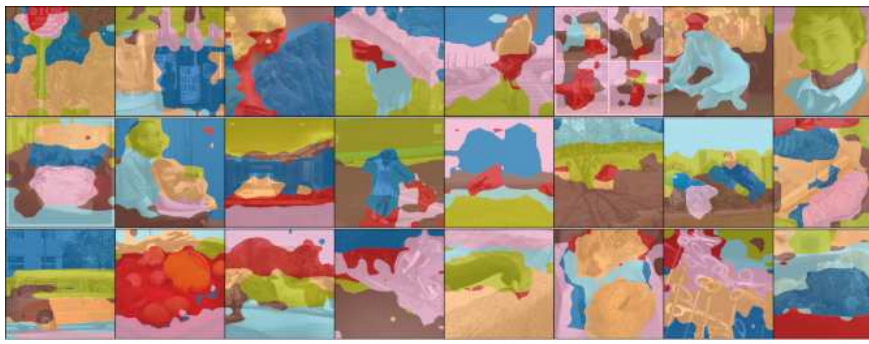


(a) CoSA-Gumbel **(b)** CoSA-Euclidian

Figure A.11: CoSA-Euclidian and CoSA-Gumbel object discovery results on ffhq dataset.



(a) DINOSAUR



(b) DINO-CoSA-Euclidian

Figure A.12: Results on COCO dataset.

S16 due to computational resources. We base our implementation by considering base implementation from `https://github.com/amazon-science/object-centric-learning-framework` and use provided hyperparameters in `'coco_feat_rec_in_small16_auto.yaml'`, with only changes in the number of GPUs and the batch size; we use a single GPU with a batch size of 32.

A.8.4 *Codebook Size Analysis*

Given we make dictionary sufficiency assumption (refer to an assumption A.4), we perform extensive ablations on dictionary size on CLEVR data for all three sampling strategies. Table A.7, A.8, and A.9 correspond to Cosine, Euclidean, and Gumbel sampling strategies. As it can be observed, a very low number of codebook embeddings (M=16) results in high MSE error



(a) DINO-CoSA-Cosine



(b) DINO-CoSA-Gumbel

Figure A.13: Results on COCO dataset.

and lower ARI scores, while increasing the embeddings drastically also does not help. While the number of embeddings around 64, 128 provide similar results without any collapse (indicated by higher perplexity scores). This indicates that the framework is not extremely sensitive to the codebook size, given the codebook does not collapse.

Table A.7: Codebook size ablations with Cosine Sampling

Methods(↓), Metrics(→)	CLEVR			
	MSE(↓)	FG-ARI(↑)	CBD(↑)	CBP(↑)
M=16	8.11	0.73	0.99	14.96
M=64	3.14	0.96	0.99	44.39
M=128	3.46	0.94	0.98	51.65
M=256	6.68	0.85	0.98	82.20

Table A.8: Codebook size ablations with Euclidian Sampling

Methods(↓), Metrics(→)	CLEVR			
	MSE(↓)	FG-ARI(↑)	CBD(↑)	CBP(↑)
M=16	4.81	0.93	0.99	8.33
M=64	4.04	0.94	0.99	32.40
M=128	3.64	0.94	0.95	20.69
M=256	4.34	0.89	0.96	19.79

Table A.9: Codebook size ablations with Gumbel Sampling

Methods(↓), Metrics(→)	CLEVR			
	MSE(↓)	FG-ARI(↑)	CBD(↑)	CBP(↑)
M=16	6.64	0.88	1.0	11.06
M=64	3.82	0.93	1.0	29.62
M=128	3.89	0.93	0.99	27.16
M=256	5.46	0.92	0.99	30.28

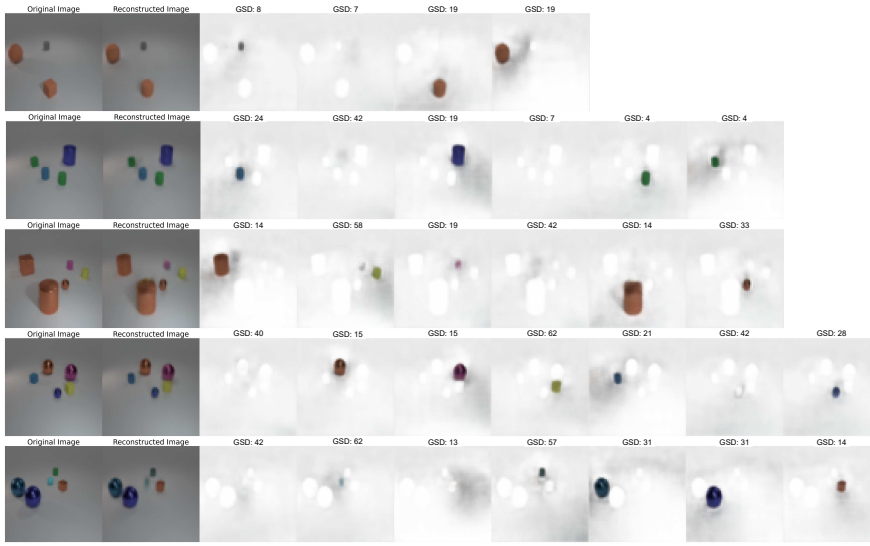


Figure A.14: Demonstration of object binding across instances in a given image. The title of each figure corresponds to the sampled element index of GSD $\mathfrak{S}$.

A.8.5 *Multiple Object Instances*

In this section, we demonstrate the effectiveness of CoSA on binding multiple instances of an object in a given image to a same element in GSD. In Figure A.14, the title of each sub figure correspond to GSD index being selected for that particular object. As observed in first-fifth row GSD elements $\mathfrak{S}_{19}, \mathfrak{S}_{4}, \mathfrak{S}_{14}, \mathfrak{S}_{15}$, and $\mathfrak{S}_{31}$ are being sampled twice, respectively.

Similarly, to previous analysis we use the coco trained model and analyse the binding capabilities across multiple frames in a video sequence. For this we use the COCO trained model and perform zero-shot adaptation in DAVIS video dataset Perazzi et al., 2016. Figure A.15 reflects this analysis, where each row correspond to a particular frame in a vide. We can observed that GSD elements $\mathfrak{S}_{54}$, and $\mathfrak{S}_{24}$ consistently map to camel and the background, respectively.



Figure A.15: Demonstration of object binding across frames in a video sequence. The title of each figure corresponds to the sampled element index of GSD $\mathfrak{S}$.

A.9 ABSTRACTION FUNCTION

We tried multiple variants of spectral decomposition for abstracting object-level features including singular value decomposition, QR decomposition, vanilla quantization, and eigenvalue decomposition. Table A.10 describes the results on the CLEVR dataset for all different types of variants of abstraction functions. We use the principle components based decomposition approach as the default abstraction function because the results do not vary too much among different types of spectral decomposition methods.

Table A.10: Abstraction functions

Methods(↓), Metrics(→)	Vanilla quant.		Principle vec. quant.		QR vec. quant.		Singular vec. quant.	
	MSE(↓)	FID(↓)	MSE(↓)	FID(↓)	MSE(↓)	FID(↓)	MSE(↓)	FID (↓)
CoSA-Gumbel	6.44	43.61	3.82	29.12	4.18	33.74	3.97	26.29
CoSA-Euclidian	8.32	54.11	4.04	33.18	4.76	32.43	4.14	28.57
CoSA-Cosine	7.68	48.72	3.14	31.84	3.64	34.12	3.98	30.22

A.10 OBJECT COMPOSITION

Unlike SLATE Singh et al., 2021, our approach does not require test time prompt generation, as the dictionary behaves as a set of prompts; we do not need to learn prompt dictionary, as we learn object/property level distribution as part of our algorithm. The prompting in our approach is solely based on the learned slot dictionary. Algorithmically, we first sample input features from a dictionary and corresponding slots, followed by positional matting and iterative slot refinement for generating realistic object-level scenes. Algorithm 6 describes the approach with pseudo-code.

Object composition illustrates that the grounded representation (every embeddings in a GSD) are specialised to capture specific individual object in a given environment. We select K slots, sample embeddings from the respective slot distributions and refine them, generating K 4 channelled images, which we further combine by merging them using softmax distribution, providing us with the prompted image.

Algorithm 6 Scene composition by random slot prompting

Prompt Dictionary Items: index $\sim \mathcal{U}\{0, M\}$
Sample Inputs: $\mathbf{z} = \mathfrak{S}^1[\text{index}]$
Compute: $\mathbf{k} = \mathcal{K}_\beta(\mathbf{z})$, and $\mathbf{v} = \mathcal{V}_\phi(\mathbf{z})$
for $i = 0, \ldots, R$ ▷ R Monte Carlo samples
  $\text{slots}_i^0 \sim \mathfrak{S}[\text{index}] \in \mathbb{R}^{K \times d_s}$ ▷ Sample prompted slots from GSD
  for $t = 1, \ldots, T$ ▷ Refine slots over T attention iterations
    $\text{slots}_i^t = f\left(g\left(\mathcal{Q}_\gamma\left(\text{LayerNorm}\left(\text{slots}_i^{t-1}\right), \mathbf{k}\right), \mathbf{v}\right)\right)$ ▷ Update slot representations
    $\text{slots}_i^t \mathrel{+}= \text{MLP}\left(\text{LayerNorm}\left(\mathcal{H}_\theta\left(\text{slots}_i^{t-1}, \text{slots}_i^t\right)\right)\right)$ ▷ GRU update & skip connection
return $\sum_i \text{slots}_i / R$ ▷ MC estimate

A.10.1 *Slot Dictionary Analysis*

Here, we describe the mapped dictionary elements, Fig. 3.3, A.16, A.17, and A.18 illustrate randomly selected dictionary index to slot mapping. To generate these images, we list the sampled codebook index for given 100 images and group them with respect to uniquely sampled codebook indices. It can be observed that the codebook kind groups similar-looking object properties together. In Fig. 3.3, it can be seen that embedding-7 of the codebook corresponds to cheeks, embedding-14 maps to head ware and forehead, embedding-25 maps to eyes, and embedding-55 maps to facial hair. It is important to note that these semantics are completely subjective and assigned by observation. Similar observations can be made on CLEVER and Object-Room models illustrated in Fig. A.16 and Fig. A.17, respectively. The model still struggles to disentangle in the case of natural images like the FFHQ dataset as illustrated in Fig. A.18, nevertheless, it is grouping some similar-looking regions together.

A.10.2 *Scene Generation*

Finally, we also demonstrate the image composition objective by randomly sampling embeddings from the codebook and performing iterative attention

a



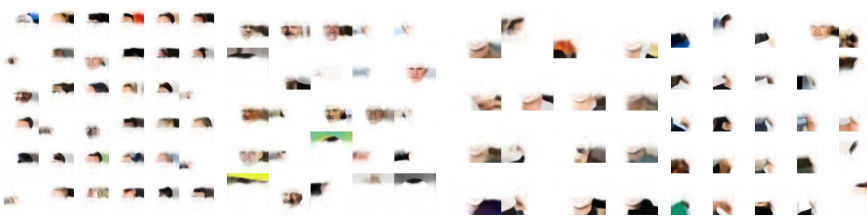
(a) Embedding-6 **(b)** Embedding-1 **(c)** Embedding-22 **(d)** Embedding-7

Figure A.16: Gumbel dictionary embedding to observation mapping for CLEVR dataset.



(a) Embedding-12 **(b)** Embedding-17 **(c)** Embedding-24 **(d)** Embedding-9

Figure A.17: Cosine dictionary embedding to observation mapping for Objects-room dataset.



(a) Embedding-1 **(b)** Embedding-4 **(c)** Embedding-5 **(d)** Embedding-6

Figure A.18: Gumbel dictionary embedding to observation mapping for FFHQ dataset.

Table A.11: Compositional FID and SFID scores.

Methods(↓), Metrics(→)	CLEVR		Tetrominoes		Objects-Room		Bitmoji		FFHQ	
	FID(↓)	SFID(↓)	FID(↓)	SFID(↓)	FID(↓)	SFID(↓)	FID(↓)	SFID(↓)	FID(↓)	SFID(↓)
CoSA-Gumbel	116.1	257.7	32.1	256.1	194.1	184.2	127.1	262.6	225.7	237.1
CoSA-Euclidian	122.2	236.2	36.3	278.4	194.1	184.2	114.3	215.8	209.3	241.3
CoSA-Cosine	117.8	240.1	28.1	256.8	184.2	186.0	114.3	273.2	185.8	235.2



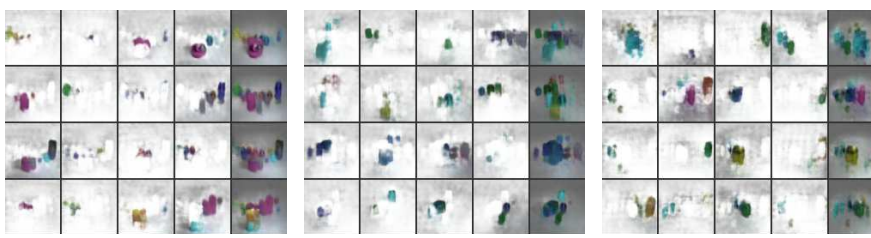
(a) CoSA-Cosine **(b)** CoSA-Euclidian **(c)** CoSA-Gumbel

Figure A.19: Image composition with random slot prompt on bitmoji dataset.

to generate results. Fig. A.19, A.20, and A.21 illustrate both randomly sampled slots and the resulting image composition.

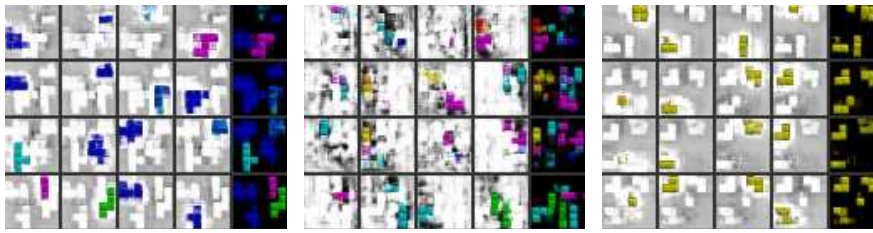
A.11 Discriminative/Reasoning Tasks

For validating discriminative tasks, we consider CLEVR-Hans and FloatingMNIST tasks, where the main objective is to segregate the image into different classes based on the contents of the image, while proving a rational for the segregation without explicitly training for the rational. More details about the datasets are described in A.3. As the classification is based on the object properties in the images, we consider the properties as *emerg-*



(a) CoSA-Cosine **(b)** CoSA-Euclidian **(c)** CoSA-Gumbel

Figure A.20: Image composition with random slot prompt on CLEVR dataset.



(a) CoSA-Cosine **(b)** CoSA-Euclidian **(c)** CoSA-Gumbel

Figure A.21: Image composition with random slot prompt on Tetrominoes dataset.

ing entities and model it as a reasoning task by considering the emerging properties as a rationale for prediction. The correctness of the generated rationale is measured using Hungarian Matching Coefficient (HMC).

To demonstrate the generalizability property of model due to grounded representations, we perform cross domain adaptability test. For adaptability test we train the model on one dataset and fine-tune the classification head on other datasets by controlling the data samples. The main hypothesis here is that the grounded representations are meaningful environmental properties, which goes beyond the downstream tasks. Qualitatively, table A.13 describes the performance of the model trained and tested on CLEVR-Hans3 and CLEVR-Hans7 datasets in columns 1 and 3, while the other two columns indicate the task adaptability results with few shot (k=100) learning performances with the different base model. Similarly, table A.12, A.14 and table A.15 describe the results on FloatingMNIST2 and FloatingMNIST3 datasets respectively with different variants of tasks (Add, Sub, and Mixed). Finally, table A.16 demonstrates the performance on Mixed tasks on FloatingMNIST2 and FloatingMNIST3.

A.12 TRAINING AND COMPUTATIONAL DETAILS

We train all the models with adam optimizer with a learning rate of 0.0004, batch size of 16, with early stopping with the patience of 5, and for a maximum of min(40epochs, 40000 steps). We use linear learning rate warmup for 10000 steps with a decay rate of 0.5 followed by reduce on the plateau learning rate scheduler. The experimental config files and training/inference scripts with be made available for reproducibility and further research.

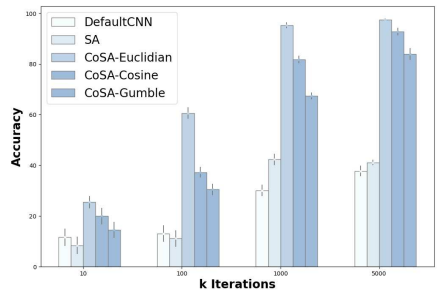


Figure A.22: Few shot task adaptability accuracy for all the methods.

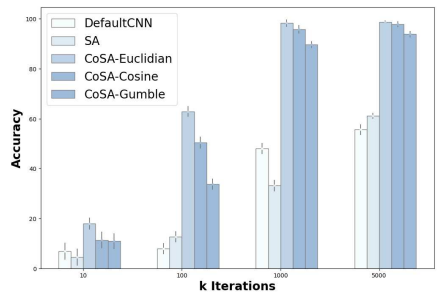


Figure A.23: Few shot adaptability on FloatingMNIST2-diff to FloatingMNIST2-add dataset.

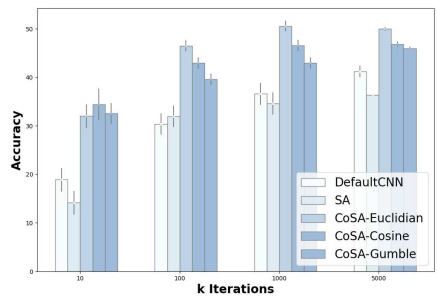


Figure A.24: Few shot adaptability on CLEVR-hans3 to CLEVR-hans7 dataset.

Table A.12: Accuracy and Hungarian matching coefficient (HMC) for reasoning objective on addition and subtraction variant of floatingMNIST2 (FMNIST2) dataset. Here, the first and third pair of columns correspond to the models trained and tested on the FMNIST2-Add and FMNIST2-Sub datasets, respectively, while the second and fourth pair correspond to few-shot(k=100) adaptability results across datasets.

Methods(↓), Metrics(→)	FMNIST2-Add$_{source}$		FMNIST2-Sub$_{target}$		FMNIST2-Sub$_{source}$		FMNIST2-Add$_{target}$	
	Acc(↑)	HMC(↓)	Acc(↑)	F1(↑)	Acc(↑)	HMC(↓)	Acc(↑)	F1(↑)
CNN	97.62	-	10.35	10.05	98.16	-	12.35	9.50
SA	97.33	0.14	11.06	09.40	97.41	0.13	08.28	7.83
CoSA-Gumbel	97.52	0.11	30.75	28.23	98.44	0.12	28.45	20.53
CoSA-Euclidian	**98.12**	**0.10**	**60.24**	**50.16**	**98.64**	**0.12**	**63.29**	**58.29**
CoSA-Cosine	**98.33**	**0.10**	37.47	36.00	**98.81**	**0.12**	50.43	44.47

Table A.13: Accuracy and Hungarian matching coefficient (HMC) for reasoning objective on CLEVR-Hans dataset. Here, the first and third pair of columns correspond to the models trained and tested on CLEVR-Hans3 and CLEVR-Hans7 datasets respectively, while the second and fourth pair corresponds to few shot(k=100) adaptability results across models.

Methods(↓), Metrics(→)	CLEVR-Hans3$_{source}$		CLEVR-Hans7$_{target}$		CLEVR-Hans7$_{source}$		CLEVR-Hans3$_{target}$	
	Acc(↑)	HMC(↓)	Acc(↑)	F1(↑)	Acc(↑)	HMC(↓)	Acc(↑)	F1(↑)
CNN	67.89	-	28.14	16.66	80.06	-	58.49	55.11
SA	68.53	0.38	31.69	26.21	81.12	0.41	70.08	68.28
CoSA-Gumbel	**71.88**	**0.26**	40.42	38.18	**82.72**	0.32	69.98	69.81
CoSA-Euclidian	70.64	0.28	**46.45**	**46.88**	**81.17**	**0.30**	71.35	68.33
CoSA-Cosine	70.28	0.28	43.23	45.05	**82.28**	**0.31**	**72.82**	**70.41**

We run all our experiments on a cluster with Nvidia Telsa T4 16GB GPU card with Intel(R) Xeon(R) Gold 6230 CPU. The processing speed of our framework in comparison to the baselines is tabulated below. Table A.17 and A.18 describe the object discovery and reasoning speed in terms of iteration/sec. It is to be noted that speed might differ slightly with respect to the considered system and the background processes.

All experimental scripts and scripts to generate datasets is available on GitHub at `https://github.com/koriavinash1/CoSA`.

A.13 LIMITATIONS AND FUTURE WORK

Our proposed method also possesses a few limitations; here we detail them and also propose a few possible ways to address them:

- Limited variation in slot-prompting: As the current prompting is solely based on dictionary size, the diversity in scene generation is

Table A.14: Accuracy and Hungarian matching coefficient (HMC) for reasoning objective on addition and subtraction variant of floatingMNIST2 (FMNIST2-mixed) dataset. Here, Base models corresponds to the models trained on source datasets and datasets described in the following row correspond to the target datasets, used in generalisability.

Base models (→)	FMNIST2-Mixed		FMNIST2-Mixed		FMNIST2-Mixed		FMNIST2-Add		FMNIST2-Sub	
Methods(↓), Metrics(→)	FMNIST2-Mixed		FMNIST2-Add		FMNIST2-Sub		FMNIST2-Mixed		FMNIST2-Mixed	
	Acc(↑)	HMC(↓)	Acc(↑)	F1(↑)	Acc(↑)	F1(↑)	Acc(↑)	F1(↑)	Acc(↑)	F1(↑)
CNN	**98.93**	-	23.96	18.13	13.53	09.56	14.01	07.10	05.62	04.22
SA	**98.12**	0.11	15.32	11.25	20.02	15.49	13.13	09.55	14.13	09.57
CoSA-Gumbel	**98.28**	0.11	37.86	34.45	41.04	34.75	29.30	24.04	32.57	24.72
CoSA-Euclidian	**98.12**	0.10	53.22	45.98	63.05	50.15	**50.59**	**44.48**	**60.39**	**53.15**
CoSA-Cosine	**98.40**	**0.08**	**56.21**	**49.30**	**64.60**	**58.18**	40.54	37.56	50.63	43.44

Table A.15: Accuracy and Hungarian matching coefficient (HMC) for reasoning objective on addition and subtraction variant of floatingMNIST3 (FMNIST3-mixed) dataset. Here, Base models corresponds to the models trained on source datasets and datasets described in the following row correspond to the target datasets, used in generalisability.

Base models (→)	FMNIST3-Mixed		FMNIST3-Mixed		FMNIST3-Mixed		FMNIST3-Add		FMNIST3-Sub	
Methods(↓), Metrics(→)	FMNIST3-Mixed		FMNIST3-Add		FMNIST3-Sub		FMNIST3-Mixed		FMNIST3-Mixed	
	Acc(↑)	HMC(↓)	Acc(↑)	F1(↑)	Acc(↑)	F1(↑)	Acc(↑)	F1(↑)	Acc(↑)	F1(↑)
CNN	96.23	-	05.29	04.61	07.06	08.13	10.66	04.11	10.19	04.99
SA	96.29	0.18	08.83	04.77	06.68	03.40	07.16	03.87	13.81	06.28
CoSA-Gumbel	**97.37**	**0.14**	14.13	08.53	21.77	12.31	11.95	09.31	18.95	12.28
CoSA-Euclidian	97.07	0.15	16.32	11.26	35.46	22.09	**23.37**	**18.97**	**20.50**	**11.76**
CoSA-Cosine	96.85	0.15	**17.87**	**13.04**	**35.35**	**30.38**	19.01	11.95	**20.94**	**10.47**

limited. To address this, we can explore the possibility of stochastic VQVAE Takida et al., 2022. Additionally, contrastive learning could also prove useful to learn shape, colour, location, and texture codebooks separately, increasing the control and variability of compositional scene generation.

- Assumption of representation overcrowding: the assumption that representation overcrowding can be controlled with the architectural design, but exploring the direction to enforce this behaviour with additional regularization would be interesting.

- Heuristic for slot-distribution selection in the case of non-unique object composition: The current heuristic to handle repeated objects in a scene is dependent on the scale of principle components (eigenvalues of the principle direction vectors); it would be interesting to explore the directions to have it implicitly within the sampling procedure.

Table A.16: Accuracy and Hungarian matching coefficient (HMC) for reasoning objective on addition and subtraction variant of floatingMNIST3 (FMNIST3) dataset. Here, the first and third pair of columns correspond to the models trained and tested on FMNIST3-Add and FMNIST3-Sub variant of datasets respectively, while the second and fourth pair corresponds to k-shot adaptability results across models.

Base models (→)	FMNIST3-Add		FMNIST3-Add		FMNIST3-Sub		FMNIST3-Sub	
Methods(↓), Metrics(→)	FMNIST3-Add		FMNIST3-Sub		FMNIST3-Sub		FMNIST3-Add	
	Acc(↑)	HMC(↓)	Acc(↑)	F1(↑)	Acc(↑)	HMC(↓)	Acc(↑)	F1(↑)
CNN	96.42	-	15.92	08.10	97.62	-	06.80	04.46
SA	96.66	0.16	12.38	07.25	97.33	0.14	05.45	02.67
CoSA-Gumbel	**98.76**	0.14	18.92	10.11	97.52	0.11	09.59	08.01
CoSA-Euclidian	**98.11**	0.14	19.82	11.72	**98.12**	**0.10**	**17.91**	**11.03**
CoSA-Cosine	**98.33**	**0.10**	**20.22**	**13.63**	**98.33**	**0.10**	14.01	07.82

Table A.17: Iteration per second details for all datasets on object discovery task.

Methods(↓), Metrics(→)	CLEVR	Tetrominoes	Objects-Room	Bitmoji	FFHQ
	it/s	it/s	it/s	it/s	it/s
SA	10	16	16	10	5
IMPLICIT	14	22	22	14	6
CoSA-Gumbel	7	18	18	7	4
CoSA-Euclidian	9	20	20	9	5
CoSA-Cosine	9	20	20	9	5

Table A.18: Iteration per second details for all datasets on reasoning task

Methods(↓), Metrics(→)	CLEVR-Hans	FlaotingMNIST
	it/s	it/s
SA	16	16
CNN	20	20
CoSA-Gumbel	12	12
CoSA-Euclidian	15	15
CoSA-Cosine	15	15

B

PSA-APPENDIX

B.1 LIST OF ASSUMPTIONS

Assumption B.1 ($\mathcal{B}$- Disentanglement, Lachapelle et al., 2024b)**.** *Let* $\mathbf{s} = \{\mathbf{s}_B, \forall\, B \in \mathcal{B}\}$ *be a set of features wrt partition set* $\mathcal{B}$*. The learned mixing function* f_d *is said to be* $\mathcal{B}$ *disentangled wrt true decoder* $\tilde{f}_d$ *if there exists a permutation respecting diffeomorphism* $v_B = \tilde{f}_d^{-1} \circ f_d \,\forall\, B \in \mathcal{B}$ *which for a given feature* $\mathbf{s}$ *can be expressed as* $v_B(\mathbf{s}) = v_B(\mathbf{s}_B)$*.*

Assumption B.2 (Additive Mixing Function)**.** *A mixing function* $f_d : \mathcal{S} \to \mathcal{X}$*, is said to be additive if there exist a partition set* $\mathcal{B}$ *and functions* $f_d^B : s \to \mathbb{R}^{|\mathcal{X}|}$ *such that:* $f_d(\mathbf{s}) = \sum_{B \in \mathcal{B}} f_d^B(\mathbf{s}_B)$*.*

Assumption B.3 (Irreducibility)**.** *Given an an object* $\mathbf{x}_k \in \mathbf{x}$*, a model is considered as irreducible if any subset of an object,* $\mathbf{y} \subseteq \mathbf{x}_k$ *is not functionally independent of the complement of the subset contained within the object,* $\mathbf{y}^c \cap \mathbf{x}$ *as expressed by the Jacobin rank inequality in equation 5 in Brady et al., 2023.*

Assumption B.4 (Compositionality)**.** *Compositionality as defined by is a structure imposed on the slot decoder* f_d *which implies that each image pixel is a function of at most one slot representation, thereby enforcing a local sparsity structure on the Jacobian matrix of* f_d *Brady et al., 2023.*

Remark B.1. *The compositionality is guaranteed by explicitly minimising the compositionality contrast but was empirically observed to be implicitly satisfied in the case of additive decoders Brady et al., 2023. Later, Lachapelle et al., 2024b showed this as the property of additive decoder models. However, the additive decoders studied by Lachapelle et al., 2024b are not expressive enough to represent the "masked decoders" typically used in object-centric representation learning, which stems from the normalization of the alpha masks. This means some care must be taken in extrapolating the results in Lachapelle et al., 2024b to the models we use in practice. Additionally, additive decoders scale linearly in the number of slots K, so some less significant scalability issues remain relative to state-of-the-art non-additive decoders (e.g. using Transformers).*

Assumption B.5 (**u** task)**.** *Conditioning latent variables on an observed variable to yield identifiable models. The main assumption is that conditioning on a (potentially observed) variable* **u** *renders the latent variables independent of each other Khemakhem et al., 2020a.*

Assumption B.6 (Object Sufficiency)**.** *A model is said to be object sufficient iff there are no additional objects in the original data distributions other than the ones expressed in training data.*

Assumption B.7 (Decoder Injectivity)**.** *The function $f_d : \mathcal{S} \to \mathcal{X}$ mapping from slot space to image space is a non-linear piecewise affine injective function. That is, it specifies a unique one-to-one mapping between slots and images.*

Remark B.2. *In practice, we use a monotonically increasing decoder with leakyReLU activation which should encourage injectivity behaviour Khemakhem et al., 2020a; Khemakhem et al., 2020b.*

Assumption (*Weak* Injectivity Kivva et al., 2022) Let $f : \mathcal{Z} \to \mathcal{X}$ be a mapping between latent space and image space, where $\dim(\mathcal{Z}) \leq \dim(\mathcal{X})$. The mapping f_d is weakly injective if there exists $\mathbf{x}_0 \in \mathcal{X}$ and $\delta > 0$ such that $|f^{-1}(\{\mathbf{x}\})| = 1, \forall\ \mathbf{x} \in B(\mathbf{x}_0, \delta) \cap f(\mathcal{Z})$, and $\{\mathbf{x} \in \mathcal{X} :|\ f^{-1}(\{\mathbf{x}\})\ |= \infty\} \subseteq f(\mathcal{Z})$ has measure zero w.r.t. to the Lebesgue measure on $f(\mathcal{Z})$.

Remark B.3. *In words, Assumption 5.1 says that a mapping f_d is weakly injective if: (i) in a small neighbourhood around a specific point $\mathbf{x}_0 \in \mathcal{X}$ the mapping is injective – meaning each point in this neighbourhood maps to exactly one point in the latent space $\mathcal{Z}$; and (ii) while f_d may not be globally injective, the set of points in $\mathcal{X}$ that map back to an infinite number of points in $\mathcal{Z}$ (non-injective points) is almost non-existent in terms of the Lebesgue measure on the image of $\mathcal{Z}$ under f_d.*

B.2 AGGREGATE POSTERIOR PROOFS

Lemma 4.2(Aggregate Posterior Mixture) Given that probabilistic slot attention induces a local (per-datapoint $\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^M$) GMM with K compo-

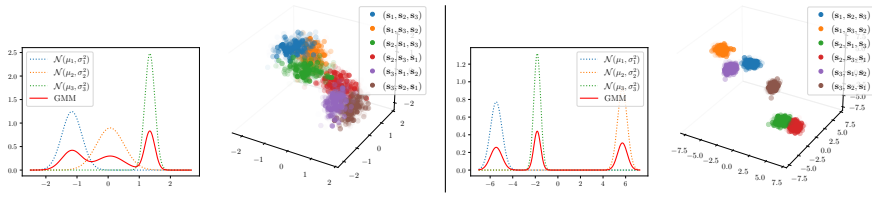


Figure B.1: Concatenated Slot Gaussian Mixture. Examples of the higher dimensional Gaussian mixture induced by the permutation equivariance of slot attention and the $K!$ ways of concatenating sampled slots. Here we start with a random 1D GMM with $K = 3$ modes, each representing a different slot distribution, which then induces a respective 3D GMM with $K! = 6$ modes.

nents, the aggregate posterior $q(\mathbf{z})$ obtained by marginalizing out $\mathbf{x}$ is a non-degenerate global Gaussian mixture with MK components given by:

$$q(\mathbf{z}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} \mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right). \tag{B.1}$$

Proof. We begin by noting that the aggregate posterior $q(\mathbf{z})$ is the optimal prior $p(\mathbf{z})$ so long as our posterior approximation $q(\mathbf{z} \mid \mathbf{x})$ is close enough to the true posterior $p(\mathbf{z} \mid \mathbf{x})$, since for a dataset $\{\mathbf{x}_i\}_{i=1}^{M}$ we have that:

$$\begin{aligned}
p(\mathbf{z}) &= \int p(\mathbf{z} \mid \mathbf{x}) p(\mathbf{x}) d\mathbf{x} && \text{(B.2)} \\
&= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[p(\mathbf{z} \mid \mathbf{x}) \right] && \text{(B.3)} \\
&\approx \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{z} \mid \mathbf{x}_i) && \text{(B.4)} \\
&\approx \frac{1}{M} \sum_{i=1}^{M} q(\mathbf{z} \mid \mathbf{x}_i) && \text{(B.5)} \\
&=: q(\mathbf{z}), && \text{(B.6)}
\end{aligned}$$

where we approximated $p(\mathbf{x})$ using the empirical distribution, then substituted in the approximate posterior and marginalized out $\mathbf{x}$. This observation was first made by Hoffman and Johnson, 2016 and we use it to motivate our setup.

In our case, probabilistic slot attention (Algorithm 5) fits a (local) GMM to each latent variable sampled from the approximate posterior: $\mathbf{z} \sim q(\mathbf{z} \mid \mathbf{x}_i)$,

for $i = 1, \ldots, M$. Let $f(\mathbf{z})$ denote the (local) GMM density, its expectation is given by:

$$\mathbb{E}_{p(\mathbf{x}),q(\mathbf{z}|\mathbf{x})}\left[f(\mathbf{z})\right] = \iint p(\mathbf{x})q(\mathbf{z} \mid \mathbf{x})f(\mathbf{z})d\mathbf{x}d\mathbf{z} \tag{B.7}$$

$$\approx \iint \frac{1}{M}\sum_{i=1}^{M}\delta(\mathbf{x} - \mathbf{x}_i)q(\mathbf{z} \mid \mathbf{x})f(\mathbf{z})d\mathbf{x}d\mathbf{z} \tag{B.8}$$

$$= \int \frac{1}{M}\sum_{i=1}^{M} q(\mathbf{z} \mid \mathbf{x}_i)f(\mathbf{z})d\mathbf{z} \tag{B.9}$$

$$= \int \frac{1}{M}\sum_{i=1}^{M}\mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}_i), \sigma^2(\mathbf{x}_i)\right) \cdot \sum_{k=1}^{K}\pi_{ik}\mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \sigma_{ik}^2\right) d\mathbf{z} \tag{B.10}$$

$$\approx \int \frac{1}{M}\sum_{i=1}^{M}\delta(\mathbf{z} - \boldsymbol{\mu}(\mathbf{x}_i)) \cdot \sum_{k=1}^{K}\pi_{ik}\mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \sigma_{ik}^2\right) d\mathbf{z} \tag{B.11}$$

$$= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K}\pi_{ik}\mathcal{N}\left(\boldsymbol{\mu}(\mathbf{x}_i); \boldsymbol{\mu}_{ik}, \sigma_{ik}^2\right) \tag{B.12}$$

$$=: q(\mathbf{z}), \tag{B.13}$$

where we again used the empirical distribution approximation of $p(\mathbf{x})$, and the following basic identity of the Dirac delta to simplify: $\int \delta(\mathbf{x} - \mathbf{x}')f(\mathbf{x})d\mathbf{x} = f(\mathbf{x}')$.

For the general case, however, we must instead compute the product of $q(\mathbf{z} \mid \mathbf{x})$ and $f(\mathbf{z})$ rather than use a Dirac delta approximation as in Equation C.46. To that end we may proceed as follows w.r.t. to each datapoint $\mathbf{x}_i$:

$$q(\mathbf{z} \mid \mathbf{x}_i) \cdot f(\mathbf{z}) = \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}_i), \sigma^2(\mathbf{x}_i)\right) \cdot \sum_{k=1}^{K}\pi_{ik}\mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \sigma_{ik}^2\right) \tag{B.14}$$

$$= \sum_{k=1}^{K}\pi_{ik}\left[\mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \sigma_{ik}^2\right) \cdot \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}_i), \sigma^2(\mathbf{x}_i)\right)\right] \tag{B.15}$$

$$= \sum_{k=1}^{K}\widehat{\pi}_{ik}\mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\sigma}_{ik}^2\right), \tag{B.16}$$

where the posterior parameters of the resulting mixture are given in closed-form by:

$$\widehat{\sigma}_{ik}^2 = \left(\frac{1}{\sigma_{ik}^2} + \frac{1}{\sigma^2(\mathbf{x}_i)} \right)^{-1}, \quad \widehat{\boldsymbol{\mu}}_{ik} = \widehat{\sigma}_{ik}^2 \left(\frac{\boldsymbol{\mu}(\mathbf{x}_i)}{\sigma^2(\mathbf{x}_i)} + \frac{\boldsymbol{\mu}_{ik}}{\sigma_{ik}^2} \right), \tag{B.17}$$

which are the standard distributional parameters obtained from a product of two Gaussians.

For the updated mixture coefficients $\widehat{\boldsymbol{\pi}}_{ik}$, we propose a principled way to include a posterior-weighted contribution of each mode to the new mixture coefficients. First, note that $(\boldsymbol{\pi}_{i1}, \boldsymbol{\pi}_{i2}, \ldots, \boldsymbol{\pi}_{iK})$ are parameters of a multinomial distribution as $\sum_{k=1}^{K} \boldsymbol{\pi}_{ik} = 1$, for each datapoint $\mathbf{x}_i$. Since the Dirichlet distribution is the conjugate prior of the multinomial distribution, we can place a Dirichlet prior over the mixing coefficients for each datapoint, then update it to a posterior using the data. Concretely, we place a symmetric Dirichlet prior over the mixing coefficients as follows:

$$(\boldsymbol{\pi}_{i1}, \boldsymbol{\pi}_{i2}, \ldots, \boldsymbol{\pi}_{iK}) \sim \text{Dirichlet}\left(\boldsymbol{\alpha}_{i1}, \boldsymbol{\alpha}_{i2}, \ldots, \boldsymbol{\alpha}_{iK}\right), \quad \text{for} \quad i = 1, 2, \ldots, M, \tag{B.18}$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^K$ are the concentration parameters of the i^{th} Dirichlet distribution, and $\forall i, k : \boldsymbol{\alpha}_{ik} = 1$, indicating uniformity over the open standard $(K-1)$-simplex. To compute the posterior Dirichlet distribution we calculate 'pseudo-counts' by integrating the product of the posterior density $q(\mathbf{z} \mid \mathbf{x}_i)$ with each one of the K modes of the Gaussian mixture, thereby measuring a posterior-weighted contribution of each mode k to the new aggregate mixture:

$$c_{ik} = \int \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \sigma_{ik}^2\right) \cdot \mathcal{N}\big(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}_i), \sigma^2(\mathbf{x}_i)\big) d\mathbf{z}, \quad \text{for} \quad i = 1, 2, \ldots, M, \tag{B.19}$$

which we can then use as pseudo-counts to compute the Dirichlet posterior:

$$(\widehat{\boldsymbol{\pi}}_{i1}, \widehat{\boldsymbol{\pi}}_{i2}, \ldots, \widehat{\boldsymbol{\pi}}_{iK}) \mid (c_{i1}, c_{i2}, \ldots, c_{iK}) \sim \tag{B.20}$$

$$\text{Dirichlet}\left(\boldsymbol{\alpha}_{i1} + c_{i1}, \boldsymbol{\alpha}_{i2} + c_{i2}, \ldots, \boldsymbol{\alpha}_{iK} + c_{iK}\right), \tag{B.21}$$

for $i = 1, 2, \ldots, M$. Each posterior probability is then readily given by the mean estimate

$$\widehat{\boldsymbol{\pi}}_{ik} = \frac{\boldsymbol{\alpha}_{ik} + c_{ik}}{\sum_{j=1}^{K}(\boldsymbol{\alpha}_{ij} + c_{ij})} \implies \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} = 1. \tag{B.22}$$

Putting everything together, the aggregated posterior is therefore given by:

$$q(\mathbf{z}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} \mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right), \quad \text{where} \quad \mathbf{z} \sim q(\mathbf{z} \mid \mathbf{x}), \tag{B.23}$$

which concludes the proof. □

Corollary B.1 (Multiview Aggregate Posterior)**.** *The aggregate posterior $q(\mathbf{z})$ is a non-degenerate Gaussian mixture, in the sense that it is a well-defined probability distribution, as the updated mixture coefficients sum to 1 over the number of components $M \times K$.*

Proof. Recall from Lemma 4.2 that the aggregate posterior $q(\mathbf{z})$ – obtained by marginizaling out $\mathbf{x}$ from a probabilistic slot attention model – is a mixture distribution of $M \times K$ components with the following parameters:

$$\left\{\widehat{\boldsymbol{\pi}}_{ik}, \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right\}, \quad \text{for} \quad i = 1, 2, \ldots, M, \quad \text{and} \quad k = 1, 2, \ldots, K. \tag{B.24}$$

To verify that $q(\mathbf{z})$ is a non-degenerate mixture, we observe the following implication:

$$\sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} = 1, \qquad \text{for} \quad i = 1, 2, \ldots, M, \tag{B.25}$$

due to the Dirichlet posterior update in Equation B.22, and therefore

$$\implies \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} = \frac{1}{M} \sum_{i=1}^{M} 1 = \frac{1}{M} \cdot M = 1 \tag{B.26}$$

$$\implies \sum_{i=1}^{M} \sum_{k=1}^{K} \frac{\widehat{\boldsymbol{\pi}}_{ik}}{M} = 1, \tag{B.27}$$

which says that the scaled sum of the mixing proportions of all K components in all M GMMs must equal 1, proving that the associated aggregate posterior mixture $q(\mathbf{z})$ is a well-defined probability distribution. □

Theorem C.1 (Mixture Distribution of Concatenated Slots)**.** Let f_s denote a permutation equivariant probabilistic slot attention function such that $f_s(\mathbf{z}, P\mathbf{s}^t) = Pf_s(\mathbf{z}, \mathbf{s}^t)$, where $P \in \{0,1\}^{K\times K}$ is an arbitrary permutation matrix. Let $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K) \in \mathbb{R}^{Kd}$ be a random variable defined as the concatenation of K individual sampled slots, where each slot is Gaussian distributed within a K-component mixture: $\mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall\, k \in \{1, \ldots K\}$. Then, it holds that $\mathbf{s}$ is also Gaussian mixture distributed comprising $K!$ mixture components:

$$p(\mathbf{s}) = \sum_{p=1}^{K!} \boldsymbol{\pi}_p \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p), \qquad \text{where} \tag{B.28}$$

$$\boldsymbol{\pi} \in \Delta^{K!-1}, \quad \boldsymbol{\mu}_p \in \mathbb{R}^{Kd}, \quad \boldsymbol{\Sigma}_p \in \mathbb{R}^{Kd\times Kd}. \tag{B.29}$$

Proof. Each slot $\mathbf{s}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is sampled independently from $\mathbf{s}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, for any $j \neq k$, meaning they are conditionally independent given the latent mixture component assignment. Thus, the concatenated slots variable $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K)$, can be described by a Kd-dimensional multivariate Gaussian distribution with a block diagonal covariance structure as follows:

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{\pi(1)} \\ \mathbf{s}_{\pi(2)} \\ \vdots \\ \mathbf{s}_{\pi(K)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{\pi(1)} \\ \boldsymbol{\mu}_{\pi(2)} \\ \vdots \\ \boldsymbol{\mu}_{\pi(K)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\pi(1)} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}_{\pi(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\Sigma}_{\pi(K)} \end{bmatrix} \right), \tag{B.30}$$

where $\pi : [K] \to [K]$ is a permutation function of the set $[K] \coloneqq \{1, 2, \ldots, K\}$. Since the slot attention function f_s is permutation equivariant, there exist $K!$ possible ways to concatenate K slots, and each permutation induces a mode

within a Gaussian mixture living in $\mathbb{R}^{Kd}$ space. Since each permutation of the slots is equally likely, the mixture coefficients are given by:

$$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_{K!}), \quad \text{where} \quad \pi_p = \frac{1}{K!} \quad \forall p \in \{1, 2, \ldots, K!\} \tag{B.31}$$

$$\implies \sum_{p=1}^{K!} \pi_p = 1, \tag{B.32}$$

which concludes the proof. □

Remark B.4. *Based on the above result, it is evident that concatenating $K \geq 2$ unique slots can be viewed as a sample from a GMM with $K!$ components. Constructing a scene requires at least two unique slots, one for the background and one for an object, thus supporting our theory regarding slot composition.*

B.3 Injective Decoders: Sufficient Conditions

In this section, we provide a theoretical overview of the decoder architecture we use and offer sufficient conditions for a leaky-ReLU decoder to be *weakly* injective in the sense of Assumption 5.1 as shown and adapted from Kivva et al., 2022.

Definition B.1 (Piece-wise Decoders, Kivva et al., 2022)**.** *Let $\mathbf{s} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K\}$ denote a given set of sampled z_i in each mixture component (K slots) in the GMM, $P(\mathcal{Z})$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ denote the leaky-ReLU activation function, and let $m = n_0, n_1, n_2, \ldots, n_t = n$ and $H(n_1, n_2)$ denote the set of full-rank affine functions $h_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_j}$. We consider piece-wise functions mapping each slot representation $\mathbf{s} \in \mathcal{S} \in \mathbb{R}^m \times K$ to an image $\mathbf{x} \in \mathcal{X} \in \mathbb{R}^n$ in the output space, $\mathcal{F}_\sigma^{mK \to n} : \mathcal{S} \to \mathcal{X}$, of the form below:*

$$\mathcal{F}_\sigma^{n_0, \ldots, n_t} = \Big\{ h_t \circ \sigma \circ h_{t-1} \circ \sigma \circ \cdots \sigma \circ h_1 \mid h_i \in H(n_{i-1}, n_i) \Big\}. \tag{B.33}$$

The following Corollary, corollary and proofs are adapted from Kivva et al., 2022.

Corollary B.2 (Injectivity of mixing function)**.** *Given $f_d \in \mathcal{F}_\sigma^{m \hookrightarrow n}$ where $f_d = h_t \circ \sigma \circ h_{t-1} \circ \sigma \circ \ldots \sigma \circ h_1$, f_d is injective.*

Proof. Each affine function h_i has full column rank and is therefore both injective and invertible. Since the activation function σ is also injective, we get that f_d is injective and invertible. □

Corollary B.3 (Inverse of mixing function)**.** *Let $f_d = h_t \circ \sigma \circ h_{t-1} \circ \sigma \circ \ldots \sigma \circ h_1 \in \mathcal{F}_\sigma^{m \to n}$ where $m = n_0 \leq n_1 \leq \cdots \leq n_k = n$. Given h_i is affine and invertible, then for almost all $x \in f_d(\mathbb{R}^m)$ there exists δ such that f_d^{-1} is a well-defined affine function, h_i on $B(x, \delta) \cap f_d(\mathbb{R}^m)$.*

Proof. We know f_d is a piecewise affine function which is invertible. Therefore, it simply follows, $\forall y \in B(x, \delta)$ there exists δ such that f_d^{-1} is an affine function in the domain $B(x, \delta)$. One can therefore deduce there exists some affine function h_i where $f_d^{-1} = h_i^{-1}$ in the domain $B(x, \delta)$. □

B.4 SLOT IDENTIFIABILITY PROOF

Definition B.2 (Slot Identifiability Brady et al., 2023)**.** *Given a diffeomorphic ground truth function $f : \mathcal{S} \to \mathcal{X}$, and inference model $\hat{g} : \mathcal{X} \to \mathcal{S}$, $\hat{g}$ correctly slot identifies every object $\mathbf{x}_j \in \mathcal{X}$ with the ground slot $\mathbf{s}_k \in \mathcal{S}$ via $\hat{\mathbf{s}}_k = \hat{g}(f(\mathbf{s}_k))$ if there exists a unique slot $k \in [K]$ for all $\mathbf{x}_j \in \mathcal{X}$, and there exists an invertible diffeomorphism, or in our case an affine transformation, such that $\mathbf{s}_k = h(\hat{\mathbf{s}}_k)$ for all $\mathbf{s}_k \in \mathcal{S}$.*

Remark B.5. *Brady et al., 2023 established that compositionality (Assumption B.4) and irreducibility (Assumption B.3) of the decoder f_d is required for slot identifiability in the sense of Definition B.2.*

SUMMARY & INTUITION. The following theorems and proof extend the identifiability results of Kivva et al., 2022 to slot-attention models, we include all the proofs and details for the sake of completion. In this work, we do not consider the irreducibility criteria in Brady et al., 2023 and define slot identifiability only by an injective mapping of each slot to subspaces representing objects in a scene to satisfy compositionality without the use of computationally heavy methods such as additive decoders and compositional contrast. We show identifiability of each slot representation up to affine transformation by passing a concatenation of samples from each mixture component which represents slots through a piecewise function in

Definition B.1 representing the decoder, f_d. The trick in this proof lies in our observation of the fact that a concatenation of samples from each slot mixture component is a sample from a high dimensional GMM $p(\mathbf{s})$, as highlighted in Theorem C.1. We then use the identifiability results of Kivva et al., 2022 to show $p(\mathbf{s})$ is identifiable up to affine transformation. This then implies identifiability up to affine transformation of the aggregate posterior $q(\mathbf{z})$, a non-degenerate GMM by Lemma 4.2 where a sample from a mixture component in $q(\mathbf{z})$ represents an individual slot representation. This contrasts with Kivva et al., 2022 which does not consider identifiability of the aggregate posterior and its mixture components in the context of slot representation learning.

In order to proceed, we begin by stating three key theorems defined and proven in the work of Kivva et al., 2022 which are essential for our slot identifiability proof. First, we restate the definition of a *generic point* as outlined by Kivva et al., 2022 below.

Definition B.3 (Generic Point). *A point $\mathbf{x} \in f_d(\mathbb{R}^m) \subseteq \mathbb{R}^n$ is generic if there exists $\delta > 0$, such that $f_d : B(\mathbf{s}, \delta) \to \mathbb{R}^n$ is affine for every $\mathbf{s} \in f_d^{-1}(\{\mathbf{x}\})$*

Theorem B.1 (Generic Functions Kivva et al., 2022). *Given $f_d : \mathbb{R}^m \to \mathbb{R}^n$ is a piecewise affine function such that $\{\mathbf{x} \in \mathbb{R}^n : |f_d^{-1}(\{\mathbf{x}\})| = \infty\} \subseteq f_d(\mathbb{R}^m)$ has measure zero with respect to the Lebesgue measure on $f_d(\mathbb{R}^m)$, this implies $\dim(f_d(\mathbb{R}^m)) = m$ and almost every point in $f_d(\mathbb{R}^m)$ (with respect to the Lebesgue measure on $f_d(\mathbb{R}^m)$) is generic with respect to f_d.*

Theorem B.2 (Extension of Generic Point, Kivva et al., 2022). *Consider a pair of finite GMMs in $\mathbb{R}^m$:*

$$\mathbf{y} = \sum_{j=1}^{J} \boldsymbol{\pi}_j \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \quad \text{and} \quad \mathbf{y}' = \sum_{j=1}^{J'} \boldsymbol{\pi}'_j \mathcal{N}(\mathbf{y}'; \boldsymbol{\mu}'_j, \boldsymbol{\Sigma}'_j). \tag{B.34}$$

Assume that there exists a ball $B(\mathbf{x}, \delta)$ such that $\mathbf{y}$ and $\mathbf{y}'$ induce the same measure on $B(\mathbf{x}, \delta)$. Then $\mathbf{y} \equiv \mathbf{y}'$, and for some permutation τ we have that $\boldsymbol{\pi}_i = \boldsymbol{\pi}'_{\tau(i)}$ and $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = (\boldsymbol{\mu}'_{\tau(i)}, \boldsymbol{\Sigma}'_{\tau(i)})$.

Theorem B.3 (Equivalence of GMM, Kivva et al., 2022). *Given $\mathbf{z} \sim \sum_{i=1}^{J} \boldsymbol{\pi}_i \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $\mathbf{z}' \sim \sum_{j=1}^{J'} \boldsymbol{\pi}'_j \mathcal{N}(\mathbf{z}'; \boldsymbol{\mu}'_j, \boldsymbol{\Sigma}'_j)$ and $f_d(\mathbf{z})$ and $\tilde{f}_d(\mathbf{z}')$ are equally distributed. We can assume for $\mathbf{x} \in \mathbb{R}^n$ and $\delta > 0$, f_d is invertible on*

$B(\mathbf{x}, 2\delta) \cap f_d(\mathbb{R}^m)$. *This implies that there exists* $\mathbf{x}_1 \in B(\mathbf{x}, \delta)$ *and* $\delta_1 > 0$ *such that both* f_d *and* $\tilde{f}_d$ *are invertible on* $B(\mathbf{x}_1, \delta_1) \cap f_d(\mathbb{R}^m)$.

We next propose our slot identifiability result below.

Theorem 4.2 ($\sim_s$-Identifiable Slot Representations)**.** Given that the aggregate posterior $q(\mathbf{z})$ is an optimal, non-degenerate mixture prior over slot space (Lemma 4.2), $f_d : \mathcal{S} \to \mathcal{X}$ is a piecewise affine weakly injective mixing function (Assumption 5.1), and the slot representation, $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_K)$ can be observed as a sample from a GMM (Theorem C.1), then $p(\mathbf{s})$ is identifiable as per Definition 4.2.

Proof. The proof extends from Kivva et al., 2022 to slot-based models. Given two piece-wise affine functions $f_d, \tilde{f}_d : \mathcal{S} \to \mathcal{X}$, $\forall k \in [K]$, let $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_K), \ni \mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and $\mathbf{s}' = (\mathbf{s}'_1, \ldots, \mathbf{s}'_K), \ni \mathbf{s}'_k \sim \mathcal{N}(\mathbf{s}'_k; \boldsymbol{\mu}'_k, \boldsymbol{\Sigma}'_k)$ be a pair of slot representations constructed by sampling and concatenating each mixture component (i.e. slots) from two distinct GMMs. As proven in Theorem C.1, given individual sampled slots $\mathbf{s}_k$ are conditionally independent given the mixture component k, then a concatenated sample is from a higher dimensional GMM in $\mathbb{R}^{Kd}$. Now, suppose for the sake of argument that $f_d(\mathcal{S})$ and $\tilde{f}_d(\mathcal{S}')$ are equally distributed. We assume that there exists $\mathbf{x} \in \mathcal{X}$ and $\delta > 0$ such that f_d and $\tilde{f}_d$ are invertible and piecewise affine on $B(\mathbf{x}, \delta) \cap f_d(\mathcal{S})$, which implies $\dim f_d(\mathcal{S}) = |\mathcal{S}|$.

We now restrict the space $B(\mathbf{x}, \delta)$ to a subspace $B(\mathbf{x}', \delta')$ where $\mathbf{x} \in B(\mathbf{x}', \delta')$ such that f_d and $\tilde{f}_d$ are now invertible and affine on $B(\mathbf{x}', \delta') \cap f_d(\mathcal{S})$. Next, we let $L \subseteq \mathcal{X}$ be an $|\mathcal{S}|$-dimensional affine subspace (assuming $|\mathcal{X}| \geq |\mathcal{S}|$), such that $B(\mathbf{x}', \delta') \cap f_d(\mathcal{S}) = B(\mathbf{x}', \delta') \cap L$. We also define $h_f, h_{\tilde{f}} : \mathcal{S} \to L$ to be a pair of invertible affine functions where $h_f^{-1}(B(\mathbf{x}', \delta') \cap L) = f_d^{-1}(B(\mathbf{x}', \delta') \cap L)$ and $h_{\tilde{f}}^{-1}(B(\mathbf{x}', \delta') \cap L) = \tilde{f}_d^{-1}(B(\mathbf{x}', \delta') \cap L)$.

Therefore, this implies $h_f(\mathbf{s})$ and $h_{\tilde{f}}(\mathbf{s}')$ are finite GMMs which coincide on $B(\mathbf{x}', \delta') \cap L$ and $h_f(\mathbf{s}) \equiv h_{\tilde{f}}(\mathbf{s}')$ based on Theorem B.2. Given, $h = h_{\tilde{f}}^{-1} \circ h_f$ and $h_f(\mathbf{s})$ and $h_{\tilde{f}}(\mathbf{s}')$ then h is an affine transformation such that $h(\mathbf{s}) = \mathbf{s}'$.

Given Theorems B.1 and C.2, there exists a point $\mathbf{x} \in f_d(\mathcal{S})$ that is generic with respect f_d and $\tilde{f}_d$ and invertible on $B(\mathbf{x}, \delta) \cap f_d(\mathcal{S})$. Having established that there is an affine transformation $h(\mathbf{s}) = \mathbf{s}'$ and two invertible piecewise affine functions f_d and $\tilde{f}_d$ on $B(\mathbf{x}, \delta) \cap f_d(\mathcal{S})$, this implies that $p(\mathbf{s})$ is iden-

tifiable up to an affine transformation and permutation of $\mathbf{s}_k \in \mathbf{s}$, which concludes the proof. □

Corollary 4.1 (Individual Slot Identifiability)**.** If the concatenated slot distribution $p(\mathbf{s})$ is $\sim_s$-identifiable (Theorem 4.2) then this implies $q(\mathbf{z})$ is identifiable up to affine transformation and permutation of the slots, $\mathbf{s}_k$ and therefore each slot distribution $\mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall\, k \in \{1, \dots K\}$ is also identifiable up to an affine transformation.

Proof. Given Theorem B.2, we know that each higher dimensional mixture component in $p(\mathbf{s})$ induces the same measure on $B(\mathbf{x}, \delta)$ and hence for some permutation τ we have that $(\boldsymbol{\mu}_{\pi(i)}, \boldsymbol{\Sigma}_{\pi(i)}) = (\boldsymbol{\mu}'_{\tau(\pi(i))}, \boldsymbol{\Sigma}'_{\tau(\pi(i))})$. Therefore, each mixture component $\mathbf{s}_{\pi(i)}$ is identifiable up to affine transformation, and permutation of slots representations in $\mathbf{s}$. Now, given sampling $\mathbf{s}_k$ is equivalent to obtaining K samples from the GMM, $q(\mathbf{z})$ and concatenating, this makes $q(\mathbf{z})$ identifiable up to affine transformation, h and permutation of slot representations in $\mathbf{s}$.

It now trivially follows that each slot representation $\mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall\, k \in \{1, \dots, K\}$ is identifiable up to affine transformation, h based on the following observed property of GMMs:

$$\sum_{k=1}^{K} \boldsymbol{\pi}_k h_\sharp \left(\mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) \sim h_\sharp\Big(\sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\mathbf{s}'_k; \boldsymbol{\mu}'_k, \boldsymbol{\Sigma}'_k)\Big), \tag{B.35}$$

which concludes the proof. □

B.5 METRICS

SIS: Slot identifiability score Brady et al., 2023, mainly focus on R2 score between ground-truth and the estimated slot representations wrt to maximum R2 score from the models fit between each and every inferred slots. By design SIS requires a model fitting at every validation step to compute this relativistic measure, due to which the metric seems to vary quite a bit across runs, as observed in Figure B.2.

SMCC: Mean correlation coefficient is a well studied metric in disentangled representational learning Khemakhem et al., 2020a; Khemakhem et al., 2020c, we extend this with additional permutational invariance on slot dimensions. SMCC measures the correlation between estimated and ground truth representations (or estimated representations across runs, in the case when ground truth representations are unavailable) once the slots are matched using Hungarian matching. To compute identifiability up to affine transformation along the representational axis and permutation in slots (check definition 4.2), similar to weak identifiability as per the definition in the paper and in Khemakhem et al., 2020b, we use the MCC up to some affine mapping $\boldsymbol{A}$, which we learned by matching slot representations across runs. In summary, SMCC can be computed with the following three steps:

- Matching the order of slots using Hungarian matching;
- Affine transformation of slot representation;
- Followed by computing mean correlation.

Apart from variations described in Figure B.2, we further analyse both the metrics by fixing the model and data; and by computing both the metrics for 10 times. We then considered the mean and variance in the performance, which is reflected as follows: SIS: 35.26 $\pm$ **6.46**, SMCC: 77.83 $\pm$ **0.36**. Here, the resulting variation is only the reflection of the metric, which clearly indicates the stability of SMCC over SIS.

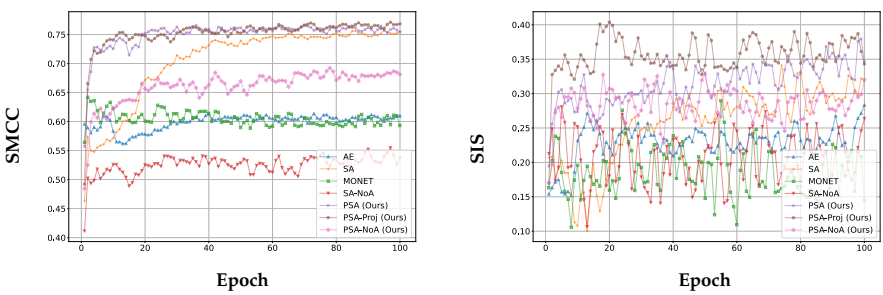


Figure B.2: Identifiability scores throughout training. Our proposed SMCC metric (top) is much more stable than SIS (bottom) in capturing identifiability, and better in discerning differences between methods, showing substantial improvements for PSA.

Table B.1: Quality of compositional image generation with FID measure

Method	CLEVR			Objects-Room		
	FG-ARI ↑	CFID ↓	RFID ↓	FG-ARI ↑	CFID ↓	RFID ↓
SA	0.96 ± 0.01	-	41.81 ± 2.82	0.79 ± 0.06	-	16.49 ± 3.34
SA-NoA	0.62 ± 0.02	-	81.21 ± 8.72	0.41 ± 0.12	-	96.64 ± 21.33
PSA-NoA	0.84 ± 0.01	36.42 ± 8.53	68.02 ± 10.21	0.78 ± 0.04	54.55 ± 1.43	21.37 ± 1.03
PSA	0.85 ± 0.02	28.50 ± 4.27	52.70 ± 1.74	0.78 ± 0.02	20.49 ± 2.36	21.00 ± 1.43
PSA-Proj	0.95 ± 0.00	28.79 ± 5.50	39.42 ± 8.29	0.81 ± 0.04	34.58 ± 4.32	16.85 ± 5.68

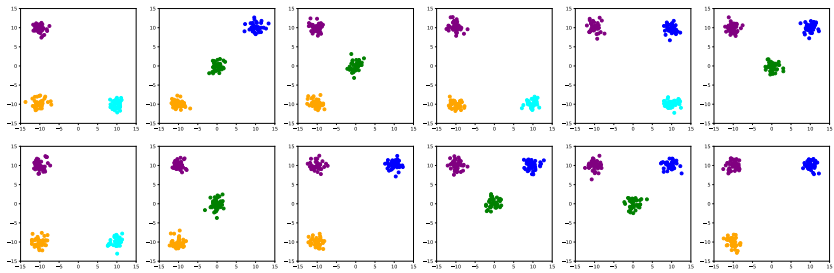


Figure B.3: Random samples from the 2D synthetic dataset used in our aggregate posterior identifiability experiments. As outlined in the main text, there are in total five 'object' clusters in the dataset, and each observation contains at most three of the clusters.

B.6 Automatic relevance determination of slots

For evaluation, we calculate the MAE between the estimated and ground truth numbers of slots and measure the reduction in FLOPs achieved using the proposed ARD method. We observe MAE values of 1.03 and 0.58, and significant savings of up to **41%** and **62%** in FLOPs on the CLEVR and Objects-Room datasets respectively, when compared to using a fixed number of slots K.

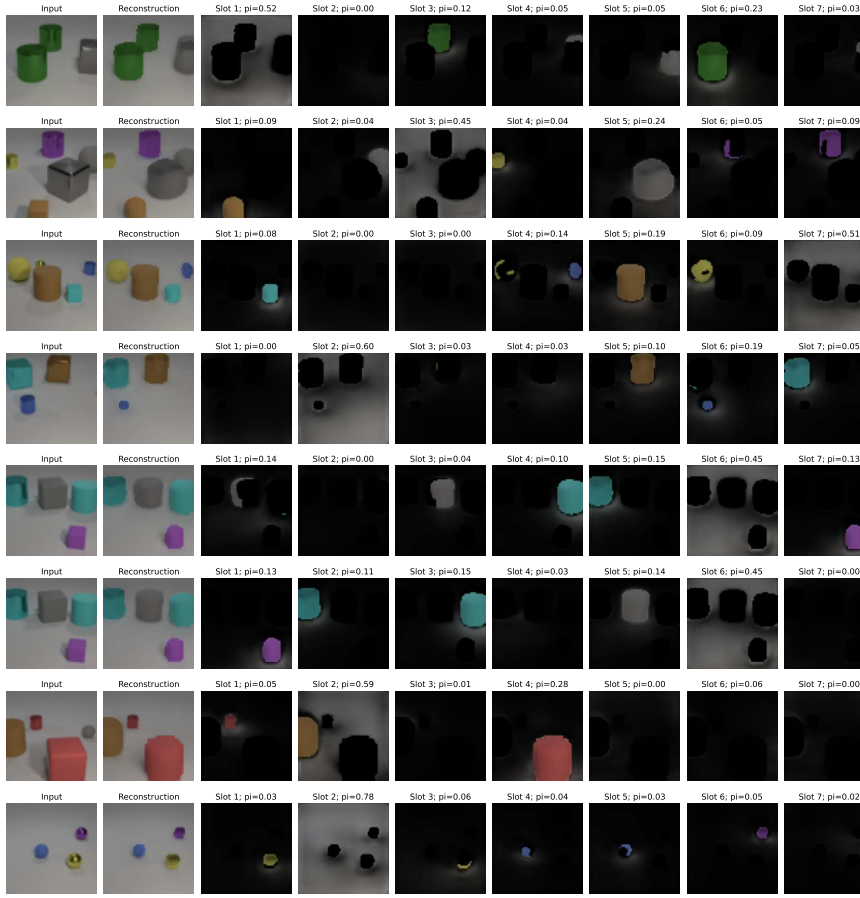


Figure B.4: Demonstration of ARD on CLEVR dataset.

B.7 COMPOSITIONALITY RESULTS

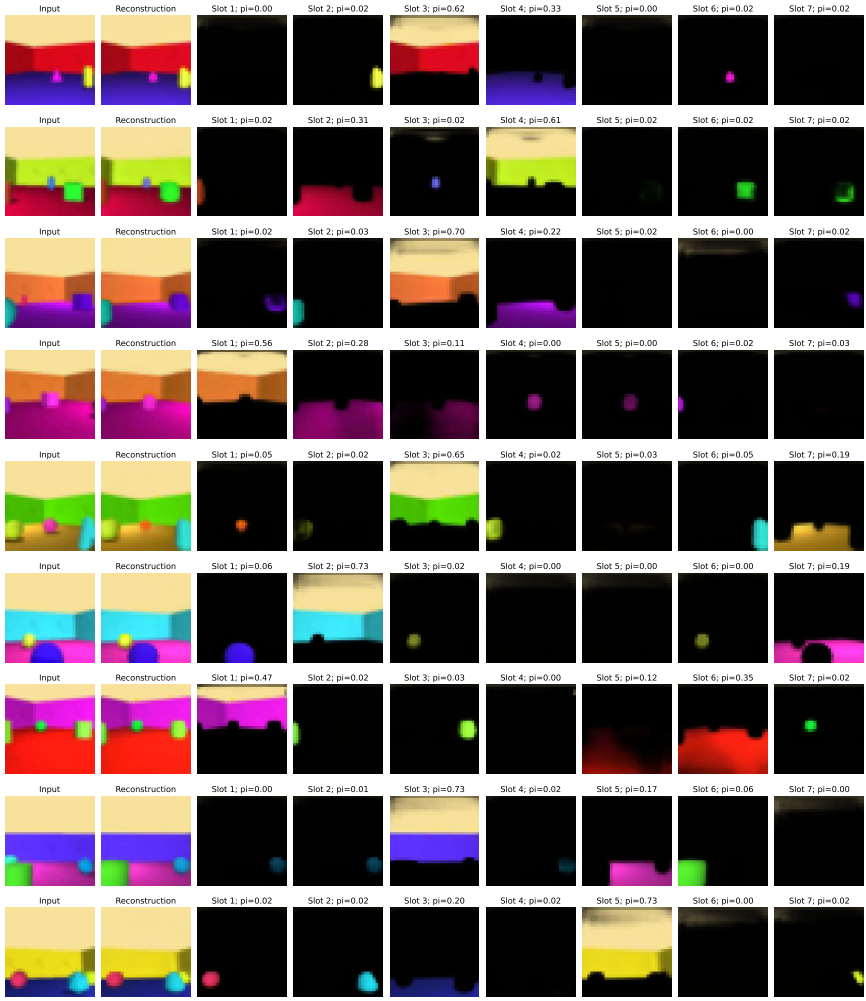


Figure B.5: Automatic Relevance Determination (ARD) of slots on the ObjectsRoom dataset. As shown, when using our proposed probabilistic slot attention (Algorithm 5), the mixing coefficients $\boldsymbol{\pi}_i \in \mathbb{R}^K, \forall i$ automatically approach zero when slots are inactive.

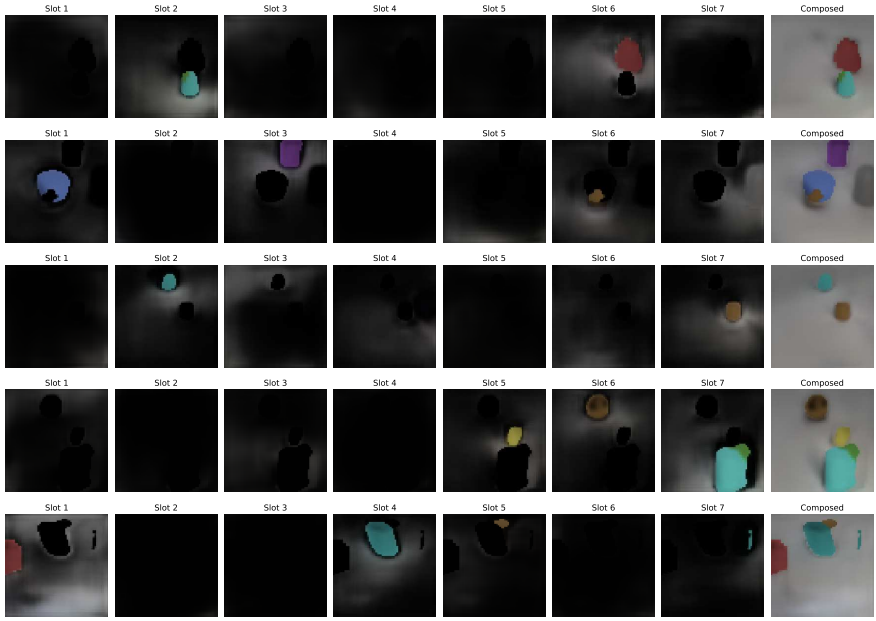


Figure B.6: Aggregate posterior sampling for image composition on CLEVR dataset.

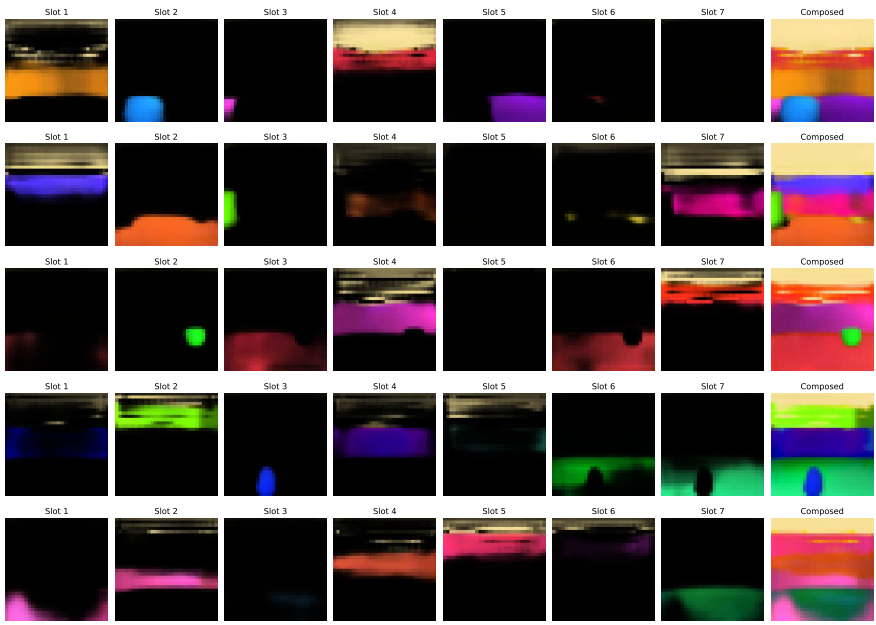


Figure B.7: Aggregate posterior sampling for image composition on ObjectsRoom dataset.

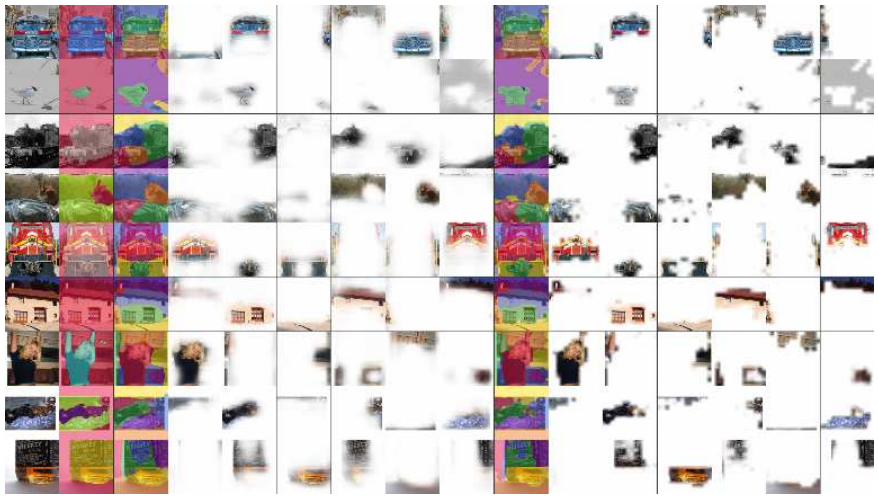


Figure B.8: Visualizations of attention and alpha masks on the Pascal VOC2012 dataset are shown, with alpha masks on the left and attention masks on the right, for a PSA-MLP model using a DINO feature extractor. In the figure above, the images from left to right represent: the original image, ground-truth segmentation, alpha mask segmentation, individual entities grouped in the alpha mask, slot attention segmentation mask, and individual entities grouped in the slot attention mask.



Figure B.9: Visualizations of attention and alpha masks on the Pascal VOC2012 dataset are shown, with alpha masks on the left and attention masks on the right, for a PSA-Transformer model using a DINO feature extractor. In the figure above, the images from left to right represent: the original image, ground-truth segmentation, alpha mask segmentation, individual entities grouped in the alpha mask, slot attention segmentation mask, and individual entities grouped in the slot attention mask.

C

VISA-APPENDIX

C.1 NOTATIONS

Table C.1: Notation and definitions used in the model.

Symbol	Description
$\mathcal{O}^v$	Abstract object set as observed from viewpoint v.
$[V] = \{1, \ldots, V\}$	Exhaustive set of viewpoints, representing all possible views.
$A, B \subset [V]$	Subsets of viewpoints, used for training.
$\mathcal{X} = \times_{v \in A} \mathcal{X}^v$	Data space, formed by the Cartesian product of data spaces for each view in subset A.
$\mathbf{x}^A = \{\mathbf{x}^v : \forall v \in A\} \in \mathcal{X}$	Data sample, where $\mathbf{x}^v$ is the data from view v, and $\mathbf{x}^A$ represents the set of data across all views in A.
f_e	Encoder model that maps input data to a latent space or feature representation.
$\mathbf{z}$	Spatial latent features, representing inferred spatial properties from the data.
$\mathcal{S}$	View-specific slot space, where features are tied to particular viewpoints.
$\mathcal{C}$	View-invariant content space, representing features that remain consistent across different viewpoints.
$\mathbf{s} \in \mathcal{S}$	Samples from the view-specific slot space, representing view-dependent latent features.
$\mathbf{c} \in \mathcal{C}$	Samples from the view-invariant content space, representing features that remain unchanged across views.

Symbol	Description
$f_s, \tilde{f}_s$	Slot attention module responsible for attending to and disentangling different parts of the input related to different views.
$f_d, \tilde{f}_d$	Mixing function that combines view-specific and view-invariant features into a unified representation.
$\mathcal{V}$	View information space that encodes information specific to each viewpoint (e.g., angle, position).
$\mathbf{v} \in \mathcal{V}$	A sample from the view information space, representing a specific view or camera configuration.
$f_v, \tilde{f}_v$	View extractor function that extracts viewpoint-related information from the data.
$\boldsymbol{\mu}_c, \boldsymbol{\mu}_s, \boldsymbol{\mu}_v$	Mean of invariant content, view-specific slots, and view distributions.
$\boldsymbol{\sigma}_c, \boldsymbol{\sigma}_s, \boldsymbol{\sigma}_v$	Standard deviation of invariant content, view-specific slots, and view distributions.
$\boldsymbol{\pi}_c, \boldsymbol{\pi}_s, \boldsymbol{\pi}_v$	Mixing coefficients of invariant content, view-specific slots, and view distributions.
A_{nk}	Assignment confidence of a slot k getting mapped to token n.
$\boldsymbol{P} \in \mathcal{P} \subseteq \{0,1\}^{K \times K}$	Permutation matrix.
m_s	Matching function used to align object representations across views.
Δ^K	Simplex in the space of dimension K.
$\mathcal{H}_x, \mathcal{H}_v$	Space of homeomorphic transformations.

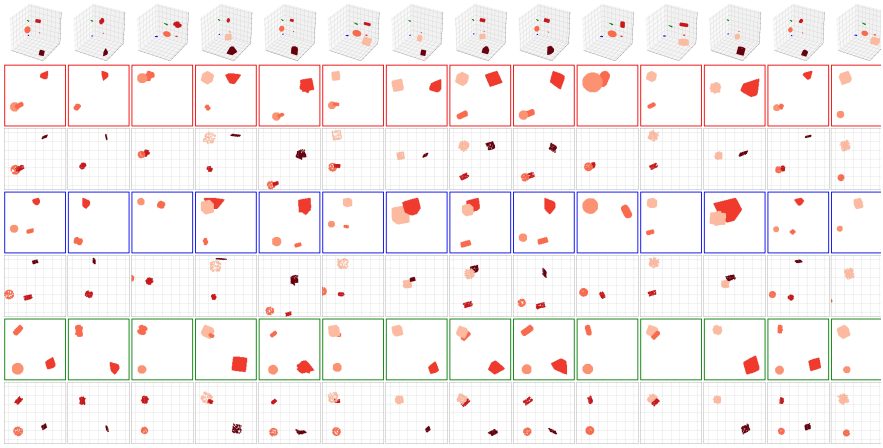


Figure C.1: Data generating process: The figure illustrates 3D point cloud data in the first row, with camera location highlighted in red, blue, and green arrow. Following rows indicates projected images and point cloud as observed from red, blue, and green cameras, respectively.

C.2 DATASETS

C.2.1 *Illustrative dataset*

To visually illustrate the effectiveness of our theory we experiment with two dimensional illustrative dataset. For this, similar to Kori et al., 2024c, we defined a $K = 5$ component GMM, with differing mean parameters $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_5\}$, and shared isotropic covariances, which we use to sample locations for an object. For a given location, we randomly select one object from `{'cube', 'cylinder', 'pyramid', 'sphere'}` and generate 1000 random points on the surface of the selected shape uniformly covering it. To create a single data point, we randomly select three of the five locations and place a randomly selected object at the location. To include multiple viewpoints, we consider V camera location and project the objects, creating V different scenes. We then fill this by considering convex hull operation resulting in projected images as illustrated in Fig. C.1. To maintain uniformity, we only use imaging modality in the main paper while also demonstrating point cloud illustrations here in the appendix. We use different colours representing different objects in Fig. C.2, C.3 and used $10,000$ data points in total to train our toy models. Unlike existing

benchmark datasets, here we remove all the confounding effects caused by lighting and depth. This provides an ideal test bed to validate all our theoretical claims.

C.2.2 *Proposed dataset*

In this work, we introduce the MV-MOVI datasets, created using Kubric Greff et al., 2022, which feature multi-view scenes with segmentation annotations. We propose two variants of the dataset: MV-MOVIC, where the camera locations for every viewpoint remain fixed across all scenes, and MV-MOVID, where the camera locations dynamically change for each scene.

Both MV-MOVIC and MV-MOVID primarily consist of scenes generated by randomly selecting a background from a set of **458** available options and choosing K objects, where $3 \leq K \leq 6$, from a pool of **930** objects. In total, a significantly high number of images can be generated in general. In contrast, for this work, we generate **72,000** scenes, each captured from 5 different viewpoints, with object segmentation masks for every view to facilitate the evaluation of model performance. In the case of MV-MOVIC, the locations of all five cameras are fixed across the $72,000$ scenes, while in MV-MOVID, the camera positions are dynamically sampled and vary across scenes.

C.3 MASK GENERATION

In the case of additive decoders, the decoder outputs K three channelled tensors along with K single channelled mask. We consider normalising these masks with `softmax` transformation along slot dimension, ensuring that each pixel only contributes to a single slot. The resulting softmaxed masks are used in composing ($\text{image} = \sum_k \text{mask}_k \cdot \text{image}_k$) the slots to reconstruct an image for training. During inference, we normalise masks with `sigmoid` transformation, allowing us to estimate occluded objects visually, resolving the spatial ambiguities with occluded objects. In a later section, we illustrate the results with both `softmax` and `sigmoid` transformations.

C.3.1 *Additivity Implications*

Definition C.1 (Additive models)**.** *Function f is considered to be an additive decoder if, for any object decoders f_{obj} and masking mechanism m_{obj}, if they can be expressed as:*

$$f(\mathbf{z}) = \sum_{k \in [K]} m_{\text{obj}}(\mathbf{z}_k) \odot f_{\text{obj}}(\mathbf{z}_k) \tag{C.1}$$

As pointed out in Lachapelle et al., 2023, `softmax`-based masks do not truly fall under the category of additive decoders due to the competition between masks for groups of pixels. This implies that the additive decoders studied in Lachapelle et al., 2023 are not expressive enough to represent the "masked decoders" typically employed in object-centric representation learning. The issue arises from the normalization of alpha masks, and care must be taken when extrapolating the findings from Lachapelle et al., 2023 to the models used in practice.

Although `sigmoid`-based masks satisfy the condition of additivity during inference, it is important to note that the model is still trained using `softmax` normalization in our setting. The effect of using `sigmoid` masks during inference can be visually observed in Appendix C.5.

C.4 PROOFS

Lemma C.1 (ELBO)**.** *With prior distributions $p(\mathbf{v})$ and $p(\mathbf{c})$ for view and content latent random variables, the likelihood $p(\mathbf{x})$ can be maximised by maximising the following expression:*

$$\log p(\mathbf{x}) \geq \mathbb{E} \log p(\mathbf{x} \mid \mathcal{T}_{\theta^v}(\mathbf{c}), \mathbf{v}) - \mathrm{KL}\left(q(\mathbf{v} \mid \mathbf{x}) \parallel p(\mathbf{v})\right) := \mathrm{ELBO}(\mathbf{x}) \tag{C.2}$$

Proof. Considering the generative model in Eqn. 4.12 respecting the graphical model in Fig. 5.3, we get:

$$p(\mathbf{x}) = \iint p(\mathbf{x}^A \mid \mathcal{T}_{\theta^v}(\mathbf{c}), \mathbf{v}^A)\ p(\mathbf{c})\ p(\mathbf{v}^A)\ d\mathbf{v}\ d\mathbf{c} \tag{C.3}$$

$$\log p(\mathbf{x}) = \log \iint p(\mathbf{x}^A \mid \mathbf{c}_{1:K}, \mathbf{v}^A)\ p(\mathbf{c}_{1:K})\ p(\mathbf{v}^A) \frac{q(\mathbf{v}, \mathbf{c} \mid \mathbf{x}^A)}{q(\mathbf{v}, \mathbf{c} \mid \mathbf{x}^A)}\ d\mathbf{v}\ d\mathbf{c}_{1:K} \tag{C.4}$$

$$\geq \iint q(\mathbf{v}^A \mid \mathbf{x}^A) q(\mathbf{c}_{1:K} \mid \mathcal{T}_{\theta^v}^{-1}(\mathbf{x}^A)) \log p(\mathbf{x}^A \mid \mathcal{T}_{\theta^v}(\mathbf{c})_{1:K}, \mathbf{v}^A) \frac{p(\mathbf{v}^A_{1:K})}{q(\mathbf{v}^A \mid \mathbf{x}^A)} \frac{p(\mathbf{c}^A_{1:K})}{q(\mathbf{c}_{1:K} \mid \mathbf{x}^A)}\ d\mathbf{v}^A\ d\mathbf{c}_{1:K} \tag{C.5}$$

$$= \sum_{v \in A} \iint q(\mathbf{v}^v \mid \mathbf{x}^v) q(\mathbf{c}_{1:K} \mid \mathcal{T}_{\theta^v}^{-1}(\mathbf{x}^v))) \log p(\mathbf{x}^v \mid \mathcal{T}_{\theta^v}(\mathbf{c})_{1:K}, \mathbf{v}^v) \frac{p(\mathbf{v}^A_{1:K})}{q(\mathbf{v}^A \mid \mathbf{x}^v)} \frac{p(\mathbf{c}^A_{1:K})}{q(\mathbf{c}_{1:K} \mid \mathbf{x}^v)}\ d\mathbf{v}^v\ d\mathbf{c}_{1:K} \tag{C.6}$$

Given the iterative update for $\mathbf{c}$ with EM algorithm, ideally we expect posterior to converge to prior, which results in:

$$\log p(\mathbf{x}) = \sum_{v \in A} \iint q(\mathbf{v}^v \mid \mathbf{x}^v) q(\mathbf{c}_{1:K} \mid \mathcal{T}_{\theta^v}^{-1}(\mathbf{x}^v))) \log p(\mathbf{x}^v \mid \mathcal{T}_{\theta^v}(\mathbf{c})_{1:K}, \mathbf{v}^v) \frac{p(\mathbf{v}^A_{1:K})}{q(\mathbf{v}^A \mid \mathbf{x}^v)}\ d\mathbf{v}^v\ d\mathbf{c}_{1:K} \tag{C.7}$$

$$= \sum_{v \in A} \mathbb{E}_{\mathbf{c},\mathbf{v}} \log p(\mathbf{x}^v \mid \mathcal{T}_{\theta^v}(\mathbf{c}), \mathbf{v}) - \mathrm{KL}\left(q(\mathbf{v} \mid \mathbf{x}^v) \parallel p(\mathbf{v})\right) \tag{C.8}$$

Given the subscript notation, the above expression can also be expressed as:

$$\mathbb{E}_{\mathbf{c},\mathbf{v}} \log p(\mathbf{x} \mid \mathcal{T}_{\theta^v}(\mathbf{c}), \mathbf{v}) - \mathrm{KL}\left(q(\mathbf{v} \mid \mathbf{x}) \parallel p(\mathbf{v})\right) := \mathrm{ELBO}(\mathbf{x}) \tag{C.9}$$

□

Lemma C.2 (Mean GMM). *Let $\mathbf{z} \in \mathbb{R}^{N \times d}$ be a random variable drawn from a GMM with K components:*

$$\mathbf{z} \sim \sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{C.10}$$

where $\boldsymbol{\pi}_k$ are the mixture weights, $\boldsymbol{\mu}_k \in \mathbb{R}^d$ are the mean vectors, and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d\times d}$ are the covariance matrices. Assuming the mixture satisfies the ICA assumption, such that the components of $\mathbf{z}$ are statistically independent. A projected random variable $\bar{z}$ as the average over the dimensions of $\mathbf{z}$:

$$\bar{z} = \frac{1}{d}\sum_{j=1}^{d} \mathbf{z}_j, \tag{C.11}$$

is also distributed according to a GMM with K components, with appropriately transformed means and variances.

Proof. Given the random variable $\mathbf{z}$ follows a GMM, so its density can be expressed as:

$$p(\mathbf{z}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{C.12}$$

where:

$$\boldsymbol{\mu}_k = [\mu_{k,1}, \mu_{k,2}, \ldots, \mu_{k,d}]^\top; \quad \boldsymbol{\Sigma}_k = \operatorname{diag}([\sigma^2_{k,1}, \sigma^2_{k,2}, \ldots, \sigma^2_{k,d}]). \tag{C.13}$$

Considering, the projection of $\mathbf{z}$ onto $\bar{z}$ is defined as:

$$\bar{z} = \frac{1}{d}\sum_{j=1}^{d} \mathbf{z}_j. \tag{C.14}$$

Given the ICA assumption, the components $\mathbf{z}_{:,j}$ are independent. For a fixed component k, the projected mean and variance of $\bar{z}$ can be derived as:

$$\mathbb{E}[\bar{z}] = \frac{1}{d}\sum_{j=1}^{d} \mu_{k,j}; \qquad \operatorname{Var}(\bar{z}) = \frac{1}{d^2}\sum_{j=1}^{d} \sigma^2_{k,j}. \tag{C.15}$$

Since the projection $\bar{z}$ is a linear combination of independent Gaussian variables, $\bar{z}$ remains Gaussian for each component k. Thus, the overall distribution of $\bar{z}$ is also a GMM:

$$p(\bar{z}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\bar{z}; \mu_{\bar{z},k}, \sigma^2_{\bar{z},k}), \tag{C.16}$$

where:

$$\mu_{\bar{z},k} = \frac{1}{d}\sum_{j=1}^{d}\mu_{k,j}; \qquad \sigma^2_{\bar{z},k} = \frac{1}{d^2}\sum_{j=1}^{d}\sigma^2_{k,j}. \tag{C.17}$$

This concludes the proof.

□

Lemma C.3 (Convex Combination of GMMs). *Let* $\mathbf{s}^1 = \{\mathbf{s}^1_1, \ldots, \mathbf{s}^1_K\}$ *and* $\mathbf{s}^2 = \{\mathbf{s}^2_1, \ldots, \mathbf{s}^2_K\}$ *be two sets of K random vectors in* $\mathbb{R}^d$*, each distributed according to GMMs:*

$$\mathbf{s}^1 \sim \sum_{k=1}^{K}\boldsymbol{\pi}_{1,k}\mathcal{N}(\boldsymbol{\mu}_{1,k}, \boldsymbol{\Sigma}_{1,k}); \quad \mathbf{s}^2 \sim \sum_{k=1}^{K}\boldsymbol{\pi}_{2,k}\mathcal{N}(\boldsymbol{\mu}_{2,k}, \boldsymbol{\Sigma}_{2,k}) \tag{C.18}$$

where $\boldsymbol{\mu}_{i,k} \in \mathbb{R}^d$*,* $\boldsymbol{\Sigma}_{i,k} \in \mathbb{R}^{d\times d}$*, and* $\boldsymbol{\pi}_{i,k}$ *are the means, covariances, and mixing coefficients respectively.*

Then for any weights $w_1, w_2 \in \mathbb{R}$ *such that* $w_1 + w_2 = 1$*, the convex combination* $\mathbf{s} = w_1\mathbf{s}^1 + w_2\mathbf{s}^2$ *is also distributed according to a GMM with K components.*

Proof. Without loss of generality, assume the components of both GMMs are aligned. For each component k, we derive the parameters of the resulting mixture:

The mixing coefficients of the resulting GMM are weighted combinations of the original coefficients:

$$\tilde{\boldsymbol{\pi}}_k = w_1\boldsymbol{\pi}_{1,k} + w_2\boldsymbol{\pi}_{2,k} \tag{C.19}$$

For each component k, the convex combination of Gaussians results in a Gaussian distribution. The mean of the resulting Gaussian is:

$$\tilde{\boldsymbol{\mu}}_k = \frac{w_1\boldsymbol{\pi}_{1,k}\boldsymbol{\mu}_{1,k} + w_2\boldsymbol{\pi}_{2,k}\boldsymbol{\mu}_{2,k}}{\tilde{\boldsymbol{\pi}}_k} \tag{C.20}$$

The covariance of the resulting Gaussian for each component k can be derived as follows. Firstly, recall that for a random variable X, the covariance is:

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^\top] = \mathbb{E}[XX^\top] - \mathbb{E}[X]\mathbb{E}[X]^\top \tag{C.21}$$

First lets compute $\mathbb{E}[\mathbf{s}_k \mathbf{s}_k^\top]$:

$$\mathbb{E}[\mathbf{s}_k \mathbf{s}_k^\top] = \mathbb{E}\left[\left(\frac{w_1 \boldsymbol{\pi}_{1,k} \mathbf{s}_k^1 + w_2 \boldsymbol{\pi}_{2,k} \mathbf{s}_k^2}{\tilde{\boldsymbol{\pi}}_k}\right)\left(\frac{w_1 \boldsymbol{\pi}_{1,k} \mathbf{s}_k^1 + w_2 \boldsymbol{\pi}_{2,k} \mathbf{s}_k^2}{\tilde{\boldsymbol{\pi}}_k}\right)^\top\right] \tag{C.22}$$

$$= \frac{w_1^2 \boldsymbol{\pi}_{1,k}^2 \mathbb{E}[\mathbf{s}_k^1 (\mathbf{s}_k^1)^\top] + w_2^2 \boldsymbol{\pi}_{2,k}^2 \mathbb{E}[\mathbf{s}_k^2 (\mathbf{s}_k^2)^\top]}{(\tilde{\boldsymbol{\pi}}_k)^2} \tag{C.23}$$

$$+ \frac{w_1 w_2 \boldsymbol{\pi}_{1,k} \boldsymbol{\pi}_{2,k}}{(\tilde{\boldsymbol{\pi}}_k)^2} \mathbb{E}[\mathbf{s}_k^1 (\mathbf{s}_k^2)^\top + \mathbf{s}_k^2 (\mathbf{s}_k^1)^\top] \tag{C.24}$$

Then, substitute known expectations:

$$\mathbb{E}[\mathbf{s}_k^i (\mathbf{s}_k^i)^\top] = \boldsymbol{\Sigma}_{i,k} + \boldsymbol{\mu}_{i,k} \boldsymbol{\mu}_{i,k}^\top \tag{C.25}$$

$$\mathbb{E}[\mathbf{s}_k^1 (\mathbf{s}_k^2)^\top] = \boldsymbol{\mu}_{1,k} \boldsymbol{\mu}_{2,k}^\top \tag{C.26}$$

Finally, by subtract $\mathbb{E}[\mathbf{s}_k^i]\mathbb{E}[\mathbf{s}_k^i]^T = \tilde{\boldsymbol{\mu}}_k \tilde{\boldsymbol{\mu}}_k^\top$ we get the covariance:

$$\tilde{\boldsymbol{\Sigma}}_k = \frac{w_1^2 \boldsymbol{\pi}_{1,k}^2 \boldsymbol{\Sigma}_{1,k} + w_2^2 \boldsymbol{\pi}_{2,k}^2 \boldsymbol{\Sigma}_{2,k}}{(\tilde{\boldsymbol{\pi}}_k)^2} \tag{C.27}$$

To verify this forms a non degenerate GMM, we show the mixing coefficients sum to 1:

$$\sum_{k=1}^{K} \tilde{\boldsymbol{\pi}}_k = \sum_{k=1}^{K} (w_1 \boldsymbol{\pi}_{1,k} + w_2 \boldsymbol{\pi}_{2,k}) \tag{C.28}$$

$$= w_1 \sum_{k=1}^{K} \boldsymbol{\pi}_{1,k} + w_2 \sum_{k=1}^{K} \boldsymbol{\pi}_{2,k} \tag{C.29}$$

$$= w_1 \cdot 1 + w_2 \cdot 1 = 1 \tag{C.30}$$

Therefore, the convex combination results in a valid Gaussian mixture model with K components, where each component has mean $\tilde{\boldsymbol{\mu}}_k$, covariance $\tilde{\boldsymbol{\Sigma}}_k$, and mixing coefficient $\tilde{\boldsymbol{\pi}}_k$. □

LEMMA 4.2. (Optimal Content Mixture) For $A \in [V]$, given the a local content distribution $q(\mathbf{c}_{1:K} \mid \mathbf{s}^A_{1:K}, \mathbf{x}^A)$ (per-scene $\mathbf{x}^A \in \{\mathbf{x}^A_i\}_{i=1}^M$), which can be expressed as a GMM with K components, the aggregate posterior $q(\mathbf{c})$ is obtained by marginalizing out $\mathbf{x}, \mathbf{s}$ is a non-degenerate global Gaussian mixture with MK components:

$$p(\mathbf{c}) = q(\mathbf{c}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\pi}_{ik} \mathcal{N}\left(\mathbf{c}; \widehat{\mu}_{ik}, \widehat{\sigma}^2_{ik}\right). \tag{C.31}$$

Proof. We extend the proof in Kori et al., 2024c, by incorporating hierarchical slot to aggregate content formalisation. For which, we begin by noting that the aggregate posterior $q(\mathbf{c})$ is the optimal prior $p(\mathbf{c})$ so long as our posterior approximation $q(\mathbf{c} \mid \mathbf{s}^A, \mathbf{x}^A)$ is close enough to the true posterior $p(\mathbf{c} \mid \mathbf{s}^A, \mathbf{x}^A)$, since for a dataset $\mathbf{x}^A \in \{\mathbf{x}^A_i\}_{i=1}^M$, for which we start with $q(\mathbf{s}^A \mid \mathbf{x}^A)$, wlog, given view point transformation is deterministic, we consider $\mathbf{x}^A = \mathcal{T}_{\theta^A}(\mathbf{x}^A)$ we have that:

$$\begin{aligned}
p(\mathbf{s}^A) &= \int p(\mathbf{s}^A \mid \mathbf{x}^A) p(\mathbf{x}^A) d\mathbf{x}^A && \text{(C.32)} \\
&= \mathbb{E}_{\mathbf{x}^A \sim p(\mathbf{x}^A)} \left[p(\mathbf{s}^A \mid \mathbf{x}^A) \right] && \text{(C.33)} \\
&\approx \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{s}^A \mid \mathbf{x}^A_i) \quad \text{(empirical approximation)} && \text{(C.34)} \\
&\approx \frac{1}{M} \sum_{i=1}^{M} q(\mathbf{s}^A \mid \mathbf{x}^A_i) \quad \text{(posterior approximation)} && \text{(C.35)} \\
&=: q(\mathbf{s}^A), && \text{(C.36)}
\end{aligned}$$

We further extend this to $q(\mathbf{c})$, with the result from Lemma C.3, we know that the $q(\mathbf{c} \mid \mathbf{s}^A)$ is a GMM with same number of components as $q(\mathbf{s}^v \mid \mathbf{s}^v)$ for any $v \in [V]$ as follows

$$p(\mathbf{c}) = \int p(\mathbf{c} \mid \mathbf{s}^A) p(\mathbf{s}^A) d\mathbf{s}^A \tag{C.37}$$

$$= \mathbb{E}_{\mathbf{s}^A \sim p(\mathbf{s}^A)} \left[p(\mathbf{c} \mid \mathbf{s}^A) \right] \tag{C.38}$$

$$\approx \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{c} \mid \mathbf{s}_i^A) \tag{C.39}$$

$$\approx \frac{1}{M} \sum_{i=1}^{M} q(\mathbf{c} \mid \mathbf{s}_i^A) \tag{C.40}$$

$$=: q(\mathbf{c}), \tag{C.41}$$

where we approximated $p(\mathbf{x})$ using the empirical distribution, then substituted in the approximate posterior, marginalizing $\mathbf{x}$ to get $p(\mathbf{s})$, we later consider the distributional form of $p(\mathbf{s})$ and marginalise $\mathbf{s}^A$ to get $p(\mathbf{c})$. This observation was first made by Hoffman and Johnson, 2016 and was used in Kori et al., 2024c we use it to motivate our setup. Given our model fits a local GMM to each latent variable sampled from the approximate posterior:

$\mathbf{z}^A \sim q(\mathbf{z}^A \mid \mathbf{x}_i^A)$, for $i = 1, \ldots, M$. Let $f_s(\mathbf{z}^A)$ denote the (local) the product of GMM density, its expectation is given by:

$$\mathbb{E}_{p(\mathbf{x}^A), q(\mathbf{z}^A|\mathbf{x}^A)}\left[f_s(\mathbf{z}^A)\right] = \iint p(\mathbf{x}^A) q(\mathbf{z}^A \mid \mathbf{x}^A) f_s(\mathbf{z}^A) d\mathbf{x}^A d\mathbf{z}^A \tag{C.42}$$

$$\approx \iint \frac{1}{M} \sum_{i=1}^{M} \delta(\mathbf{x}^A - \mathbf{x}_i^A) q(\mathbf{z}^A \mid \mathbf{x}^A) f(\mathbf{z}^A) d\mathbf{x}^A d\mathbf{z}^A \tag{C.43}$$

$$= \int \frac{1}{M} \sum_{i=1}^{M} q(\mathbf{z}^A \mid \mathbf{x}_i^A) f(\mathbf{z}^A) d\mathbf{z}^A \tag{C.44}$$

$$= \int \frac{1}{M} \sum_{i=1}^{M} \mathcal{N}\left(\mathbf{z}^A; \boldsymbol{\mu}(\mathbf{x}_i^A), \sigma^2(\mathbf{x}_i^A)\right) \cdot \sum_{k=1}^{K} \pi_k(\mathbf{x}_i^A) \mathcal{N}\left(\mathbf{z}^A; \boldsymbol{\mu}_k(\mathbf{x}_i^A), \sigma_k^2(\mathbf{x}_i^A)\right) d\mathbf{z}^A \tag{C.45}$$

$$\approx \int \frac{1}{M} \sum_{i=1}^{M} \delta(\mathbf{z}^A - \boldsymbol{\mu}(\mathbf{x}_i^A)) \cdot \sum_{k=1}^{K} \pi_k(\mathbf{x}_i^A) \mathcal{N}\left(\mathbf{z}^A; \boldsymbol{\mu}_k(\mathbf{x}_i^A), \sigma_k^2(\mathbf{x}_i^A)\right) d\mathbf{z}^A \tag{C.46}$$

$$= \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \pi_k(\mathbf{x}_i^A) \mathcal{N}\left(\mathbf{z}^A; \boldsymbol{\mu}_k(\mathbf{x}_i^A), \sigma_k^2(\mathbf{x}_i^A)\right) \tag{C.47}$$

$$=: q(\mathbf{z}^A), \tag{C.48}$$

where we again used the empirical distribution approximation of $p(\mathbf{x})$, and the following basic identity of the Dirac delta to simplify: $\int \delta(\mathbf{x} - \mathbf{x}') f_e(\mathbf{x}) d\mathbf{x} = f_e(\mathbf{x}')$.

For the general case, however, we must instead compute the product of $q(\mathbf{z}^A \mid \mathbf{x}^A)$ and $f_s(\mathbf{z}^A)$ rather than use a Dirac delta approximation as in

Eqn. C.46. To that end we may proceed as follows w.r.t. to each datapoint $\mathbf{x}_i^A$:

$$q(\mathbf{z}^A \mid \mathbf{x}_i^A) \cdot f_s(\mathbf{z}^A) = \mathcal{N}\big(\mathbf{z}^A; \boldsymbol{\mu}(\mathbf{x}_i^A), \boldsymbol{\sigma}^2(\mathbf{x}_i^A)\big) \cdot \sum_{k=1}^{K} \boldsymbol{\pi}_k(\mathbf{x}_i^A) \mathcal{N}\left(\mathbf{z}^A; \boldsymbol{\mu}_k(\mathbf{x}_i^A), \boldsymbol{\sigma}_k^2(\mathbf{x}_i^A)\right) \tag{C.49}$$

$$= \sum_{k=1}^{K} \boldsymbol{\pi}_k(\mathbf{x}_i^A) \mathcal{N}\big(\mathbf{z}^A; \boldsymbol{\mu}(\mathbf{x}_i^A), \boldsymbol{\sigma}^2(\mathbf{x}_i^A)\big) \cdot \mathcal{N}\left(\mathbf{z}^A; \boldsymbol{\mu}_k(\mathbf{x}_i^A), \boldsymbol{\sigma}_k^2(\mathbf{x}_i^A)\right) \tag{C.50}$$

Given that means across all views are aligned, similar to Lemma C.3, we know the resulting combined GMM has same number of components:

$$q(\mathbf{z}^A \mid \mathbf{x}_i^A) \cdot f_s(\mathbf{z}^A) = \prod_{v=1}^{|A|} \sum_{k=1}^{K} \bar{\boldsymbol{\pi}}_{ik}^v \mathcal{N}\left(\mathbf{z}; \bar{\boldsymbol{\mu}}_{ivk}, \bar{\boldsymbol{\sigma}}_{ivk}^2\right), \tag{C.51}$$

Given the product of GMM is a GMM with the number of components equal to the product of a number of components in individual GMM, however in our setting we consider all the components in individual GMM across viewpoints to be aligned resulting in GMM with a number of components equal to the sum of individual components which in our case correspond to K. The posterior parameters of the resulting mixture are given in closed form by:

$$\bar{\boldsymbol{\sigma}}_{ivk}^2 = \left(\frac{1}{\boldsymbol{\sigma}_k^2(\mathbf{x}_i^v)} + \frac{1}{\boldsymbol{\sigma}^2(\mathbf{x}_i^v)}\right)^{-1}, \quad \bar{\boldsymbol{\mu}}_{ivk} = \bar{\boldsymbol{\sigma}}_{ivk}^2 \left(\frac{\boldsymbol{\mu}(\mathbf{x}_i^v)}{\boldsymbol{\sigma}^2(\mathbf{x}_i^v)} + \frac{\boldsymbol{\mu}_k(\mathbf{x}_i^v)}{\boldsymbol{\sigma}_k^2(\mathbf{x}_i^v)}\right), \tag{C.52}$$

The resulting GMM is still on the view-specific slots, the aggregation of these slots to obtain content vectors marginalises the viewpoint-level information with convex combination of parameters across all the viewpoints considered as described in cf. C.3, results in:

$$\prod_{v=1}^{|A|} \sum_{k=1}^{K} \bar{\boldsymbol{\pi}}_{ik}^v \mathcal{N}\left(\mathbf{z}; \bar{\boldsymbol{\mu}}_{ivk}, \bar{\boldsymbol{\sigma}}_{ivk}^2\right) = \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} \mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right), \tag{C.53}$$

$$\widehat{\sigma}_{ik}^2 = g(\bar{\sigma}_{ik}, \bar{\pi}_{ik}) = \sum_{v=1}^{|A|} \left(\frac{\bar{\pi}_{ik}^v}{\sum_{v=1}^{|A|} \bar{\pi}_{ik}^v} \right)^2 \bar{\sigma}_{ik}^2, \tag{C.54}$$

$$\widehat{\mu}_{ivk} = g(\bar{\mu}_{ik}, \bar{\pi}_{ik}) = \sum_{v=1}^{|A|} \frac{\bar{\pi}_{ik}^v}{\sum_{v=1}^{|A|} \bar{\pi}_{ik}^v} \bar{\mu}_{ik}, \tag{C.55}$$

Now to show that the resulting GMM is non-degenerate we need to show $\sum_{k=1}^{K} \widehat{\pi}_{ik} = 1$, for $i = 1, 2, \ldots, M$. Based on Eqn. C.47:

$$\implies \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\pi}_{ik} = \frac{1}{M|A|} \sum_{i=1}^{M} \sum_{k=1}^{K} \sum_{v=1}^{|A|} \bar{\pi}_{ik}^v \tag{C.56}$$

$$= \frac{1}{M|A|} \sum_{i=1}^{M} |A| = \frac{1}{M|A|} \cdot M|A| = 1, \tag{C.57}$$

$$\implies \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\pi}_{ik} = 1. \tag{C.58}$$

based on the above equation we can say that the scaled sum of the mixing proportions of all K components in all M GMMs when the components are aligned must equal 1, show that the resulting aggregate posterior is non-degenerate and a valid probability distribution. □

Assumption (*Weak* Injectivity) Let $f : \mathcal{Z} \to \mathcal{X}$ be a mapping between latent space and image space, where $\dim(\mathcal{Z}) \leq \dim(\mathcal{X})$. The mapping f_d is weakly injective if there exists $\mathbf{x}_0 \in \mathcal{X}$ and $\delta > 0$ such that $|f^{-1}(\{\mathbf{x}\})| = 1$, $\forall \mathbf{x} \in B(\mathbf{x}_0, \delta) \cap f(\mathcal{Z})$, and $\{\mathbf{x} \in \mathcal{X} : |f^{-1}(\{\mathbf{x}\})| = \infty\} \subseteq f(\mathcal{Z})$ has measure zero w.r.t. to the Lebesgue measure on $f(\mathcal{Z})$ (cf. Kivva et al., 2022).

Remark C.1. *In words, Assumption 5.1 says that a mapping f_d is weakly injective if: (i) in a small neighbourhood around a specific point $\mathbf{x}_0 \in \mathcal{X}$ the mapping is injective – meaning each point in this neighbourhood maps to exactly one point in the latent space $\mathcal{Z}$; and (ii) while f_d may not be globally injective, the set of points in $\mathcal{X}$ that map back to an infinite number of points in $\mathcal{Z}$ (non-injective points) is almost non-existent in terms of the Lebesgue measure on the image of $\mathcal{Z}$ under f_d.*

Theorem C.1 (Mixture of Concatenated Slots)**.** *Let f_s denote a permutation equivariant probabilistic slot attention function such that $f_s(\mathbf{z}^v, P\mathbf{s}^v) = Pf_s(\mathbf{z}^v, \mathbf{s}^v)$, where $P \in \{0, 1\}^{K \times K}$ is an arbitrary permutation matrix. Let $\mathbf{c} =$*

$(g(\mathbf{s}_1^A, .), \ldots, g(\mathbf{s}_K^A, .)) \in \mathbb{R}^{Kd}$ *be the concatenation of K individual content vectors, where each vector is an aggregate of all the slots obtained from considered viewpoints in a viewpoint-set* $A \subseteq [V]$ *(cf. Kori et al., 2024c). Due to the nature of the aggregator function, the individual content vector is Gaussian distributed within a K-component mixture:* $\mathbf{c}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall k \in \{1, \ldots K\}$*. Then,* $\mathbf{c}$ *is also GMM distributed with K! mixture components:*

$$p(\mathbf{c}) = \sum_{p=1}^{K!} \boldsymbol{\pi}_p \mathcal{N}(\mathbf{c}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p), \text{ where } \boldsymbol{\pi} \in \Delta^{K!-1}, \boldsymbol{\mu}_p \in \mathbb{R}^{Kd}, \boldsymbol{\Sigma}_p \in \mathbb{R}^{Kd \times Kd}. \tag{C.59}$$

We additionally borrow some theorems and definitions from Kivva et al., 2022 which are essential for our proofs.

Theorem C.2 (Invertibility Kivva et al., 2022)**.** *Given* $\mathbf{z} \sim \sum_{i=1}^{J} \boldsymbol{\pi}_i \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ *and* $\mathbf{z}' \sim \sum_{j=1}^{J'} \boldsymbol{\pi}'_j \mathcal{N}(\mathbf{z}'; \boldsymbol{\mu}'_j, \boldsymbol{\Sigma}'_j)$ *and* $f_d(\mathbf{z})$ *and* $\tilde{f}_d(\mathbf{z}')$ *are equally distributed. We can assume for* $\mathbf{x} \in \mathbb{R}^n$ *and* $\delta > 0$*,* f_d *is invertible on* $B(\mathbf{x}, 2\delta) \cap f_d(\mathbb{R}^m)$*. This implies that there exists* $\mathbf{x}_1 \in B(\mathbf{x}, \delta)$ *and* $\delta_1 > 0$ *such that both* f_d *and* $\tilde{f}_d$ *are invertible on* $B(\mathbf{x}_1, \delta_1) \cap f_d(\mathbb{R}^m)$*.*

THEOREM 4.3 (Affine Equivalence of aggregate content) For any subset $A \subseteq [V]$, such that $|A| > 0$, given a set of images $\mathbf{x}^A \in \mathcal{X}^A$ and a corresponding aggregate content $\mathbf{c} \in \mathcal{C}$ and a non-degenerate content posterior $q(\mathbf{c} \mid \mathbf{s}^A)$, considering two mixing function $f_d, \tilde{f}_d$ satisfying assumption 5.1, with a shared image, then $\mathbf{c}$ are identifiable up to $\sim_s$ equivalence.

Proof. Based on the results of Kori et al., 2024c we know that when $p(\mathbf{s})$ is aggregate posterior of $q(\mathbf{s} \mid \mathbf{x})$, $p(\mathbf{s})$ is identifiable up to $\sim_s$ equivalence. Additionally, based on lemma 4.2 we know that both $q(\mathbf{s} \mid \mathbf{x})$ and $q(\mathbf{c} \mid \mathbf{s})$ are a non-degenerate GMM with valid probability distribution. Using similar arguments in Kivva et al., 2022; Kori et al., 2024c we show that $p(\mathbf{c})$ and $p(\mathbf{s})$ are identifiable up to $\sim_s$ equivalence. `W.l.o.g`, given view point transformation is deterministic, we consider $\mathbf{x}^A = \mathcal{T}_{\theta^A}(\mathbf{x}^A)$.

We know that

$$p(\mathbf{s}^A) = \int q(\mathbf{s}^A_{1:K} \mid \mathbf{x}^A) p(\mathbf{x}^A) d\mathbf{x}^A \tag{C.60}$$

$$= \int \prod_{v \in A} q(\mathbf{s}^v \mid \mathbf{x}^v) p(\mathbf{x}^v) d\mathbf{x}^A \tag{C.61}$$

$$= \int \prod_{v \in A} \left(\sum_{k=1}^{K} \boldsymbol{\pi}_k^v \mathcal{N}\left(\mathbf{s}^v; \boldsymbol{\mu}_k(\mathbf{x}^v), \sigma_k^2(\mathbf{x}^v)\right) \right) p(\mathbf{x}^v) d\mathbf{x}^A \tag{C.62}$$

$$= \prod_{v \in A} \frac{1}{|\mathcal{X}|} \int \left(\sum_{k=1}^{K} \boldsymbol{\pi}_k^v \mathcal{N}\left(\mathbf{c}^v; \boldsymbol{\mu}_k(\mathbf{x}^v), \sigma_k^2(\mathbf{x}^v)\right) \right) \delta(\mathbf{x}^v - \mathbf{x}_i^v) d\mathbf{x}^A \tag{C.63}$$

$$= \prod_{v \in A} \left(\sum_{k=1}^{|\mathcal{X}|K} \frac{1}{|\mathcal{X}|} \hat{\boldsymbol{\pi}}_{ik}^v \mathcal{N}\left(\mathbf{s}^v; \hat{\boldsymbol{\mu}}_{ivk}, \tilde{\sigma}_{ivk}^2\right) \right) \tag{C.64}$$

Change of variables from $\mathbf{s}$ to $\mathbf{c}$ to get prior over random variable $\mathbf{c}$, with matching function g, results in:

$$p(\mathbf{c}_{1:K}) = \int p(\mathbf{s}^A_{1:K}) \delta\left(\mathbf{s}^A_{1:K} - g(\mathbf{s}^A_{1:K}, \boldsymbol{\pi}^A_{1:K})\right) d\mathbf{c}^A_{1:K} \tag{C.65}$$

Given the transformation g is linear, resulting us with the distribution with mean given by:

$$\mathbb{E}_{\mathbf{c}}(\mathbf{c}_{1:K}) = \mathbb{E}_{\mathbf{s}}\left(g(\mathbf{s}^A_{1:K}, \boldsymbol{\pi}_{A,1:K})\right) \tag{C.66}$$

$$= g\left(\mathbb{E}_{\mathbf{s}}(\mathbf{s}^A_{1:K}), \boldsymbol{\pi}^A_{1:K}\right) \tag{C.67}$$

$$= \sum_{v \in A} \frac{\boldsymbol{\pi}^v_{1:K}}{\sum_{v \in A} \boldsymbol{\pi}^v_{1:K}} \mathbb{E}_{\mathbf{s}}(\mathbf{s}^A_{1:K}) \tag{C.68}$$

and the covariance follows the diagonal structure as in $p(\mathbf{c})$, which can be described as follows:

$$\mathbb{V}\mathrm{ar}(\mathbf{c}_{1:K}) = \sum_{v \in A} \left(\frac{\boldsymbol{\pi}^v_{1:K}}{\sum_{v \in A} \boldsymbol{\pi}^v_{1:K}} \right)^2 \mathbb{V}\mathrm{ar}_{\mathbf{c}}(\mathbf{c}^A_{1:K}) \tag{C.69}$$

Finally, the mixture components can be expressed as:

$$\tilde{\boldsymbol{\pi}}_{1:K} = \frac{\sum_{v \in A} \boldsymbol{\pi}^v_{1:K}}{|A|} \tag{C.70}$$

With distribution parameters described in equations C.68, C.69, and C.70, we define the aggregate content distribution as GMM expressed as follows:

$$p(\mathbf{c}) = \sum_{k=1}^{|\mathcal{X}|K} \frac{1}{|\mathcal{X}|} \tilde{\boldsymbol{\pi}}_k^v \mathcal{N}\left(\mathbf{v}; \mathbb{E}(\mathbf{c})_k), \mathbb{V}\text{ar}(\mathbf{c})_k\right) \tag{C.71}$$

VALIDITY OF $p(\mathbf{c})$: The outer summation in equation C.71 can be split into two one for image samples and other for original mixing coefficients, which results in the equation:

$$p(\mathbf{c}) = \sum_{i=1}^{|\mathcal{X}|} \sum_{k=1}^{K} \frac{1}{|\mathcal{X}|} \tilde{\boldsymbol{\pi}}_{ik}^v \mathcal{N}\left(\mathbf{v}; \mathbb{E}(\mathbf{c})_{ik}), \mathbb{V}\text{ar}(\mathbf{c})_{ik}\right) \tag{C.72}$$

Based on this we can observe the each component in our GMM corresponds to particular slots for a given image in a given viewpoint, triple describing each component is:

$$\left\{\tilde{\boldsymbol{\pi}}_{ik}^v, \tilde{\boldsymbol{\mu}}_{vik}, \tilde{\sigma}_{vik}^2\right\}, \text{for} \quad v = 1, \ldots, |A| \quad i = 1, 2, \ldots, |\mathcal{X}|, \tag{C.73}$$

$$\text{and} \quad k = 1, 2, \ldots, K. \tag{C.74}$$

To verify that $p(\mathbf{c})$ is a non-degenerate mixture, we observe the following implication:

$$\sum_{i=1}^{|\mathcal{X}|} \sum_{k=1}^{K} \frac{1}{|\mathcal{X}|} \frac{\sum_{v \in A} \tilde{\boldsymbol{\pi}}_{ik}^v}{|A|} = 1, \tag{C.75}$$

$$\implies \frac{1}{|\mathcal{X}|} \frac{1}{|A|} \sum_{i=1}^{|\mathcal{X}|} \sum_{v \in A} \sum_{k=1}^{K} \boldsymbol{\pi}_{ik}^v = \frac{1}{|\mathcal{X}|} \frac{1}{|A|} |\mathcal{X}| \cdot |A| \cdot 1 = 1 \tag{C.76}$$

similar to lemma 4.2, this says that the scaled sum of the mixing proportions of all K components in all $|\mathcal{X}|$ GMMs must equal 1, proving that the associated aggregate posterior mixture $p(\mathbf{c})$ is a well-defined and non degenerate probability distribution.

INVERTIBILITY RESTRICTIONS: Given two piece-wise affine compositional functions $f_d, \tilde{f}_d : \mathcal{C} \times \mathcal{V} \to \mathcal{X}$, for a given set of views $\mathbf{v}^A$, let $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_K), \ni \mathbf{c}_k \sim \mathcal{N}(\mathbf{c}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and $\mathbf{c}' = (\mathbf{c}'_1, \ldots, \mathbf{c}'_K), \ni \mathbf{c}'_k \sim \mathcal{N}(\mathbf{c}'_k; \boldsymbol{\mu}'_k, \boldsymbol{\Sigma}'_k)$ be a pair of aggregate content representations, result of sampling a concatenated higher dimensional GMM distribution in $\mathbb{R}^{Kd}$, as shown in Theorem C.1, Kori et al., 2024c. In the case when, $f_{d\sharp}(\mathcal{C}, \{\mathbf{v}^A\})$ and $\tilde{f}_{d\sharp}(\mathcal{C}', \{\mathbf{v}^A\})$[1] are equally distributed. Now assume that there exists $\mathbf{x}^A \in \mathcal{X}$ and $\delta > 0$ such that f_d and $\tilde{f}_d$ are invertible and piecewise affine on $B(\mathbf{x}^A, \delta) \cap f_d(\mathcal{S})$, for a given set of views $\mathbf{v}^A$, which implies $\dim f_d(\mathcal{C}, \{\mathbf{v}^A\}) = |\mathcal{C}|$.

AFFINE SUBSPACE: We now restrict the space $B(\mathbf{x}^A, \delta)$ to a subspace $B(\mathbf{x}'^A, \delta')$ where $\mathbf{x}^A \in B(\mathbf{x}'^A, \delta')$ such that f_d and $\tilde{f}_d$ are now invertible and affine on $B(\mathbf{x}'^A, \delta') \cap f_d(\mathcal{C} \times \{\mathbf{v}^A\})$. With $L \subseteq \mathcal{X}^A$ be an $|\mathcal{C}|$-dimensional affine subspace (assuming $|\mathcal{X}^A| \geq |\mathcal{C}|$), such that $B(\mathbf{x}'^A, \delta') \cap f_{d\sharp}(\mathcal{C}, \{\mathbf{v}^A\}) = B(\mathbf{x}'^A, \delta') \cap L$. We also define $h_f, h_{\tilde{f}} : \mathcal{C} \to L$ to be a pair of invertible affine functions where $h^{-1}_{f\sharp}(B(\mathbf{x}'^A, \delta') \cap L) = f^{-1}_{d\sharp}(B(\mathbf{x}'^A, \delta') \cap L; \mathbf{v}^A)$ and $h^{-1}_{\tilde{f}\sharp}(B(\mathbf{x}'^A, \delta') \cap L) = \tilde{f}^{-1}_{d\sharp}(B(\mathbf{x}'^A, \delta') \cap L; \mathbf{v}^A)$. Implying $h_f(\mathbf{c})$ and $h_{\tilde{f}}(\mathbf{c}')$ are finite GMMs that coincide with $B(\mathbf{x}'^A, \delta') \cap L$ and $h_f(\mathbf{c}) \equiv h_{\tilde{f}}(\mathbf{c}')$, theorem B.2, Kivva et al., 2022. Given, $h = h^{-1}_{\tilde{f}} \circ h_f$ and $h_f(\mathbf{c})$ and $h_{\tilde{f}}(\mathbf{c}')$ then h is an affine transformation such that $h(\mathbf{c}) = \mathbf{c}'$.

$\sim_s$ EQUIVALENCE: Given Theorems B.1 and C.2, there exists a point $\mathbf{x} \in f_{d\sharp}(\mathcal{C}, \{\mathbf{v}^A\})$ that is generic with respect f_d and $\tilde{f}_d$ and invertible on $B(\mathbf{x}, \delta) \cap f_{d\sharp}(\mathcal{C}, \{\mathbf{v}^A\})$. Having established that there is an affine transformation $h(\mathbf{c}) = \mathbf{c}'$ and invertiblility of piece-wise affine functions f_d and $\tilde{f}_d$ on $B(\mathbf{x}^A, \delta) \cap f_{d\sharp}(\mathcal{C}, \{\mathbf{v}^A\})$, this implies that $\mathbf{c}$ is identifiable up to an affine transformation and permutation of $\mathbf{c}_k \in \mathbf{c}$, concluding our proof.

REMARK: Given Theorem B.2, we know that for each higher dimensional mixture component in $p(\mathbf{c})$ induces the same measure on $B(\mathbf{x}^A, \delta)$ and hence for some permutation τ we have that $(\boldsymbol{\mu}_{\pi(i)}, \boldsymbol{\Sigma}_{\pi(i)}) = (\boldsymbol{\mu}'_{\tau(\pi(i))}, \boldsymbol{\Sigma}'_{\tau(\pi(i))})$. Therefore, each mixture component $\mathbf{c}_{\pi(i)}$ is identifiable up to affine transformation, and permutation of aggregate content representations in $\mathbf{c}$. Now,

1 $f_{d\sharp}$ correspond to push forward operation, applying function f_d on all the elements of the given set.

given sampling $\mathbf{c}_k$ is equivalent to obtaining K samples from the GMM, $q(\mathbf{z})$ and concatenating, this makes $q(\mathbf{z})$ identifiable up to affine transformation, h and permutation of slot representations in $\mathbf{c}$. It now trivially follows that each of the aggregate content representation $\mathbf{c}_k \sim \mathcal{N}(\mathbf{c}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall\, k \in \{1, \ldots, K\}$ is identifiable up to affine transformation, h based on the following observed property of GMMs:

$$\sum_{k=1}^{K} \boldsymbol{\pi}_k h_\sharp \left(\mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) \sim h_\sharp\Big(\sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\mathbf{s}'_k; \boldsymbol{\mu}'_k, \boldsymbol{\Sigma}'_k)\Big), \tag{C.77}$$

□

THEOREM 4.4 (Invariance of aggregate content) For any subset $A, B \subseteq [V]$, such that $|A| > 0, |B| > 0$ and both A, B satisfy an assumption 4.1, we consider aggregate content to be invariant to viewpoints if $f_A \sim_s f_B$ for data $\mathcal{X}^A \times \mathcal{X}^B$.

Proof. Based on equation C.71, $p_A(\mathbf{s})$ and $p_B(\mathbf{s})$ can be expressed as follows:

$$p_A(\mathbf{c}) = \sum_{k=1}^{|\mathcal{X}|K} \frac{1}{|\mathcal{X}|} \frac{\sum_{v \in A} \boldsymbol{\pi}_k^v}{|A|} \mathcal{N}\left(\mathbf{c}; \sum_{v \in A} \frac{\boldsymbol{\pi}_k^v}{\sum_{v \in A} \boldsymbol{\pi}_k^v} \boldsymbol{\mu}_{vk}, \sum_{v \in A} \left(\frac{\boldsymbol{\pi}_{vk}}{\sum_{v \in A} \boldsymbol{\pi}_k^v}\right)^2 \boldsymbol{\sigma}_{vk}^2\right)$$

$$p_B(\mathbf{c}) = \sum_{k=1}^{|\mathcal{X}|K} \frac{1}{|\mathcal{X}|} \frac{\sum_{u \in B} \boldsymbol{\pi}_k^u}{|B|} \mathcal{N}\left(\mathbf{c}; \sum_{u \in B} \frac{\boldsymbol{\pi}_{uk}}{\sum_{u \in B} \boldsymbol{\pi}_k^u} \boldsymbol{\mu}_k^u, \sum_{u \in B} \left(\frac{\boldsymbol{\pi}_k^u}{\sum_{u \in B} \boldsymbol{\pi}_k^u}\right)^2 \boldsymbol{\sigma}_{uk}^2\right)$$

Given the assumption of viewpoint sufficiency 4.1 we know the objects observed in viewpoint set A are same as the object observed in set B. Following the results of Theorem 4.3, we know that both $p_A(\mathbf{s})$ and $p_B(\mathbf{s})$ are independently identifiable up to $\sim_s$ equivalence, which means f_A and f_B are invertible for a given views $\mathbf{v}^A$ and $\mathbf{v}^B$ respectively.

AFFINE MAPPING. Without loss of generality, for a given set of views $\mathbf{v}^A$, there exists some $L \subseteq \mathcal{X}^A$ be an $|\mathcal{S}|$-dimensional affine subspace, such that $B(\mathbf{x}'^A, \delta) \cap f_{A\sharp}(\mathcal{C}, \{\mathbf{v}^A\}) \cap f_{B\sharp}(\mathcal{C}, \{\mathbf{v}^A\}) = B(\mathbf{x}'^A, \delta) \cap L$. This implies their exists an affine map between $\mathbf{c} = f_A^{-1}(\mathbf{x}^A; \mathbf{v}^A)$ and $\tilde{\mathbf{c}} = f_B^{-1}(\mathbf{x}^B; \mathbf{v}^A)$. Let $h_A : \mathcal{C} \to L$ to be an invertible affine functions where $h_{A\sharp}^{-1}(B(\mathbf{x}'^A, \delta') \cap L) =$

$f_{A\sharp}^{-1}(B(\mathbf{x}'^A, \delta') \cap L; \mathbf{v}^A) = f_{B\sharp}^{-1}(B(\mathbf{x}'^B, \delta') \cap L; \mathbf{v}^A)$ resulting in $h_A(\mathbf{c}) = \mathbf{c}'$. Similarly, we can show their exists an affine map between $\tilde{\mathbf{c}} = f_A^{-1}(\mathbf{x}^A; \mathbf{v}^B)$ and $\tilde{\mathbf{c}}' = f_B^{-1}(\mathbf{x}^B; \mathbf{v}^B)$, such that $h_B(\tilde{\mathbf{c}}) = \tilde{\mathbf{c}}'$.

INVARIANCE SETUP. In the case when representations are invariant, $p_A(\mathbf{c})$ and $p_B(\mathbf{c})$ are equally distributed, which means aggregate content domain in both cases are same or similar $\mathcal{C}_A = \mathcal{C}_B$.

$$\mathbf{c}' = h(\tilde{\mathbf{c}}') \tag{C.78}$$
$$\implies h_A(\mathbf{c}) = (h \circ h_B)(\tilde{\mathbf{c}}) \tag{C.79}$$
$$\implies \mathbf{c} = (h_A^{-1} \circ h \circ h_B)(\tilde{\mathbf{c}}) \tag{C.80}$$

Given composition of affine maps is affine, we can consider the mapping $(h_A^{-1} \circ h \circ h_B)$ to be an affine, resulting in an $\sim_s$ equivalence between f_A and f_B.

□

THEOREM 4.5 (Approximate representational equivariance) For a given aggregate content $\mathbf{c}$, for any two views $\mathbf{v}, \tilde{\mathbf{v}} \sim p_A(\mathbf{v})$, resulting in respective scenes $\mathbf{x} \sim p_A(\mathbf{x} \mid \mathbf{v}, \mathbf{c})$ and $\tilde{\mathbf{x}} \sim p_A(\mathbf{x} \mid \tilde{\mathbf{v}}, \mathbf{c})$, for any homeomorphic, monotonic transformation $h_x \in \mathcal{H}_x$ such that $h_x(\mathbf{x}) = \tilde{\mathbf{x}}$, their exists another homeomorphic and monotonic transformation $h_v \in \mathcal{H}_v$ such that $\mathcal{H}_v \subseteq \mathcal{H}_x \subseteq \mathbb{R}^{\dim(\mathbf{x})}$ and $\mathbf{v} = h_v^{-1}\left(f_d^{-1}(h_x(\mathbf{x}); \mathbf{c})\right)$.

Proof. For a given view $\mathbf{v}$ and a mixing function f_d that satisfy assumptions 5.1 and is piecewise affine, from theorem 4.3 we know the latent view representations are identifiable up to $\sim_s$ equivalence for a given aggregate content vector. We know that $p(\mathbf{v})$ is expressed as GMM with a considered set of viewpoints, ideally learning each component for each viewpoint.

$$p(\mathbf{v}) = \sum_{v=1}^{|A|} \boldsymbol{\pi}^v \mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_v, \boldsymbol{\sigma}_v)$$

Following similar arguments in Theorem 4.3 and Kivva et al., 2022, we can show that for a given content representation $\mathbf{c}$ the view distribution $p(\mathbf{v})$ is

identifiable up to affine transformation. This means, for any two considered models $f_d, \tilde{f}_d$, such that $f_{d\sharp}(\mathcal{V}; \{\mathbf{c}\})$ and $\tilde{f}_{d\sharp}(\mathcal{V}; \{\mathbf{c}\})$ are equally distributed, then for any $\mathbf{x}^A \sim \mathcal{X}$ with the corresponding content representations given by $\mathbf{c}$ the views $\mathbf{v} = f_d^{-1}(\mathbf{x}^v; \mathbf{c})$, $\mathbf{v}' = \tilde{f}_d^{-1}(\mathbf{x}^v; \mathbf{c})$ are related in by an affine transformation $h(\mathbf{v}) = \mathbf{v}'$, results in:

$$\sum_{v=1}^{|A|} \boldsymbol{\pi}^v h_\sharp \left(\mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_v, \boldsymbol{\sigma}_v^2) \right) \sim h_\sharp \left(\sum_{v=1}^{|A|} \boldsymbol{\pi}^v \mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_v, \boldsymbol{\sigma}_v^2) \right), \tag{C.81}$$

Without loss of generality we can consider any function $f : \mathcal{C} \times \mathcal{V} \to \mathcal{X}$ is identifiable up to affine transformation, with this for given views $\mathbf{v}, \tilde{\mathbf{v}} \sim p(\mathbf{v})$ and for any object representations $\mathbf{c}$, the resulting scenes are sampled by distributions learned with mixing function f is given by $\mathbf{x} \sim p_f(\mathbf{x} \mid \mathbf{c}, \mathbf{v}), \tilde{\mathbf{x}} \sim p_f(\mathbf{x} \mid \mathbf{c}, \tilde{\mathbf{v}})$. As previously established for some affine transformation h,

$$h(\mathbf{v}) = f^{-1}(\tilde{\mathbf{x}}; \mathbf{c}) \implies \mathbf{v} = h^{-1} \left(f^{-1}(\tilde{\mathbf{x}}; \mathbf{c}) \right) \tag{C.82}$$

Given $h_x(\mathbf{x}) = \tilde{\mathbf{x}}$, when combined with above equation we know $\mathbf{v} = h^{-1} \left(f^{-1}(\mathbf{x}; \mathbf{c}) \right), \tilde{\mathbf{v}} = h'^{-1} \left(f^{-1}(h_x(\mathbf{x}); \mathbf{c}) \right)$, for some invertible affine transformations h and h'.

Given h_x is homeomorphic and monotonic, and f is piecewise linear, the inverse can be transferred resulting in $\tilde{\mathbf{v}} = h'^{-1} \left(\bar{h}_v (f^{-1}(\mathbf{x}; \mathbf{c})) \right)$, similarly we can also swap h'^{-1} with $\bar{h}_v$, resulting in $\tilde{\mathbf{v}} = \bar{h}_v \left(h'^{-1} \left(f^{-1}(\mathbf{x}; \mathbf{c}) \right) \right)$. Additionally combining the results from theorem 4.3 and Kivva et al., 2022, we know that $h'^{-1} \circ h$ is an affine transformation $\bar{h}$. This results in:

$$\bar{h} = h'^{-1} \circ h \tag{C.83}$$

$$\implies \tilde{\mathbf{v}} = (\bar{h}_v \circ h \circ \bar{h}) \left(f^{-1}(\mathbf{x}; \mathbf{c}) \right) \tag{C.84}$$

$$\implies \tilde{\mathbf{v}} = h_v(\mathbf{v}) \tag{C.85}$$

Given affine transformation preserves monotonicity and homeomophism, the resulting transformation $h_v \in \mathcal{H}_v$ and $h_v \in \mathcal{H}_x$, concluding the proof.

□

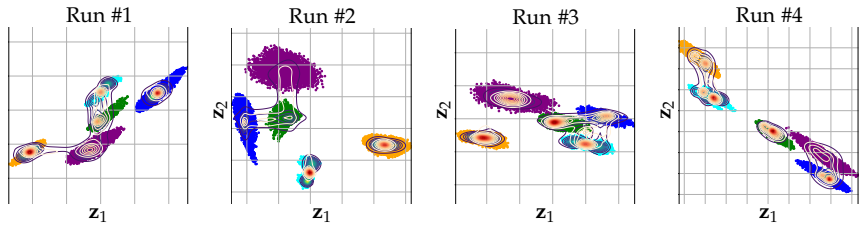


Figure C.2: Identifiability of $q(\mathbf{c})$ and $q(\mathbf{s})$. Estimated marginalised slot distribution ($q(\mathbf{s})$–blue contours) and marginalised content distribution ($q(\mathbf{c})$–orange contours, across 4 runs of VISA. This provides strong evidence of recovery of the latent space up to affine transformations, empirically verifying our claims in Thm. 4.3.

C.5 EXPERIMENTS

C.5.1 *Toy Setting*

Here, we repeat the experiments in CASE STUDY 1 with point cloud giving us two dimensional distributions, which can analysed visually. In Fig. C.2, we display the distributions of marginalized aggregate content distribution $q(\mathbf{c})$, comparing different runs that are either rotated, skewed, or mirrored with respect to each other, indicating identifiability up to affine transformation. To quantitatively measure the same, we computed SMCC and observed it to be $\mathbf{0.95} \pm 0.01$, empirically verifying our Thm. 4.3. Furthermore, to illustrate the invariance of distribution $q(\mathbf{c})$ across viewpoints (Thm. 4.4), we consider three different views. We use all possible pairs to learn $q(\mathbf{c})$ distributions as illustrated in Fig. C.3, where the distributions from second to last sub-figures are learned wrt viewpoints described by {g, r}, {r, b}, and {g, b}, respectively. Similar to our previous findings, these distributions were also found to be rotated, skewed, or mirrored relative to each other, with an observed SMCC of $\mathbf{0.87} \pm 0.11$, further confirming the claims in Thm. 4.4.

C.5.2 *Synthetic dataset results*

Here, we illustrate visual results reflecting object binding in the case of view ambiguities. Table C.2, demonstrates identifiability results on CLEVR-AUG datasets. In Fig. C.4, we demonstrate the results of VISA across three

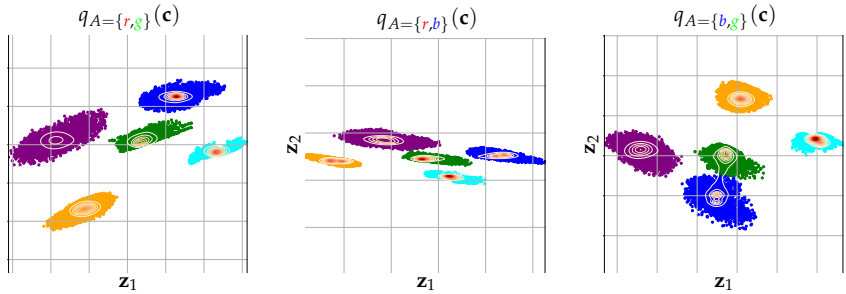


Figure C.3: Viewpoint invariance for $q(\mathbf{c})$. Estimated marginalised aggregate content distribution $q(\mathbf{c})$ when trained with different view pairs {(green, red), (red, blue), (green, blue)} are illustrated in later figures. As the resulting distributions with different datasets only vary by an affine transformation, providing strong evidence for Thm. 4.4.

Table C.2: Comparing identifiability of $q(\mathbf{s})$, $q(\mathbf{c})$, and $p(\mathbf{v})$ scores wrt existing OCL methods on CLEVR-AUG dataset.

METHOD	SMCC ↑	INV-SMCC ↑	MCC ↑
AE	$0.26 \pm .01$	-	-
SA	$0.45 \pm .05$	-	-
PSA	$0.48 \pm .03$	-	-
MulMON	$0.56 \pm .04$	$0.57 \pm .01$	-
OCLOC	$0.58 \pm .02$	$0.60 \pm .01$	$0.48 \pm .04$
VISA	$0.64 \pm .01$	$0.66 \pm .01$	$0.57 \pm .04$

different views. We additionally highlight some of the occluded regions which seem to be better captured by our proposed model, which can be attributed to the multi-view setting and the `sigmoid` mask.

Additionally, we also illustrate the results from CLEVR-MV dataset in figure C.5.

C.5.3 *Influence of Number of Views*

Here, we demonstrate the influence of the number of views on the overall identifiability of object-centric representations. Similar to Fig. 4.1(c), in Fig. C.6, we observe an increasing number of views increase overall results.

C.5.4 MVMOVI *Results*

Here, we discuss the results obtained from the proposed dataset. To reiterate, MVMOVI-C is a variant where fixed camera positions are maintained for all

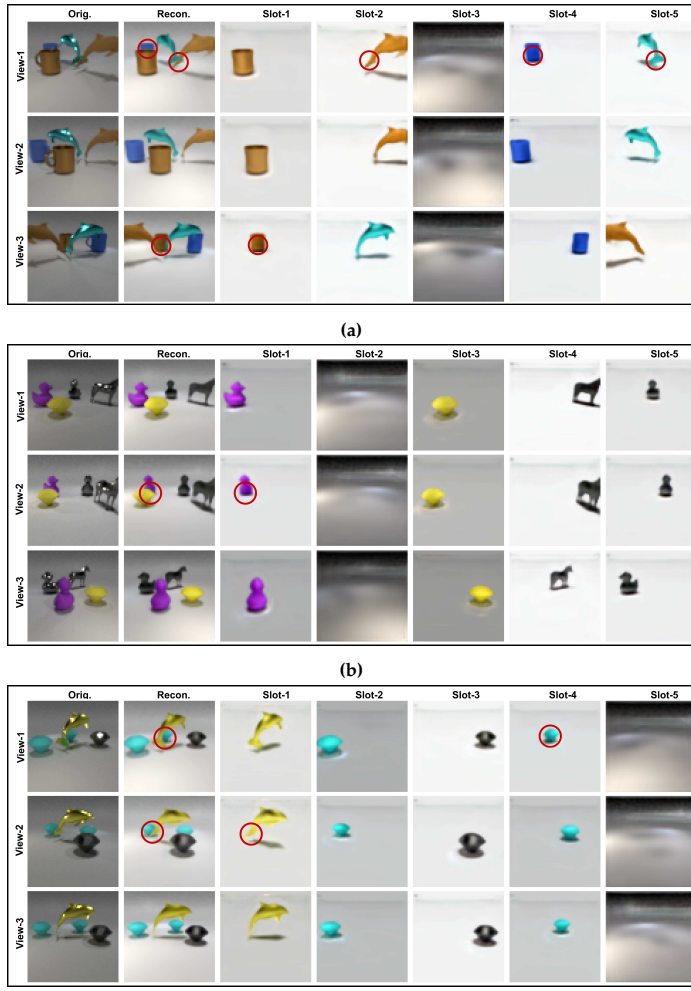


Figure C.4: Visual illustrations of benchmark results on CLEVR-AUG dataset.

viewpoints across all scenes in the dataset. This setup helps assign a fixed type of viewpoint conditioning for all images captured from a particular camera.

The detection and binding quality of different models are illustrated in Table 4.5. From these results, we can clearly observe that while the model demonstrates similar binding capabilities, the identifiability of object representations is improved in our proposed model. This suggests that

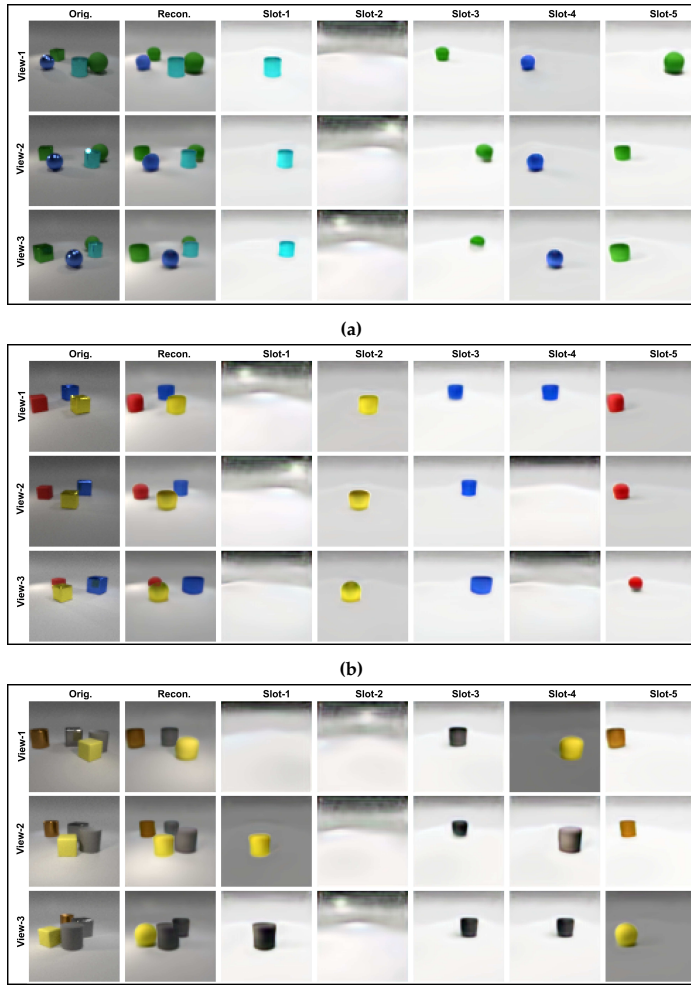


Figure C.5: Visual illustrations of benchmark results on CLEVR-MV dataset.

the use of fixed camera positions in MVMOVI-C enhances the consistency and quality of object representation learning, leading to better detection performance across different viewpoints.

Figure C.7 & C.8 showcases the object discovery capabilities of the VISA. In the iteration of the MVMOVI-D dataset, we vary the camera position for each scene, making the dataset more dynamic and allowing for the potential violation of assumption 4.1 in certain cases. Table C.3 presents the

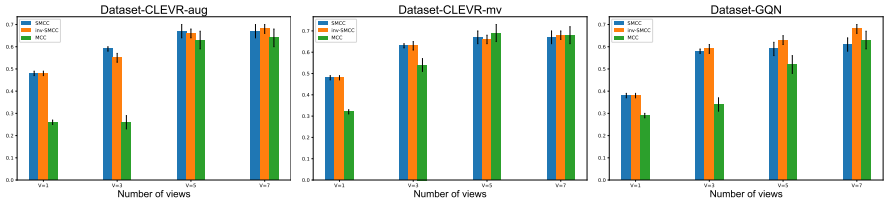


Figure C.6: Influence of Number of viewpoints on identifiability for synthetic datasets.

binding and identifiability results for both in-domain and out-of-domain data, following a similar analysis as in Table 4.5. We observe consistent trends and behaviours, suggesting that the impact of the assumption is minimal. A more detailed analysis of the assumption's effects will be left for future work.

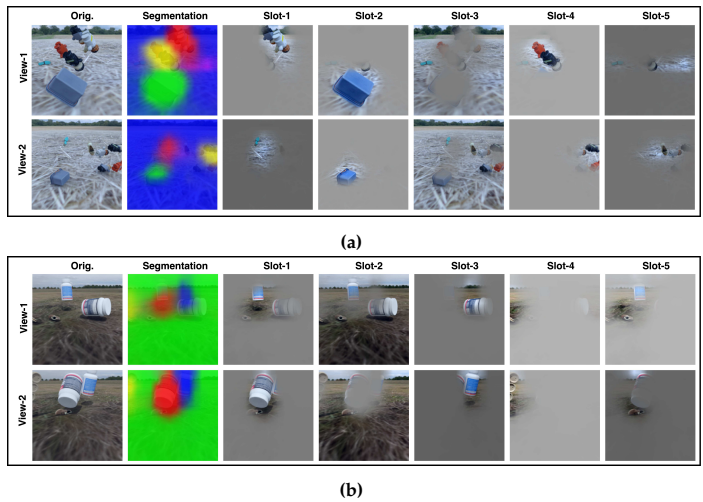
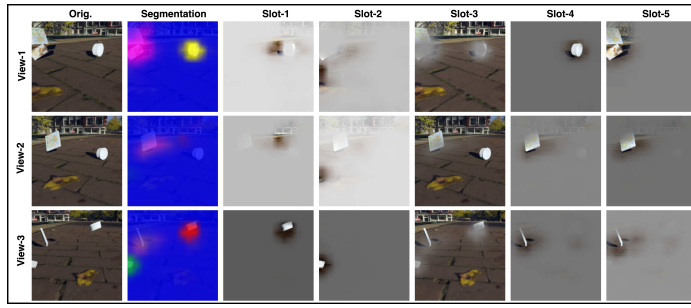


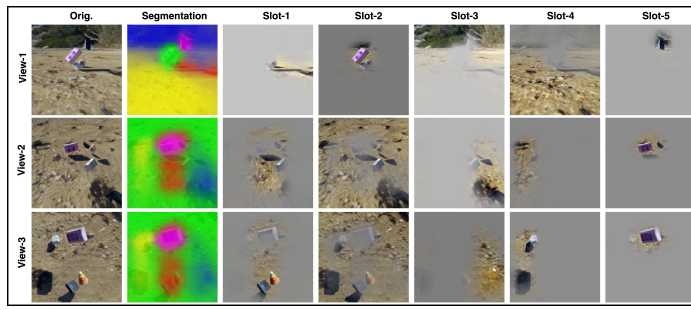
Figure C.7: Visual illustrations of benchmark results on MVMOVI-C dataset with 2 views.

C.5.5 *View warm-up*

Given the stochasticity during the initial phase of training, to facilitate meaningful representation in content aggregator function, we consider view warm-up strategy. For the initial 100,000 iterations, we randomly use the



(a)



(b)

Figure C.8: Visual illustrations of benchmark results on MVMOVI-C dataset with 3 views.

view-specific slots for reconstruction instead of invariant content with a probability of 0.5.

This primary makes sure the feature extractor extracts meaningful representations before aggregation, which helps to stabilize the training process and allows the model to effectively bind and integrate information from different perspectives in later stages of training.

C.5.6 *Hyperparameters*

In Table C.4, we detail all the hyper-parameters used in our experiments. In the case of benchmark experiments, we use trainable CNN encoder as used in Locatello et al., 2020b; Kori et al., 2024b, while in the case of proposed MVMOVI datasets we use DINO (Caron et al., 2021) encoder to extract image features and change our objective to reconstruct these features rather than the original image as proposed in Seitzer et al., 2022. For most of

Table C.3: Identifiability and generalisability analysis on MV-MoViD dataset.

Method	Indomain Analysis				Out of domain			
	mBO ↑	SMCC ↑	inv-SMCC ↑	MCC ↑	mBO ↑	SMCC ↑	inv-SMCC ↑	MCC ↑
SA-MLP	0.24 ± 0.031	0.44 ± 0.005	-	-	0.24 ± 0.097	0.45 ± 0.008	-	-
PSA-MLP	0.26 ± 0.022	0.44 ± 0.006	-	-	0.25 ± 0.012	0.42 ± 0.006	-	-
VISA-MLP	0.24 ± 0.099	0.48 ± 0.009	0.46 ± 0.054	0.57 ± 0.021	0.25 ± 0.011	0.48 ± 0.006	0.51 ± 0.021	0.55 ± 0.021
SA-Transformer	0.34 ± 0.017	0.40 ± 0.041	-	-	0.34 ± 0.066	0.38 ± 0.031	-	-
PSA-Transformer	0.37 ± 0.021	0.38 ± 0.007	-	-	0.36 ± 0.024	0.36 ± 0.016	-	-
VISA-Transformer	0.39 ± 0.016	0.46 ± 0.001	0.48 ± 0.001	0.54 ± 0.032	0.37 ± 0.051	0.46 ± 0.022	0.45 ± 0.010	0.54 ± 0.029

hyperparameters we use the values suggested by Locatello et al., 2020b; Seitzer et al., 2022, based on their ablation results.

C.5.7 *Computational Resources*

We run all our experiments on a cluster with a Nvidia NVIDIA L40 48GB GPU cards. Our training usually takes between eight hours to a couple of days, depending on the model and the dataset. It is to be noted that speed might differ slightly with respect to the considered system and the background processes.

Table C.4: Experimental details w.r.t datasets

Datasets(↓)	Parameters	Values
CLEVR, GSO	No. Layers	4
	No. Views	10 (GSO: 8)
	No. Slots	7
	Training Epochs	5000
	Batch Size	32
	Optimizer	Adam
	Learning Rate	0.0002
	Initial Slot $\boldsymbol{\mu}$	$\mathcal{N}(0, 1)$
	Initial Slot $\boldsymbol{\sigma}$	$\mathbb{I}$
	Warmup Steps	10000
	Decoder	Spatial Broadcasting-CNN
	$\mathbf{x}-$ likelihood	$\mathcal{N}(\boldsymbol{\mu}_x, \sigma_x^2\mathbb{I})$
GQN	No. Layers	4
	No. Views	10
	No. Slots	4
	Training Epochs	5000
	Batch Size	64
	Optimizer	Adam
	Learning Rate	0.0002
	Initial Slot $\boldsymbol{\mu}$	$\mathcal{N}(0, 1)$
	Initial Slot $\boldsymbol{\sigma}$	$\mathbb{I}$
	Warmup Steps	10000
	Decoder	Spatial Broadcasting-CNN
	$\mathbf{x}-$ likelihood	$\mathcal{N}(\boldsymbol{\mu}_x, \sigma_x^2\mathbb{I})$
mvMoVi-C, mvMoVi-D	No. Layers	4
	No. Views	5
	No. Slots	7
	Training Epochs	560
	Batch Size	64
	Optimizer	AdamW
	Learning Rate	0.0002
	Initial Slot $\boldsymbol{\mu}$	$\mathcal{N}(0, 1)$
	Initial Slot $\boldsymbol{\sigma}$	$\mathbb{I}$
	Warmup Steps	10000
	Pretrained Encoder	dino_vitb16
	Decoder	MLP, Transformer
	$\mathbf{x}-$ likelihood	$\mathcal{N}(\boldsymbol{\mu}_x, \mathbb{I})$

COLOPHON

This thesis was typeset using `classicthesis`, developed by André Miede, with adaptations from Tino Wagner and Charles Jones. The style was inspired by Robert Bringhurst's book "*The Elements of Typographic Style*". The LaTeX source for `classicthesis` is available for free under a GNU GPLv2 license. Official releases are hosted at `https://ctan.org/pkg/classicthesis`.

André Miede requests that grateful users of `classicthesis` send a postcard. A collection of postcards is featured at `http://postcards.miede.de/`.