

**Consistent and Coherent Relational
Representation Learning**

Harald Strömfelt

20.11.2023

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London and
the Diploma of Imperial College London

Declaration of Originality

I, Harald Strömfelt, hereby declare that the work presented in this thesis is entirely my own. Collaborators and the work of others have each been clearly acknowledged and referenced.

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Licence (CC BY NC-SA).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that; you credit the author, do not use it for commercial purposes and share any derivative works under the same licence.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence 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 licence or permitted under UK Copyright Law.

Abstract

Achieving human-like general intelligence is a long-standing goal of Artificial Intelligence (AI) research. A hallmark of this intelligence is the freeing of learned concepts from the particular experiences that led to their acquisition. In contrast, Deep Learning (DL) methods are typically unable to generalise in this manner. Inspired by this capacity, this thesis proposes and formalises a new approach to learning at the intersection of symbolic AI and representation learning. The contributions are as follows.

Firstly, in the context of Lifelong Reinforcement Learning, trade-offs are explored between traditional deep Q-learning and architectures that explicitly incorporate relational abstractions. Results suggest that including such inductive biases can improve the lifelong learning performance of an agent.

Next, disentangled representation learners, namely Variational AutoEncoders (VAEs), are combined with relation-decoders, towards obtaining stronger representations for downstream learning. The second contribution is to show that vanilla VAEs are susceptible to dataset ambiguities and may learn representations that fail to capture abstract properties of certain objects, negatively impacting downstream performance. Including an explicit relational inductive bias, however, mitigates this issue.

Whilst the above approaches resemble the learning of abstract relational concepts, their semantics are unclear. The third contribution is the introduction of a formal framework for relation learning, taking inspiration from formal semantics in symbolic AI. In particular, formal definitions and conditions for the consistency and coherence of relational systems are presented, underpinned by soft-structures, a DL counterpart to the model-theoretic structure.

As the final contribution, this thesis introduces a novel consistent and coherent relation-decoder model, Dynamic Comparator (DC), and a Partial Relation Transfer experimental setting that exposes the semantics of learned relations. Using a proposed practical consistency and coherence measurement, results show that DC is able to learn a more generalisable set of relations and suggest a strong correlation between soft-structure coherence and generalisation performance.

Acknowledgements

Behind any PhD thesis is a large support network. I do not see how it could be any other way. As with many endeavours, there are many ups and downs, many moments to celebrate, but equally many that require you to pick yourself back up and keep going. It helps having people around you that can help you do that and I have been lucky to have many such people.

Firstly, I want to sincerely thank my supervisors, Alessandra Russo, Luke Dickens and Artur d'Avila Garcez. Foremost for their continuous support and belief. Since my very first day with them, each gave serious consideration to my ideas, even though at the time I knew relatively little. This gave me a lot of confidence and I am grateful for that. Under their supervision, setbacks did not feel like setbacks for long. Their leadership meant that the way forward quickly became clear and their support meant that I was ready to go again in no time.

Alessandra seems to have a sixth sense for when to apply pressure, or give support, or anything in between. I found that on the days where I was less jovial, she was able to provide the energy and enthusiasm to keep me going. On the days where I might be getting too ahead of myself, she was able to provide the focus and direction to keep me on track. I can say much the same for my other supervisors, but I think Alessandra has a special talent for this. As for Luke, I like to think he was my commander and in the thick of it with me. He has pushed me to be more thorough, but without straying from a place of enthusiasm and curiosity. I have thoroughly enjoyed our discussions and I have learned so much from him. Artur has been pivotal to many of my achievements. He has helped me realise where my work fits in the bigger picture and has been particularly encouraging in pushing me to submit my work. I am extremely grateful for this.

Under each of their supervision, I feel I have grown both as a researcher and as a person. Thank you all. If anyone is reading this and is considering working under their supervision. Do it. They will each give you the space to be curious and follow your passion, all the while keeping you focused and moving forwards, sometimes without you even realising it!

Next, my parents, Zeynep and Otto, and my brother, Gustav. My parents. I would not be here without them (quite literally). I am firstly grateful to both for showing me love and support throughout my life, which they continue to do to this day. To my mum, thank you for giving me a safe space to be myself and for always being there for me, but also knowing when to tell me to snap out of a bad mood. To my dad, thank you for giving me the sense that I can achieve whatever I want, and that I am in charge of my own life. This has very much led me to the PhD, and has been critical for me to push until the finish line. To my brother, I know I have a partner in life forever.

I would also like to thank Nuri, Alex, and the rest of the SPIKE group. Each has been helpful in their own way, and I am grateful to have been part of such a supportive and fun group.

Lastly, Nina Quach. Put bluntly, without her I would not have made it this far. Nina embodies all of the above to me, and more. She is the star to the solar system that is my life, giving me the energy to push forward and the gravity to keep me from getting lost on the way. You know you have found someone special, when each time you look around, after toiling away, life continues to be better than you could have ever imagined.

To Nina Quach

Contents

1	Introduction	25
1.1	Relational representations	27
1.2	Thesis structure and contributions	28
1.3	Publications	29
2	Background	31
2.1	Computational Logic	31
2.1.1	Propositional Logic	33
2.1.2	First-Order Logic	35
2.2	Deep Learning	39
2.2.1	Artificial Neural Networks	39
2.2.2	Common architectures	42
2.2.3	Optimisation	45
2.2.4	Forms of supervision	48
2.3	Types of learning	48
2.3.1	Transfer Learning	49
2.3.2	Multi-Task Learning	49
2.3.3	Reinforcement Learning	50
2.4	Representation Learning	53
2.4.1	A theoretical definition of disentanglement	54
2.4.2	Variational AutoEncoders	57
3	Deep relational representations as transferable knowledge	59
3.1	Relational abstractions towards reusable knowledge	60
3.2	End-to-end Relational Reinforcement Learning	64
3.3	Experimental setup	68
3.4	Results	70

3.5	Discussion	73
4	Improving representation learning with relation-decoders	75
4.1	Disentanglement and relational representations	76
4.2	A straightforward relational representation learning architecture	78
4.3	Experiments	80
4.3.1	Relation-decoding transfer and disentangled representations	80
4.3.2	Abstract visual reasoning over non-visual semantics	83
4.4	Results	85
4.5	Discussion	93
5	Defining coherence and consistency of Soft-Structures	97
5.1	From logical structures to soft-structures	98
5.1.1	Defining consistency of soft-substructures	101
5.1.2	Defining coherence of soft-substructures	107
5.2	Exploring when we can obtain coherence	109
5.3	Coherence in real-world domains	112
5.4	Disentangled soft-structures	117
5.4.1	A disentangled view of relations and theories	119
6	Learning consistent and coherent disentangled soft-structures	122
6.1	Designing a consistent and coherent relation-decoder for disentangled soft-structures	123
6.1.1	Design principles	123
6.1.2	Dynamic Comparator	126
6.1.3	Learning to approximate disentangled families of relations	133
6.2	Partial Relation Transfer	135
6.3	Measuring consistency and coherence in practice	140
6.4	Results	142
6.5	Discussion	148
7	Related Work	151
8	Conclusion	159
8.1	Future work	162

Bibliography	166
Appendix	187
A Additional experimentation details	187
B Architecture descriptions	192
C Specification for theory of ordinality	195
D Expanded consistency loss derivation	198

List of Figures

2.1	Basic Artificial Neural Network and common activation functions	40
2.2	Illustration of a convolutional layer and a kernel convolution	43
2.3	Illustration of the agent-environment interaction modelled as an MDP	50
2.4	Illustration of a standard AE and VAE architecture	57
3.1	Agent processing pipeline for Example 3.1	63
3.2	Illustration of π used in the EW-CNN- ϕ architecture	66
3.3	Breakdown of end-to-end Deep Reinforcement Learning architectures	67
3.4	Example dSprites data for each EF that has more than one attribute value . .	68
3.5	Evaluation L^1 reward classification loss and rank reward for each DRL architecture	71
3.6	Illustration of evaluation performance for a collection of training runs	72
4.1	Combined Variational AutoEncoder and relation-decoder architecture	78
4.2	Example of an arithmetic Raven’s Progressive Matrices instance	83
4.3	Mutual Information Gap score distributions across λ settings	86
4.4	Relationship between relation-decoding prediction accuracy and MIG scores . .	87
4.5	σ^+ relation-decoding performances across λ settings	88
4.6	σ^- relation-decoding performances across λ settings	88
4.7	Comparison between source and downstream relation-decoding performances . .	90
4.8	Target phase RPM task error rates for different weak relational supervision settings	91
4.9	Mutual information statistics for each latent dimension	92
4.10	TSNE plots of the MNIST dataset with/without weak relational supervision . . .	93
5.1	Plotting $\Gamma_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ for different ground atom beliefs	107
6.1	Illustration of possible choices for the $\phi^\dagger$ DC mode	129
6.2	Illustration of possible choices for the $\phi^\ddagger$ DC mode	130
6.3	Depiction of a set of ordinal relation DC relation-decoders	132

6.4	Masks for a set of DC relation-decoders trained on the dSprites dataset	134
6.5	dSprites EF versus $\mathcal{Z}$ dimension activations for DC relation-decoders	135
6.6	Depiction of the architecture used for PRT tasks	136
6.7	PRT accuracy analysis across models, relations, and β settings	145
6.8	Consistency-Across loss breakdown across models and β settings	146
6.9	Consistency-Individual loss breakdown across models and β settings	147
8.1	Illustration of proposed relation-decoder coupling to avoid catastrophic forgetting	166
A.1	Depiction of DC flexible fitting requirement for isSuccessor	190
A.2	Example images from the BlockStacks dataset	191

List of Tables

2.1	Overview of the Propositional Logic connectives	33
2.2	Truth-tables for the Propositional Logic connectives	34
2.3	Overview of the syntactical elements of First-Order Logic	36
2.4	Taxonomy of supervision types	48
3.1	Breakdown of the dSprites dataset Explanatory Factors	69
6.1	Characteristic sentences that a binary relation may adhere to	123
6.2	Spearman rank coefficients between consistency loss and PRT accuracy	143
6.3	Coherence proxy comparison with respect to source and target data-embeddings	147
A.1	Optimal Hyperparameters for Chapter 3 architectures	188
B.1	RL encoder architecture parameters for the dSprites dataset	193
B.2	CNN (baseline) and EW-CNN policy architecture parameters	194
B.3	β -TCVAE encoder and decoder architecture parameters, for the dsprites dataset	194
B.4	β -VAE encoder and decoder architecture parameters, for the MNIST dataset . .	195
B.5	β -VAE encoder and decoder architecture parameters, for the BlockStacks dataset	195
B.6	Wild Relation Network model parameters	196
C.1	Characteristic properties of the ordinal relations	197

Acronyms

Acronym	Description	Pages
AE	AutoEncoder	44, 45, 47, 48, 57, 74, 153
AGI	Artificial General Intelligence	25, 26, 166
AI	Artificial Intelligence	5, 25, 26, 27, 50, 53, 54, 59, 60, 166
ANN	Artificial Neural Network	13, 39, 40, 41, 42, 44, 45, 46, 47, 49, 53, 85, 152
CNN	Convolutional Neural Network	42, 43, 44, 53, 55, 65, 66, 67, 74, 153
DC	Dynamic Comparator	5, 29, 122, 123, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 139, 144, 145, 146, 147, 149, 150, 155, 157, 158, 161, 164, 165, 190
DL	Deep Learning	5, 25, 29, 31, 39, 40, 50, 53, 64, 69, 101, 135, 148, 152, 159, 160, 163, 164, 166
DRL	Deep Reinforcement Learning	13, 50, 53, 60, 64, 67, 151, 153, 154

Acronym	Description	Pages
EF	Explanatory Factor	53, 54, 55, 59, 61, 62, 63, 67, 68, 69, 72, 74, 75, 76, 77, 78, 80, 81, 82, 87, 89, 94, 95, 96, 99, 113, 118, 119, 120, 121, 123, 125, 126, 133, 134, 135, 138, 149, 154, 157, 158, 160, 162, 164, 165, 166
ELBO	Evidence Lower BOund	57, 78, 79, 81, 83, 157, 189
FOL	First-Order Logic	15, 28, 29, 33, 35, 36, 37, 38, 59, 60, 61, 62, 63, 64, 66, 67, 68, 70, 72, 73, 74, 75, 76, 77, 78, 79, 80, 88, 93, 94, 96, 97, 98, 99, 101, 122, 123, 124, 135, 137, 142, 148, 151, 152, 154, 155, 156, 159, 160, 161, 162, 163, 165
GAN	Generative Adversarial Network	76, 79
GNN	Graph Neural Network	154, 163
HolE	Holographic Embeddings relation-decoder	139, 144, 145, 146, 147, 148, 193
KGE	Knowledge Graph Embedding	154
KL	Kullback-Leibler	57, 82
LFM	Latent Feature/Factor Model	154, 155, 156
LLM	Large Language Model	26, 166
LRL	Lifelong Reinforcement Learning	5, 62, 68, 70, 75, 76, 162
LTN	Logic Tensor Network	153, 154, 155, 162, 164
MDP	Markov Decision Process	50, 51, 61
MI	Mutual Information	91, 92, 95, 158
MIG	Mutual Information Gap	13, 82, 86, 87, 91, 92, 93, 151, 158, 188
ML	Machine Learning	39, 45, 97, 113

Acronym	Description	Pages
MLP	Multi-Layer Perceptron	39, 40, 42, 44, 65, 66
MTL	Multi-Task Learning	48, 49, 50, 79
NN	Basic neural network relation-decoder	139, 144, 145, 146, 147, 149, 150, 193
NTN ⁺	Neural Tensor Network ⁺	66, 79, 89, 94, 134, 139, 144, 145, 146, 147, 187, 188, 192
PL	Propositional Logic	15, 33, 34, 35, 36, 38
PRT	Partial Relation Transfer	5, 29, 122, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 148, 149, 161, 164, 190
ReLU	Rectified Linear Unit	41, 65, 193
RL	Reinforcement Learning	28, 46, 49, 50, 51, 52, 53, 60, 69, 73, 151, 152, 153, 159, 160, 193
RN	Relation Network	84, 85, 156
RPM	Raven’s Progressive Matrices	13, 83, 84, 85, 90, 91, 92, 93, 95, 189
RRL	Relational Reinforcement Learning	64, 97, 151
TL	Transfer Learning	48, 49, 50, 136, 137
TransR	Translational embedding relation-decoder with relation-specific parameters	139, 144, 145, 146, 147, 148, 192
VAE	Variational AutoEncoder	5, 13, 45, 57, 58, 76, 77, 78, 79, 80, 81, 83, 99, 101, 125, 134, 135, 137, 138, 139, 149, 151, 156, 157, 160, 162, 163, 190, 191, 193
WReN	Wild Relation Network	15, 84, 85, 91, 92, 93, 95, 156, 189, 194, 196

Notations

This thesis follows the notation standard of the “Deep Learning” textbook (Goodfellow et al., 2016). Some notations overlap and will be clarified within the text as required.

Numbers and Arrays

a	A scalar (integer or real)
$\mathbf{a}$	A vector
$\mathbf{A}$	A matrix
$\mathbf{A}$	A tensor
$\mathbf{I}_n$	Identity matrix with n rows and n columns
$\mathbf{I}$	Identity matrix with dimensionality implied by context
A	A random variable which may be scalar-, vector- or matrix-valued
$[a, b]$	Vector with scalar element a and b
$[\mathbf{a} \mid \mathbf{b}]$	Matrix given by column-wise concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$
$[\mathbf{a} ; \mathbf{b}]$	Vector given by row-wise concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$
$[\mathbf{A} \mid \mathbf{B}]$	Matrix given by column-wise concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$
$[\mathbf{A} ; \mathbf{B}]$	Matrix given by row-wise concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$

Sets and Graphs

$\mathcal{A}$	A set
$a \in \mathcal{A}$	An element of a set
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of positive real numbers
$\mathbb{Z}$	The set of integers
$\mathbb{Z}_0^+$	The set of non-negative integers
$\mathbb{B}$	The Boolean set $\{\top, \perp\}$
$\{0, 1\}$	The set containing 0 and 1
$\{0, 1, \dots, n\}$	The set of all integers between 0 and n
$[a, b]$	The real interval including a and b
$(a, b]$	The real interval excluding a but including b
(a, b)	The open interval between a and b
$[a, b]$	The closed interval between a and b

Indexing

a_i	Element i of vector $\mathbf{a}$, with indexing starting at 1
a_{-i}	All elements of vector $\mathbf{a}$ except for element i
A_{ij}	Element i, j of matrix $\mathbf{A}$
$\mathbf{A}_{i:}$	Row i of matrix $\mathbf{A}$
$\mathbf{A}_{:,i}$	Column i of matrix $\mathbf{A}$
A_{ijk}	Element (i, j, k) of a 3-D tensor $\mathbf{A}$
$\mathbf{A}_{::i}$	2-D slice of a 3-D tensor

Calculus

$\frac{dy}{dx}$	Derivative of y with respect to x
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\mathbf{x}} y$	Gradient of y with respect to $\mathbf{x}$
$\int f(\mathbf{x}) d\mathbf{x}$	Definite integral over the entire domain of $\mathbf{x}$

$\int_{\mathcal{S}} f(\mathbf{x}) d\mathbf{x}$	Definite integral with respect to $\mathbf{x}$ over the set $\mathcal{S}$
--	---

Probability and Information Theory

$p_A(a)$	A probability distribution over a continuous variable, or discrete variable, or over a variable whose type has not been specified
$a \sim p_A(A)$	Random variable a has distribution P
$\mathbb{E}_{x \sim p_X(X)}[f(x)]$ or $\mathbb{E}[f(x)]$	Expectation of $f(x)$ with respect to $P(x)$
$\text{Var}(f(x))$	Variance of $f(x)$ under $P(x)$
$\text{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $p_X(X)$
$H(X)$	Shannon entropy of the random variable X
$H(X, Y)$	Cross entropy between random variables X and Y
$I(A; B)$	Mutual information between random variables A and B
$D_{\text{KL}}(P \ Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Logic

$p, p_1, q, q' \dots$	Propositions
$A, A_1, B, B' \dots$	A formulae in Propositional Logic, or a variable in First-Order Logic, made clear from context
$\neg$	Negation
$\wedge$	Conjunction
$\vee$	Disjunction
$\rightarrow$	First-Order Logic implication
$\leftrightarrow$	First-Order Logic equivalence
$\forall$	First-Order Logic universal quantification
$\exists$	First-Order Logic existential quantification
relation	A relation symbol, denoted by the font
function	A function symbol, denoted by the font
constant	A constant symbol, denoted by the font

τ	A First-Order Logic sentence
$\square$	A First-Order Logic formula
$\mathcal{T}$	A First-Order Logic theory
$\mathcal{L}^{\text{FOL}}$	A First-Order Logic language
σ	A First-Order Logic signature
$\mathcal{S}_\sigma$	A model-theoretic structure, with domain $\mathcal{S}$ and signature σ
$\mathcal{M}_\mathcal{S}^\mathcal{T}$	A the set of all models of a theory $\mathcal{T}$ and domain $\mathcal{S}$

Functions

$f : \mathcal{A} \rightarrow \mathcal{B}$	The function f with domain $\mathcal{A}$ and range $\mathcal{B}$
$f \circ g$	Composition of the functions f and g
$f(\mathbf{x}; \boldsymbol{\theta})$	A function of $\mathbf{x}$ parametrized by $\boldsymbol{\theta}$ (sometimes the argument $\boldsymbol{\theta}$ is omitted to lighten notation)
$\ln x$	Natural logarithm of x
$\mathbf{1}_{x'}(x)$	Indicator function giving 1 if the $x = x'$ and 0 otherwise
$\sigma(x)$	Logistic Sigmoid function
$\tanh(x)$	Hyperbolic Tangent function
$\text{ReLU}(x)$	Rectified Linear Unit function
$\text{softmax}(\mathbf{x})$	Softmax function
$\ \mathbf{x}\ _p$	L^p norm of $\mathbf{x}$
$\ \mathbf{x}\ $	L^2 norm of $\mathbf{x}$
$\dim(\mathcal{V})$	gives the number of dimensions of a (context assumed) vector space $\mathcal{V}$
$\text{ar}(b)$	gives the arity (number of arguments) of a function or relation symbol b

Chapter 1

Introduction

Human-level intelligence has been a central guiding principle since the very earliest days of Artificial Intelligence (AI) research (McCarthy, 1959; McCarthy et al., 2006), and remains so to this day (Goyal and Bengio, 2022; Bubeck et al., 2023). Since the Dartmouth Summer Research Project on Artificial Intelligence in 1956 (McCarthy et al., 2006), arguably the inception of AI as a field of study, substantial progress has been made (Silver et al., 2017; Bubeck et al., 2023; OpenAI, 2023), to a point where now AI systems can match or even outperform humans in a variety of isolated tasks (Silver et al., 2017; Mnih et al., 2015; He et al., 2015) and the possibility of superintelligent Artificial General Intelligence (AGI) is being seriously considered (Bostrom, 2014; Bubeck et al., 2023). Deep Learning (DL) has been a central driving force in this intense activity towards progress (LeCun et al., 2015), with the ability to learn from data and the explosion of available data and available compute being a key component of its success.

An important capacity of human intelligence is our ability to acquire, reuse, and update knowledge throughout life, adapting and employing it as we face new challenges Barnett and Ceci (2002); Piaget (2005); Lake et al. (2017). Despite the remarkable recent successes of DL, achieving parity in these abilities, however, remains a challenge of major interest for researchers in AI Thrun and Mitchell (1995); Thrun (1995); Parisi et al. (2019); Chandar et al. (2022); DARPA (2017); Eaton et al. (2022), with serious hindrances, such as catastrophic forgetting, prohibiting AI systems from building on their knowledge (McCloskey and Cohen, 1989; French, 1993). Instead, the main progress has been in the form of isolated learning, where DL architectures are optimised from scratch against an objective starting from a randomly

initialised starting state, often using standalone datasets (Chen and Liu, 2018; Liu, 2017).

In addition, and perhaps importantly to its transferable value, humans appear to seek consistency, where our experiences should stay aligned with what our knowledge tells us to expect (Piaget, 2005). Moreover, we are able to connect knowledge through logical association to form increasingly complex conceptual understanding. Once established, we can then make sense of new experiences using these concepts, recognizing when they are consistent with our existing knowledge, and when they are not. For instance, in learning that, *e.g.* “A is larger than B” or “B is smaller than A”, we obtain a conceptual understanding of ordinality, in which its characteristic relations are *consistent* with one another: “If A is larger than B, then B is smaller than A”. Equipped with this *concept*, we can then coherently apply it to new objects and new ordinal properties: “D is louder than C, so C is quieter than D”. Here again state-of-the-art AI, which exhibit impressive generalisation capacity in some contexts (Brown et al., 2020; Kojima et al., 2022), seem to represent their knowledge in a manner that does not induce logical consistency where it should (Wu et al., 2023; Nye et al., 2021; Yang et al., 2023; Berglund et al., 2023). For instance, state-of-the-art Large Language Models (LLMs), exhibiting AGI-like behaviour in some regards (Bubeck et al., 2023) and even able to play chess, fail to adapt when the chessboard configuration is restructured, resulting in near random-level performance (Wu et al., 2023). Moreover, LLMs have been found to suffer from the so-called *Reversal Curse*, for example successfully answering questions such as “Who is Tom Cruise’s mother?” [A: Mary Lee Pfeiffer] and subsequently failing to answer “Who is Mary Lee Pfeiffer’s son?” highlighting a possible failure to perform logical deduction (Berglund et al., 2023). Together this suggests that although significant progress is being made, equally significant failure modes need to be addressed with regards to the logical consistency and coherence of conceptual understanding.

Towards bridging the gap between AI and human-like intelligence, the central questions to this thesis are: what knowledge can be useful for an AI to learn and what characteristics should it have? In endeavouring to make progress on these questions, the guiding intuitions for this thesis are as follows. First, common ground between a human and a human-like AI is that both operate in the real-world and so the tasks that each can be reasonably expected to perform can be scoped (Bengio et al., 2013; Bengio and LeCun, 2007). Bengio and LeCun (2007) refer to this common ground as the *AI-set*, including tasks such as object perception, language understanding, and motor control. In existing within a fixed world, which exhibits certain

regularities, both humans and AI can benefit from *representing* the world in a way that exposes these regularities (Bengio et al., 2013). Second, desirable properties for knowledge are that it is *modular* and *compositional*. The former, inspired by Piagetian “schemata” (Piaget, 2005), allows for selective reuse and training, where certain components can be updated or added without needing to change others (Mendez and Eaton, 2021). The latter allows for the formation of new knowledge through reuse of previous knowledge, benefiting from combinatorial explosion, an idea best exemplified by language in which a finite set of words can be combined to express an infinite amount of possible sentences (Platts, 1979). Third, a central property of human knowledge is that it appears to be *relational* (Halford et al., 2010), in which our knowledge is expressed in terms of entities and the properties/relations that they have/share. This is to say that we can compare entities based on their intrinsic properties, and form abstract associations between them. Putting these intuitions together, this thesis will focus on learning consistent and coherent *relational representations*.

1.1 Relational representations

The remainder of this thesis will often refer to a *relational representation*, so it pays to more precisely establish its characteristics. To begin with, a relational representation should represent observations in terms of entities, where each entity is represented independently of one another. For instance, if an image contains two shapes and we want to represent their positions, one option is to express one object’s position as an absolute (x, y) coordinate, and then express the other as a translation of the first. The view in this thesis is that instead it is preferable if each object’s position is represented irrespective of the other, as its entity-wise (x, y) coordinates. Next, the representation should be amenable to decoding the relationships that exist between entities, which is to say that the information pertaining to each entity’s properties is structured in a way that permits comparison. Moreover, the manner in which these comparisons can be made should be *logical*, exhibiting logical consistency of and between comparisons and able to remain coherent when faced with new representations over equivalent properties. For instance, if entities are represented as (x, y) coordinates, then the relationship between them can be decoded as a relation that is true or false over the coordinates.

This thesis will refer to the process of comparison as a *relation-decoding*, facilitated by the notion

of a *relation-decoder*. The expectation that relation-decoders behave in a logically consistent manner means we can leverage the notions of model-based semantics of Logic to evaluate their quality. In particular, this thesis will employ First-Order Logic (FOL) as a lens to evaluate the quality of the relational representations learned through relation-decoding. Relational representations should be FOL-like, and each chapter will provide a different perspective on what this means.

1.2 Thesis structure and contributions

This thesis is structured as follows. Chapter 2 provides technical background regarding the technology and learning methods employed in this thesis. Furthermore, it provides background on representation learning, particularly the formalization of so-called disentangled representations, which are central to the work in this thesis.

Chapter 3 serves as a kick-off point, starting with an ambitious attempt at learning a relational representation in a lifelong learning-inspired Reinforcement Learning (RL) setting. This pilot experimental chapter seeks to learn FOL-like relation-decoders purely from reward, by applying a series of relational inductive biases to the learning architecture. Here, FOL-like refers to the discreteness and mutual exclusivity decoding characteristics of the resulting relation-decoders. Results demonstrate that while FOL-like properties are present, their effect is not as strong as desired, and, throughout learning, their FOL-like behaviour is unstable due to lower-level representation failures, motivating a closer look at the representation learning process.

This segues into Chapter 4, which employs state-of-the-art disentangled representation learning methods to learn a ‘greedy’ relational representation¹, in a weakly supervised setting. Here, a relational supervision is employed using weakly supervised relation-decoders, to ensure that they learn to approximate a desired set of relations, allowing for control over their influence on the lower-level representations. The resulting lower-level representation will then be evaluated for its FOL-like properties, which refer here to the ability to learn a subsequent set of relation-decoders that are logically related to the initial representation learning task set, and in the downstream reasoning performance of a state-of-the-art relational reasoner on an associated task.

¹In terms of capturing as much representational information pertaining to the input as possible

Taking a step away from experimentation and into theory, Chapter 5 establishes theoretical foundations to expand our understanding and assessment of what it means for a relational representation to be FOL-like, particularly expanding on the expected logical properties of each relation, and between relations. Key outcomes include a formal definition of a soft-structure (a DL applicable analog for a structure in model-theory) and associated definitions of consistency and coherence of relational representations. Building from these definitions, formal theorems are derived that establish the circumstances under which consistency and coherence can be guaranteed.

Chapter 6 then returns to experimentation to evaluate the practical value of the theory presented in the previous chapter. As part of this assessment, design-criteria for a consistent and coherent relation-decoder are extracted from the previous section and used to guide the design of a new class of Dynamic Comparator (DC) relation-decoders. Subsequent experimentation will then evaluate the FOL-like zero-shot transfer between relational representations. This is achieved using a proposed Partial Relation Transfer (PRT) experimental setup, that is designed to expose whether a set of relation-decoders, and the induced lower-level representations, can capture and leverage the logical dependencies inherent to the task. The evaluation will show that consistency and coherence, particularly when assessing broad regions of lower-level representational space, appear integral to the FOL-like zero-shot transferability of relation-decoders.

Related work are reviewed in Chapter 7, and Chapter 8 concludes the thesis, summarizing the contributions and discussing future research directions.

1.3 Publications

The work in this thesis produced the following peer-reviewed publications:

Strömfelt, H., Dickens, L., d’Avila Garcez, A. S., and Russo, A. (2020). On the transferability of vae embeddings using relational knowledge with semi-supervision. *CoRR*, abs/2011.07137

Strömfelt, H., Dickens, L., d’Avila Garcez, A. S., and Russo, A. (2021). Coherent and consistent relational transfer learning with auto-encoders. In *NeSy*, pages 176–192

Strömfelt, H., Dickens, L., d’Avila Garcez, A., and Russo, A. (2022). Formalizing consistency and coherence of representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 6873–6885. Curran Associates, Inc

Chapter 2

Background

This chapter provides the technical foundations that this thesis builds upon. These include Computational Logic, Deep Learning (DL) foundations, and a brief overview of relevant learning paradigms. Additionally, key background will be provided on representation learning and disentangled representations (including a state-of-the-art disentangled representation learning architecture), as these are central to the work that this thesis presents.

2.1 Computational Logic

The formal study of Logic provides a calculus for establishing the validity of arguments, and is a branch of mathematics and philosophy that aims to capture the principles of correct reasoning. Whether someone studies Logic (Hodges, 2001; Gensler, 2017) or not, they use the rules of Logic continually in their day-to-day live. This is because humans *reason* - we take what we know/think to be true or false, to produce conclusions. The foundation of Logic is often credited to Aristotle, as one of the first to systematically study and write about Logic and reasoning, particularly in terms of what makes an argument valid or invalid. Aristotle developed a formal system of deductive reasoning known as syllogistic logic, where statements (or premises) lead to conclusions. A canonical example of a syllogism is:

1. All men are mortal.
2. Socrates is a man.

3. Therefore, Socrates is mortal.

This is an example of *deductive reasoning*, where we use our knowledge of the world to make a conclusion. Syllogisms are concerned with notions “all,” “no” and “some,” and uses additional words “is” and “not.” Looking at the above syllogism under this lens, we can see the general structure:

1. All B is C.
2. A is B.
3. Therefore, A is C.

Drawing to every-day situations, a syllogism is collectively an argument, where premises are used to draw a conclusion. If an argument is *invalid* it means that the premises do not lead to the conclusion. If premises are not true, they are considered to be *unsound*, but the overall argument can still be *valid*. Soundness refers to the truth of the consequences from the truth of given premises, whereas validity refers semantic properties of a given argument.

The syllogism example shows how Logic formalises something used in everyday life. However, syllogisms are limited in their expressive capacity and are insufficient if we want to express some of the more complex arguments we might find in every day life. For instance, consider the following argument:

1. If it is raining, then the ground is wet.
2. It is raining.
3. Therefore, the ground is wet.

This argument is valid, but it is not a syllogism. It is an example of a *conditional* statement, where the truth of one statement is dependent on the truth of another. In this case, the truth of the statement “the ground is wet” is dependent on the truth of the statement “it is raining.” The critical new terminology here is the word “if,” which is used to express a conditional statement and “then,” which is used to express the consequence of the conditional statement. Furthermore, syllogisms do not let us discuss the truth of statements that are not universal, or

Table 2.1: An overview of the Propositional Logic (PL) connectives. These are presented in order of their binding strength, which dictates the order in which they must be evaluated within a formula.

Connective	Description	Example
Negation ($\neg$)	The negation of a proposition p is true if and only if p is false.	If p is “It is raining”, then $\neg p$ is “It is not raining”.
Conjunction ($\wedge$)	The conjunction of propositions p and q is true if and only if both p and q are true.	If p is “It is raining” and q is “It is cold”, then $p \wedge q$ is “It is raining and it is cold”.
Disjunction ($\vee$)	The disjunction of propositions p and q is true if and only if at least one of p or q is true.	If P is “It is raining” and q is “It is sunny”, then $p \vee q$ is “It is raining or it is sunny”.
Conditional ($\rightarrow$)	The conditional of propositions p and q is false if and only if p is true and q is false. Otherwise, it is true.	If p is “It is raining” and q is “I will use an umbrella”, then $p \rightarrow q$ is “If it is raining, then I will use an umbrella”.
Biconditional ($\leftrightarrow$)	The biconditional of propositions p and q is true if and only if p and q have the same truth-value.	If p is “It is raining” and q is “There are dark clouds”, then $p \leftrightarrow q$ is “It is raining if and only if there are dark clouds”.

statements which require “and” or “or” to express. To deal with this (and many more areas where at-the-time current logic was insufficient), the last century saw a great deal of work in the field of Logic, which of particular interest to this thesis, resulted in the development of Propositional Logic (PL) and First-Order Logic (FOL)¹.

2.1.1 Propositional Logic

PL², was developed as a formalisation of logic which makes it possible to formally define what constitutes a valid argument, by looking at the structure of the propositions instead of their content. In PL, a statement (or proposition) is represented via a *symbol* (typically a lowercase letter *e.g.* $p, p_1, q, q' \dots$), which, through the use of *connectives*, can be combined into more complex *formulae* (represented as uppercase letters $A, A_1, B, B' \dots$) describing the relationships between propositions. Note that a formula can be broken down into smaller parts, which are themselves formulae. Examples of connectives used in simple formulae are given in table Table 2.1. Connectives are defined as n -ary functions $\mathbb{B}^n \rightarrow \mathbb{B}$, where n is the

¹The content in this section is primarily sourced from (Hodges, 2001; Gensler, 2017).

²Also known as sentential logic, propositional calculus, sentential calculus, or zeroth-order logic.

Table 2.2: Truth-tables for the PL connectives.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

arity of the connective and $\mathbb{B} = \{\top, \perp\}$ is the set of truth-values (read “top” and “bottom”), which lets us define when a proposition or formulae is true or false. For instance, the *negation* connective is a unary function $\neg : \mathbb{B} \rightarrow \mathbb{B}$, which takes a single truth-value and returns the opposite truth-value. Using these connectives, the canonical syllogism example given above can be expressed in PL by replacing each statement with an independent symbol. We label the first premise a , the second premise b , and the conclusion c , and then use the conditional and conjunction connectives to express the relationship between them: $a \wedge b \rightarrow c$. In words: if both a and b are true propositions, then c must also be true. Note that this formula does not say anything about the truth of a , b , but instead describes a general relationship between the premise and conclusion³. To fully describe the meaning of the overall formula we test it under different *situations*, where each situation is a mapping from propositions to a truth-value. A useful tool in this analysis is the *truth-table*, which lists all possible situations and the truth-value of the formula/sub-formulae under each situation. Table 2.2 shows the truth-table for the example formula given in Table 2.1. The evaluation of formulae across all situations is what defines their validity, taking propositions as symbols and not concerning ourselves with their content. We can therefore say that the *semantics* of PL is given by evaluating formulae under situations, using the granular semantics of each connective as building blocks. When doing so, we must consider the relative binding strengths of each connective, which dictates which sub-formula (parts of the formula) are evaluated first. For instance, the negation connective has a higher binding strength than the conjunction connective, so the formula $\neg p \wedge q$ is evaluated as $(\neg p) \wedge q$ and not $\neg(p \wedge q)$. In the absence of parentheses, the binding strengths of the propositional connectives are ordered (from strongest to weakest): $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$. When parentheses are present, the formula is evaluated from the innermost parentheses outwards, internally following the above binding strength order.

³In PL a *premise* collectively refers to the sub-formula on the left of $\rightarrow$ and conclusion the sub-formula to the right.

In PL, an argument $A_1, \dots, A_n \models B$ says that when formulae A_1 through A_n are true, then formula B is also true. An argument is *valid* as long as this holds true in every situation. To say that an individual formula A is valid means that it is true in every situation and as an argument can be written as $\models A$. Arguments such as these are known as *tautologies*, and are always valid, for instance the formula $a \vee \neg a$ is always true. A *satisfiable* (or *consistent*) formula is one that is true in at least one situation, and an *unsatisfiable* (or *inconsistent*) formula is one that is false in every situation. For instance, the formula $a \wedge \neg a$ is unsatisfiable, as it is false in every situation - regardless of whether a is true or false, $a \wedge \neg a$ is false. A valid formula is satisfiable but a satisfiable formula may not necessarily be valid. These definitions can be extended to sets of formula, where a satisfiable set of formula means they are collectively true in at least one situation, and an unsatisfiable set of formula means there is no situation where all formula can be true at the same time.

Although PL lets us formalize many statements about the world and is far more expressive than the early syllogistic logic, it is still limited in its ability to express more complex statements. Most importantly, PL is unable to express general statements about objects and their properties, or the relationships between objects. For instance, the statement “All humans are mortal” cannot be expressed in PL, as it requires *quantifying* over all humans and all mortals. This is where first-order logic comes in.

2.1.2 First-Order Logic

FOL, also known as predicate logic, is an extension of PL that offers further expressiveness and generality in the statements that can be made. In particular, FOL introduces *relation* symbols (or *predicates*), *function* symbols, *variables*, *constants*, and *quantifiers*, to enable discourse over objects and their relationships. For example, in FOL we express “all humans are mortal” as $\forall A (\text{Human}(A) \rightarrow \text{Mortal}(A))$, where $\forall$ is the universal quantifier, **Human** and **Mortal** are *unary* (1-ary) relation symbols, and A is a variable. We could also express that “Socrates is Mortal” as **Mortal**(Socrates), by defining the constant, Socrates. When a formula has no variables, it is said to be *ground*. For instance, **Mortal**(Socrates) is a ground formula, whereas **Mortal**(A) is not. When a formula has no *free variables*, *i.e.* variables that are not *bound* to a quantifier, it is referred to as a *sentence*. For instance, $\forall A (\text{Mortal}(A))$ is a sentence, but **Mortal**(A) is not due, to the unbound variable A . In a sentence, a quantifier binds a variable to the scope of

Table 2.3: Overview of the syntactical elements of FOL.

Syntactical element	Description	Example
Variable	A symbol representing an arbitrary element of the domain	$A, A_1, B, \dots$
Constant	A symbol representing a specific, fixed element of the domain	$a, a_1, b, \dots$, or words/names like “Socrates”, “One”, “2”
Function	A symbol representing a mathematical n -ary function that takes n terms as arguments and returns a term.	$f(A), g(A, B)$
Term	A variable, a constant, or a function applied to terms	$A, a, f(A), g(A, B)$
Relation Symbol	A symbol representing a property or relation. It has a specific arity (number of arguments). Normally takes terms as arguments.	$p(\cdot)$ (unary relation symbol), $r(\cdot, \cdot)$ (binary relation symbol), and sometimes $r(\cdot)$ or $r(\cdot, \cdot)$ when appropriate
Atom	A relation symbol applied to a sequence of terms. It is the simplest type of formula that can’t be broken down into smaller parts.	$p(a), r(a, b)$
Literal	A relation symbol applied to a sequence of terms or the negation of the same	$p(a), \neg p(a, A), p(a, g(X))$
Connectives	Symbols representing logical connections between formulae	Same as in PL (see Table 2.1)
Quantifiers	Symbols representing universal or existential properties of variables	$\forall A (P(A))$ (“For all A , $p(X)$ is true”), $\exists A (p(A))$ (“There exists an A such that $p(A)$ is true”)
Equality	A special binary relation symbol that states two terms (variables or constants) are the same object	$A = B, A = a$
Formula	A syntactically correct combination of the above elements with unbound variables, denoted by $\square$	$(p(A) \rightarrow \neg q(B))$ or $p(A)$
Sentence	A formula with no unbound variables, denoted by τ	$\forall A (P(A) \rightarrow \neg Q(A))$ or $\forall A (P(A))$

the quantifier, so that the variable is only valid within the scope of the quantifier. Table 2.3 provides a summary of the full syntax of FOL, including some that have not been mentioned above, but are derivative, namely: *terms*, *literals*, *atoms*, and *equality*.

A FOL language, $\mathcal{L}^{\text{FOL}}$, comprises of a signature, logical connectives/quantifiers, and the sentences that can be expressed using the given signature and logical operators. An important construct here is the *signature*, denoted σ , comprising the non-logical⁴ symbols of a language, namely relation, function, and constant symbols. To put this altogether in a practical example, consider the following sentences:

$$\forall A, B \ ((\text{managerOf}(A) = B) \wedge \text{isManager}(B) \rightarrow \text{earnsMore}(B, A)) \quad (2.1)$$

$$\forall A, B \ (\text{earnsMore}(B, A) \leftrightarrow \text{isGreater}(\text{salaryOf}(B), \text{salaryOf}(A))). \quad (2.2)$$

In words, this sentence expresses a general rule that all managers (by the use of $\forall$) earn more money than their employees. The signature, σ , that is used to express this sentence includes $\mathcal{R} = \{\text{isManager}(\cdot), \text{isGreater}(\cdot, \cdot), \text{earnsMore}(\cdot, \cdot)\}$, $\mathcal{F} = \{\text{managerOf}(\cdot), \text{salaryOf}(\cdot)\}$, $\mathcal{C} = \emptyset$, where $\mathcal{R}$ is the set of relation symbols, $\mathcal{F}$ the set of function symbols, and $\mathcal{C}$ the set of constants.

To define the semantics of FOL we can refer to a *domain of discourse*, or *domain* which scopes the domain that we may describe using $\mathcal{L}^{\text{FOL}}$. In this case, an appropriate domain is the set of working people and natural numbers. Notice that σ does not include any constants, which means we do not speak about any individual working person, or a specific natural number, and only make general statements about the domain. However, to propose a sentence that explains a scenario in which a particular manager John may earn less than an employee, we could use $\mathcal{C} = \{\text{John}\}$ and the sentence:

$$\exists A \ ((\text{managerOf}(A) = \text{John}) \wedge \text{earnsMore}(A, \text{John})). \quad (2.3)$$

However, this cannot be the case if we take Equations 2.1 and 2.2 to be true. Together these sentences are inconsistent; there is no possible world that can satisfy each sentence.

This set of sentences starts to illustrate the semantics of FOL. The signature presents only a set of symbols - if we were to swap out the relation/function symbol names for single letters this becomes more apparent: what is the meaning of $\text{p}(A)$? It is only when we combine these symbols

⁴non-logical in that they have no inherent logical meaning, in contrast to connectives and quantifiers.

into sentences that we can start express the meaning of these symbols. We can discuss this more concretely using additional tools called *structures*. A structure is a tuple $\mathcal{S}_\sigma = \{\mathcal{S}, I_{\mathcal{S}_\sigma}\}$, where $\mathcal{S}$ is a non-empty domain and $I_{\mathcal{S}_\sigma}$ is a function that maps each symbol in the signature σ to sets of tuples over $\mathcal{S}$:

1. for an n -ary relation symbol, the structure defines the n -tuples of elements in $\mathcal{S}_n$ to express the semantics of the relation symbol, *e.g.* for a binary relation r ,

$$I_{\mathcal{S}_\sigma}(r) = \{ (s_i, s_j) \mid (s_i, s_j) \in \mathcal{S}_2, \text{ such that } r(s_i, s_j) \text{ is true} \};$$

2. for an n -ary function symbol, the structure determines the mapping from n -tuples of elements in $\mathcal{S}_n$ to elements of $\mathcal{S}$. A function over an n -tuple of elements from $\mathcal{S}$ always produces a term in $\mathcal{S}$,

$$I_{\mathcal{S}_\sigma}(f) = \{ ((s_i, s_j), s_k) \mid (s_i, s_j) \in \mathcal{S}_2, s_k \in \mathcal{S}, \text{ such that } f(s_i, s_j) = s_k \};$$

3. for a constant, the structure determines the element of $\mathcal{S}$ that the constant refers to, *e.g.* for a constant c ,

$$I_{\mathcal{S}_\sigma}(c) = s_i \in \mathcal{S}.$$

The notion of whether a structure satisfies a given sentence is an important one. A structure $\mathcal{S}_\sigma$ satisfies a *universally quantified* sentence⁵, τ , if we find that every ground instance of the sentence is true in $\mathcal{S}_\sigma$, denoted $\mathcal{S}_\sigma \models \tau$. A non-empty set of sentences is referred to as a theory $\mathcal{T}$. A theory is *satisfiable* if there is at least one structure that can satisfy all of the sentences in the theory. A structure that satisfies a theory gets a special name and is called a *model* of the theory (Hodges, 1993).

Definition 2.1 (Model of a theory) *Let $\mathcal{T}$ be a theory written in a language $\mathcal{L}^{FOL}$ and let $\mathcal{S}_\sigma = (\mathcal{S}, I_{\mathcal{S}_\sigma})$ be a structure, where σ is the signature of $\mathcal{L}^{FOL}$. $\mathcal{S}_\sigma$ is a model of $\mathcal{T}$ if and only if $\mathcal{S}_\sigma \models \tau$ for every sentence $\tau \in \mathcal{T}$.*

A structure replaces the notion of a situation in PL, where we evaluate structures to determine satisfiability of theories. This thesis will make use of FOL to express desirable FOL-like behaviour

⁵This thesis will only work with universally quantified sentences.

of Deep Learning architectures. The signatures that will follow will only consist of relation symbols and $r \in \sigma$ will be used as a shorthand for $r \in \mathcal{R}$.

2.2 Deep Learning

Deep Learning (DL) is a subfield of Machine Learning (ML) that has a long and storied history, heavily inspired by the desire to understand the human brain. Its earliest foundations can be traced back to the 1940s, when McCulloch and Pitts (1943) proposed a model of artificial neurons, the so-called McCulloch-Pitts neurons. However, to reach the modern DL systems we know today additional components were required, including innovations in *activation* function, the inclusion of scalar *weights* connecting neurons, and, critically, a scalable *learning* procedure to update these weights based on an *error signal*. Each of these aspects will be presented in the following⁶.

2.2.1 Artificial Neural Networks

The core ‘hardware’ of DL is the Artificial Neural Network (ANN). Fundamentally, an ANN consists of a collection of neurons connected by weights, which are critically learnable via an appropriate optimisation algorithm. When the aggregate weighted input to a neuron exceeds a certain threshold, the neuron is activated, according to an activation function, and its output may be passed on to other connected neurons. A basic ANN can be quite simply viewed as a function,

$$\hat{\mathbf{y}} = f(\mathbf{x}; \theta), \quad (2.4)$$

with weights θ , and input $\mathbf{x}$ and output $\hat{\mathbf{y}}$ vectors. The goal of the ANN will be to output a target $\mathbf{y}$, and so the weights θ will be learned such that its prediction $\hat{\mathbf{y}}$ is as close to $\mathbf{y}$ as possible⁷. Although other ANN variants exist, this thesis will focus on Multi-Layer Perceptrons (MLPs), also referred to as *feedforward* ANNs, wherein information flows ‘forward’ from input to output. In which case, under the hood f can be represented as a composition of functions:

$$\hat{\mathbf{y}} = f(\mathbf{x}; \theta) = f^L \circ f^{L-1} \circ \dots \circ f^1(\mathbf{x}; \theta_1), \quad (2.5)$$

⁶The content presented in this section is based primarily on (Goodfellow et al., 2016) and (Kruse et al., 2022).

⁷The way this learning takes place will be presented in Section 2.2.3.

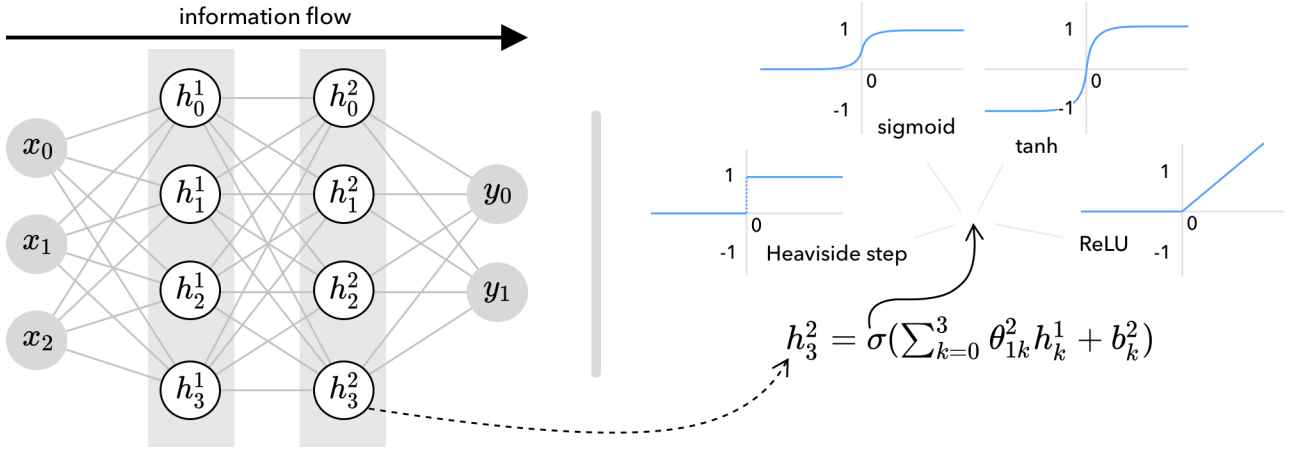


Figure 2.1: Illustration of a basic ANN and common activation functions. In the remainder of this thesis, sets of neurons will be generalised by blocks.

where $\circ$ denotes function composition, L is the number of layers, and f^l is the l -th layer. For $1 \leq l \leq L - 1$, the output of f^l is referred to as a *hidden layer*, whereas $\mathbf{x}$ is the *input layer* and $\hat{\mathbf{y}}$ is the *output layer*. In an MLP, each f^l is computed as a function of the previous layer,

$$f^l(\mathbf{h}^{l-1}; \theta^l) = \sigma(\boldsymbol{\theta}^l \mathbf{h}^{l-1} + \mathbf{b}^l) = \mathbf{h}^l. \quad (2.6)$$

where for d_l neurons in the l -th layer and d_{l-1} in the $(l - 1)$ th layer: $\boldsymbol{\theta}^l \in \mathbb{R}^{d_l \times d_{l-1}}$ is the weights matrix of the l -th layer, $\mathbf{b}^l \in \mathbb{R}^{d_l}$ is the *bias* vector, $\mathbf{h}^{l-1} \in \mathbb{R}^{d_{l-1}}$ is the output of the previous layer, and σ is an activation function typically of the form $\sigma : \mathbb{R} \rightarrow \mathbb{R}$. The overall ANN parameters are then $\theta = \{ \theta^l \mid l \in \{1, \dots, L\} \}$, where each $\theta^l = \{ \boldsymbol{\theta}^l, \mathbf{b}^l \}$. The overall structure of a deep ANN is depicted in Figure 2.1-left. The *deep* in DL gets its name from the fact that multiple hidden layers can be stacked together, where typically ANNs that have more than one hidden layer are considered deep. An important theoretical property of even single hidden layer ANNs is that they are *universal approximators*, meaning that with enough hidden neurons, they can approximate any function to an arbitrary degree of accuracy (Hornik et al., 1989).

Whilst early ANNs used a Heaviside step function as the activation (McCulloch and Pitts, 1943; Rosenblatt, 1958) (illustrated in Figure 2.1-right), which produces an “all-or-nothing” binary output, modern ANNs use a variety of more flexible activation functions⁸:

Logistic Sigmoid: an early activation function, which produces a scalar output in the range

⁸Although each activation is defined as acting on a scalar, they may also act on other dimensionality tensors, in which case it is assumed that they are applied element-wise.

$(0, 1)$,

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \quad (2.7)$$

The sigmoid is prone to the *vanishing gradient* problem, where the gradient of the function approaches zero as the input magnitude becomes large. This can limit how many layers can be stacked together, as the backpropagation signal quickly becomes attenuated, known as the *vanishing gradient* problem (see Section 2.2.3 for background on backpropagation).

Hyperbolic Tangent: a scaled version of the sigmoid activation function, which produces a scalar output in the range $(-1, 1)$,

$$\tanh(x) = 2\sigma(2x) - 1. \quad (2.8)$$

The tanh activation function is also prone to the vanishing gradient problem, but is centred around zero, and empirically tanh can perform better during optimisation.

Rectified Linear Unit: perhaps the most popular choice of activation function in modern ANNs, the Rectified Linear Unit (ReLU) activation function (Glorot et al., 2011; Nair and Hinton, 2010; He et al., 2015) produces a scalar output in the range $[0, \infty)$,

$$\text{ReLU}(x) = \max(0, x). \quad (2.9)$$

The ReLU activation function is not prone to the vanishing gradient problem, but the gradient is zero for $x \leq 0$ and can cause the *dying ReLU* problem, where neurons activations are always zero and so cannot be updated (Lu et al., 2019). However, it is thought that as long as some information processing pathways are differentiable, the network can learn to ignore the non-differentiable ones (Glorot et al., 2011), and that the sparsity that this induces can be beneficial (Glorot et al., 2011).

The above activations (illustrated in Figure 2.1) share two important properties. Firstly, they are *non-linear*, which is a requirement for ANNs to be able to approximate non-linear functions (since stacking linear operations can only produce another linear operation) (Bengio et al., 2013). Second, each activation is *differentiable*, allowing for the use of gradient-based learning algorithms which will be presented in Section 2.2.3. The partial exception is ReLU, which is not differentiable at $x \leq 0$, but is differentiable everywhere else. Lastly, a useful activation that

instead operates over the full layer is the Softmax function:

Softmax: The Softmax activation function is a ‘layer-aware’ generalisation of the sigmoid, which produces a vector output in the range $(0, 1)^{d_l}$ for the l th layer,

$$\text{softmax}(\mathbf{z}) = \mathbf{z}', \text{ where } z'_i = \frac{e^{z_i}}{\sum_{j=0}^{d_l-1} e^{z_j}}. \quad (2.10)$$

A key feature is that the sum of the elements of $\mathbf{z}'$ is one and so the Softmax can be used to transform the activations of a layer such that they can be treated as a probability distribution. This is particularly beneficial for multinomial classification tasks, but can also be used to produce an attention-like weighting (Bahdanau et al., 2015). The Softmax will be used as an attention mechanism in Chapter 6.

In the thesis the term *architecture* will be common. An architecture generally refers to some combination of ANN component functions to form a composite model, but can also refer to a simple MLP. Before exploring how such architectures can be trained in Section 2.2.3, we will first see some examples of ANN variants. The variants presented in this section are not exhaustive, but are the ones that will be used throughout this thesis.

2.2.2 Common architectures

Convolutional Neural Networks

In Equation 2.4, both the input and output of the ANN are vectors, but modern ANNs are able to work with more structured forms. In particular, the Convolutional Neural Network (CNN) makes it possible to directly operate on images, which may be represented as 3-dimensional tensors, where the first two dimensions correspond to the height and width of the image, and the third dimension corresponds to the colour channels (*e.g.* red, green, blue). The early CNN was developed by (Fukushima, 1980; Fukushima and Miyake, 1982), which took direct inspiration from the brain’s visual cortex. But CNNs would be popularised after subsequent work by LeCun and co-authors, which showed that CNNs could be trained with modern learning algorithms LeCun (1989); LeCun et al. (1989b,a), culminating in the seminal LeNet architecture (LeCun et al., 1998)⁹, which achieved state-of-the-art classification performance on the MNIST

⁹It was here that the name Convolutional Neural Network was introduced.

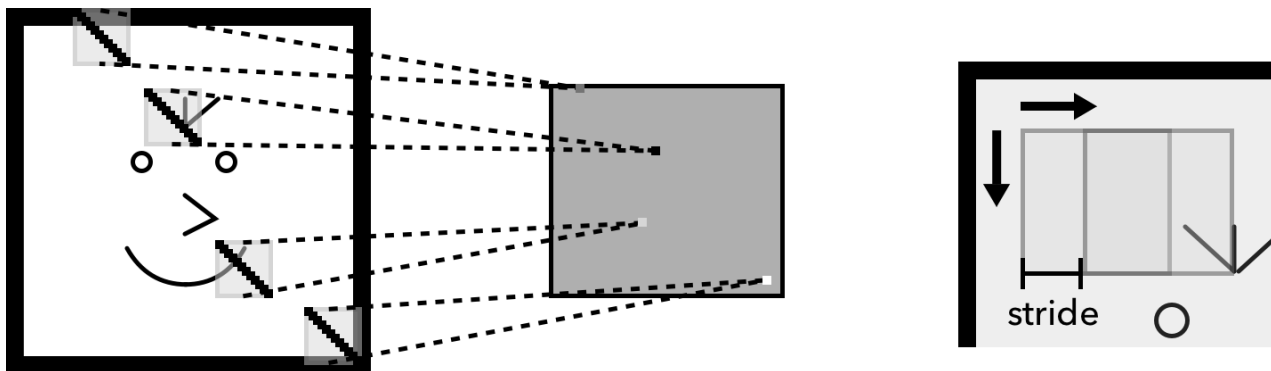


Figure 2.2: Illustration of a convolutional layer [Left] and a kernel convolution [Right]. The black borders indicate the padding and the small squares illustrate a kernel that is sensitive to diagonal edges. The overall activate of the kernel is aggregated to produce a smaller subsequent convolutional map. The kernel passes over its input according to a stride size.

handwritten digits dataset (LeCun et al., 1998).

CNNs incorporate local connectivity, weight sharing and partial translation invariance. This makes them well-suited to grid-like¹⁰ data such as 1-dimensional time-series data, and 2-dimensional images/video frames. The basic building block of a CNN is the *convolutional layer*, which is composed of a set of *kernels*. Each kernel is a small matrix of weights and the convolutional layer *convolves* the kernel across each each kernel-sized *receptive field* in the preceding layer data, leading to an aggregate activation for each receptive field. Receptive fields are typically far smaller than the input data and so the kernel acts as a small lens, looking for local regularities in each receptive field, which are then aggregated into a single activation. In doing so, the convolutional layer focuses on local connectivity patterns in the data. The receptive field can traverse the full input, through *sliding* of the kernel, where the amount of overlap between receptive fields is determined by the *stride* of the convolutional layer. Optionally the input data can be *padded* with zeros, in order to ensure that receptive fields pass over the edges of the data. In applying the kernel to each receptive field, the convolutional layer then produces a *feature map* and if the receptive field is smaller than the input data, the feature map will be smaller than the input data. This *reuse* of the same kernel across the input data is a form of weight sharing, where for each kernel a single set of weights needs to be stored and learned, rather than learning an explicit set of weights for each receptive field. Additionally, sliding (or convolving) the kernel across the full data means that a specific feature can be detected anywhere in the input, leading to a degree of translation invariance. Many kernels (1 per channel

¹⁰Grid-like' coming from the fact that the input data is arranged into discrete rows and columns, where spatial relationships between neighbouring elements are important.

or feature) can be defined for a single convolutional layer, which may each be sensitive to a different geometric property. By convolving multiple distinct kernels, the convolutional layer produces a set of feature maps which each capture a different geometric survey of the input.

To achieve increasingly abstract feature maps, convolutional layers can be stacked - whilst lower layers may be sensitive to edges, higher layers may be sensitive to more complex shapes. A common approach in stacking layers is to employ a *pooling layer*, which reduces the dimensionality of the feature maps. Each pooling operation is functionally similar to the convolving of a kernel across the input. However, in the case of *max pooling*, the aggregation involves taking the maximum activation across a receptive field. Pooling adds to the translation invariance of CNNs. For instance, the max pooling operation does not record the exact location of the maximum activation within its receptive field, and returns only its value. Typically, the final set of feature maps are flattened into a vector and passed through an MLP to produce the final output. The overall architecture of a typical CNN is shown in Figure 2.2.

In this thesis, a CNN will be used as part of an AutoEncoder architecture, which involves reconstructing images from compressed data representations obtained using the CNN. To be able to do this reconstruction, a *deconvolutional* network is used, which can be thought of as an inverse of a CNN, applying the same operations but in reverse (Zeiler et al., 2011; Zeiler and Fergus, 2014). The autoencoder is described in more detail in the following section.

Autoencoders

An AutoEncoder (AE) is a relatively simple architecture that consists an encoder ψ^{enc} and a decoder ψ^{dec} , which are both implemented using ANNs. In terms of learning, the objective of an AE is typically to reconstruct its input. In this case, we can view an AE as a function:

$$\hat{\mathbf{x}} = f(\mathbf{x}; \theta), \quad (2.11)$$

where f is an ANN. However, the AE's value is in the resulting intermediary representations of the data. In particular, AEs can be viewed as producing the following mapping:

$$\mathbf{x} \xrightarrow{\psi^{\text{enc}}} \mathbf{z} \xrightarrow{\psi^{\text{dec}}} \hat{\mathbf{x}} \quad (2.12)$$

where $\mathbf{x} \in \mathcal{X}$ is the input data, $\mathbf{z} \in \mathcal{Z}$ is the *latent code* (or as will be referred to in the main text, the *latent representation*), and $\hat{\mathbf{x}} \in \mathcal{X}$ is the reconstructed input data. $\mathcal{X}$ and $\mathcal{Z}$ are vector spaces and commonly $\dim(\mathcal{Z}) < \dim(\mathcal{X})$, which can encourage the AE to learn an encodings that capture only the most salient features of the data. Moreover, the AE is typically structured such that the dimensionality of the representation is reduced up until latent code $\mathbf{z}$, and then increased again up until the reconstruction. Under appropriate assumptions, this latent code can then be taken as a compressed representation of the data and may be useful for other tasks. AE-based architectures are used throughout this thesis following this property¹¹.

An illustration of an AE is provided by Figure 2.4, where it is contrasted against a Variational AutoEncoder (VAE), an overview of which is presented in Section 2.4.2.

2.2.3 Optimisation

To leverage the expressiveness of ANNs for a given task, we need to learn the parameters of the architecture. Firstly we need a measure of performance. This is achieved through defining a given *loss function*, $\mathcal{L}(\mathcal{D}; \theta)$ ¹², where $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$ is a dataset consisting of inputs $\mathbf{x}_i$ and corresponding targets $\mathbf{y}_i$. The aim of learning is to find the parameters that minimise the loss function on the training data, via the *optimisation*:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\mathcal{D}; \theta). \quad (2.13)$$

The primary optimisation technique, which propelled ANNs to the forefront of ML research, uses *gradient descent* to update the ANN weights against the loss function. At its core, gradient descent involves updating parameters via

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta} \mathcal{L}(\mathcal{D}; \theta), \quad (2.14)$$

¹¹More specifically due to the structure that can be influenced via appropriate techniques, which is discussed further in Section 2.4.

¹²Also commonly referred to as a *cost* or *objective* function

where the *learning rate* α determines the size of the update step and $\nabla_{\theta} \nabla_{\theta} \mathcal{L}(\mathcal{D}; \theta)$ is the *gradient* of the loss function with respect to the parameters:

$$\nabla_{\theta} \mathcal{L}(\mathcal{D}; \theta) = \left(\frac{\partial \mathcal{L}(\mathcal{D}; \theta)}{\partial \theta_1}, \dots, \frac{\partial \mathcal{L}(\mathcal{D}; \theta)}{\partial \theta_n} \right). \quad (2.15)$$

For gradient descent to be possible, the entire ANN architecture must be differentiable¹³. In which case, the chain rule can be used to compute the gradient of the loss function with respect each parameter via:

$$\frac{\partial \mathcal{L}(\mathcal{D}; \theta)}{\partial \theta_i} = \frac{\partial \mathcal{L}(\mathcal{D}; \theta)}{\partial o_{i+1}} \frac{\partial o_{i+1}}{\partial o_i} \frac{\partial o_{i+1}}{\partial \theta_i} \quad (2.16)$$

where o_i is the output of the i th layer. The first term in Equation 2.16 is the gradient of the loss function with respect to the output of the $i + 1$ th layer. This is then *backpropagated* through to the preceding layers to compute the gradient of the loss function with respect to the parameters of the i th layer. This is the *backpropagation* algorithm, which gets its name from the fact that error signals are first calculated at the end of the network and then propagated backwards (Rumelhart et al., 1985, 1986). A benefit of backpropagation, which will be exploited throughout this thesis, is that certain weights can be *frozen* whilst others continue to be updated. Gradients can still be propagated through the frozen weights, but the weights themselves will not be updated. Loss functions are an integral part of ANN training and depending on the task at hand a different loss function may be more or less appropriate. Some common options include:

Mean Absolute Error (MAE): commonly used for regression tasks, where the output is a continuous variable:

$$l(\mathcal{D}; \theta) = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (2.17)$$

The MAE loss will be used in Chapter 3 to train a Reinforcement Learning architecture.

Mean Squared Error (MSE): also commonly used for regression tasks, but is more sensitive to outliers than the mean absolute error:

$$l(\mathcal{D}; \theta) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2, \quad (2.18)$$

¹³Which is why early Heaviside activation function was not suitable and innovation was required.

Binary Cross Entropy: commonly used for binary classification tasks:

$$l(\mathcal{D}; \theta) = -\frac{1}{n} \sum_{i=1}^n (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)), \quad (2.19)$$

This loss function will be used in Chapters 4 and 6 to train a set of ANN binary predictors.

Note that here that each $\hat{y}_i$ and y_i are scalars, but in practice ANNs may produce a vector of outputs, $\hat{\mathbf{y}}_i$, and the ground-truth may be a vector, $\mathbf{y}_i$. In which case the loss function is applied element-wise and then aggregated over the vector. An important concern when defining a loss function is whether the overall loss differential is convex or not. If it is, then there is a single global minimum, which is the optimal solution. However, convexity may not be guaranteed and, in this case, learning may get trapped in local minima. Loss functions may also include *regularisation*, which can play a pivotal role in the success of an architecture. In particular, they can help to avoid *overfitting*, where a model effectively memorises the training data but fails to generalise to unseen data. Penalty based regularisation is a form of *inductive bias* towards a certain set of solutions, by computing additional loss terms that are a function of the architecture itself (*e.g.* the parameters or the activations) (Goodfellow et al., 2016). A common choice is to encourage the parameters to be small through *weight decay*, leading to an overall loss function of the form:

$$\mathcal{L}(\mathcal{D}; \theta) = l(\mathcal{D}; \theta) + \underbrace{\lambda \sum_{i=1}^n \|\theta_i\|_2^2}_{\text{weight decay}}, \quad (2.20)$$

where $\|\cdot\|$ is the L^2 norm and λ controls the strength of the regularisation. In the above, both λ and α are referred to as *hyperparameters*, which are parameters that are not learned through gradient descent, but instead are set by the user. Suitable tuning of hyperparameters is a key part of ANN research and may be found using manual tuning, or automated search (grid search or bayesian optimisation) (Goodfellow et al., 2016; Wu et al., 2019). Many more forms of regularisation exist and the search for suitable regularisation is a key part of ANN research. In this thesis, regularisation will play an important role in the architectures that will be used. In particular, in Chapter 4 and Chapter 6, the AE architecture will incorporate a regularisation based on the statistical properties of its intermediary outputs, to help produce meaningful representations of the data.

Table 2.4: A taxonomy of supervision types, adapted from (Hernández-González et al., 2016). A ‘class’ may refer to the overall classification type (*e.g.* a car), or a specific sub-property (*e.g.* the colour of the car).

Type	Description
Fully Supervised	Each instance comes with complete class information. Usually ‘supervised learning’ should be interpreted as ‘fully supervised learning’.
Unsupervised	Instances lack any class information.
Supervised (Semi)	A mix of instances, some fully supervised and others without supervision.
Supervised (Mutual Label Constraints)	For groups of instances, there is an explicit relationship among their class labels (<i>e.g.</i> , all instances sharing the same classification).

2.2.4 Forms of supervision

In practice the loss function is typically computed with respect to a set of training data, $\mathcal{D}$, which comprises a set of input-output pairs, $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$, where output pairs provide ground-truth labelling (often curated by practitioners). However, as we saw in Section 2.2.2, it may also be computed purely from the input data, $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. The former is referred to as *supervised learning*, and the latter as *unsupervised learning*, where no explicit ground-truth labelling is provided. However, there is a spectrum of levels of supervision in between these two extremes.

A selection of these is provided in Table 2.4, following the taxonomy provided by Hernández-González et al. (2016) for classification tasks. Only the levels relevant for this thesis are provided. In particular, the experiments that will follow in Chapters 4 and 6 will combine an unsupervised AE with semi-supervision using mutual label constraints. Since “semi-supervision using mutual label constraints” is cumbersome to reiterate, following Hernández-González et al. (2016) the term *weak supervision* will be used as a catch-all for supervision in between fully supervised and unsupervised learning.

2.3 Types of learning

Multiple types of learning exists, and this thesis will focus on a number of them. To provide a clear taxonomy between learning types is difficult, since there can easily be overlap between them. Here three types will be briefly covered: Transfer Learning (TL), Multi-Task Learning (MTL),

and Reinforcement Learning (RL). The reason they are split up and discussed individually is that their fundamental problem definitions are quite different. However, as will be seen in the remainder of this thesis, they can be combined in interesting ways. Chapter 3 uses RL to train an agent in a lifelong learning inspired task; in being lifelong inspired, meaning the agent is expected to perform in a number of different scenarios over its lifetime, TL plays an important role (Thrun and Mitchell, 1995). Chapters 4 and 6 use a MTL setup to jointly learn representations of data and binary ANNs predictors that are useful in downstream tasks and so transfer is again important¹⁴.

2.3.1 Transfer Learning

At its core, TL involves learning a source task, and then learning one or more target tasks using the knowledge gained from the source task. Tasks may differ in their input data distribution, their output task, or both (Pan and Yang, 2010). However, even if distributions differ, it may be the case that either the tasks are different but related, or there is some representational overlap between input data. For the former, consider a simulated robot learning to stack cubes, and then later wanting a real-world robot to perform the same task. An example of the latter could be learning to classify images of cats and dogs, and then later using this knowledge to classify images of other animals. TL is especially useful when the target task has a small amount of labelled data available, or when errors in the target task carries a high risk and so learning in the target task is not desirable. A common approach to TL is to take an architecture that has been pretrained on a large dataset, where there is likely overlap in the data distribution with the target task, and then fine-tune the architecture on the target task (Ding et al., 2023; Han et al., 2021). Relevant forms of TL are zero-shot learning, one-shot learning, and few-shot learning, in which the target task has no, one, or a few labelled examples respectively.

2.3.2 Multi-Task Learning

MTL involves learning a set of tasks in conjunction (Caruana, 1997), where the overall architecture can be divided into task-specific parameters and task-shared parameters. It was shown in early work that assuming task-relatedness, training with shared parameters means that with

¹⁴The content in this section is primarily based on (Goodfellow et al., 2016).

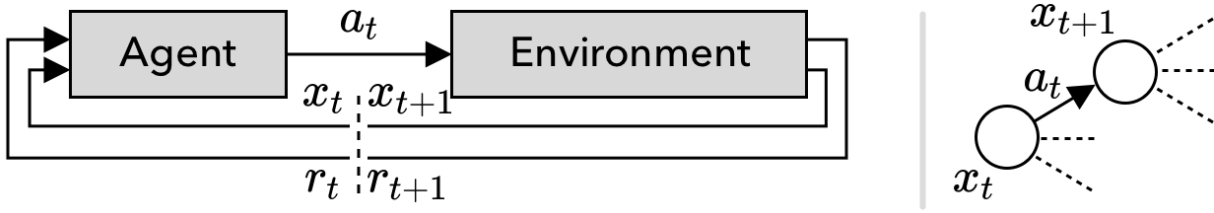


Figure 2.3: An illustration of the agent-environment interaction modelled as an Markov Decision Process (MDP). The left shows the agent perceiving state x_t and reward r_t , and then choosing a new action a_t to take. In taking a_t , the environment is transitioned to state x_{t+1} , which the agent subsequently perceives, together with a possible elicited reward r_{t+1} . On the right is a basic illustration of this overall process, where nodes are states, and arrows are actions taken and dotted lines actions not taken. In this case the agent has three possible actions in each state *i.e.* $\forall x \in \mathcal{X}, |\mathcal{A}(x)| = 3$. In this plot it is assumed that the environment is deterministic, in which case each actions leads to a single state. On the other hand, in a stochastic environment each action may lead to multiple states, with some probability.

far fewer examples across tasks, a greater overall generalisation performance and error bound can be achieved by the representation learning component, compared to training each task independently (Baxter, 1995). The key difference between MTL and TL is that in MTL the tasks are encountered jointly, whereas in TL the tasks are encountered one after another.

2.3.3 Reinforcement Learning

RL is another form of learning that is employed in Artificial Intelligence (AI) that took early inspiration from the way animals learn by interacting with their environment. As a field of study within AI, RL has a very large research domain, with applications spanning multiple fields including (but not limited to): robotics, autonomous driving, game playing, and even industrial resource management (Shakya et al., 2023). Some of the most prominent successes of AI have been achieved by combining RL with DL into so-called Deep Reinforcement Learning (DRL), such as surpassing human-level performance on Atari games (Mnih et al., 2015) and the related AlphaGo program Silver et al. (2017). RL will be used in Chapter 3 to train an agent to accomplish a lifelong learning inspired task. Overall, RL serves only a small role in this thesis, and so only the required concepts will be covered here. For a much more extensive introduction to RL, which the following text itself is primarily based on, the reader is referred to the textbook by Sutton and Barto (2018).

Markov Decision Process

A common way to model an RL problem is as a MDP, in which a learner (*agent*) exists in an environment that it perceives through *state* and in taking *actions* in the environment, may be *rewarded*. Concretely, at time t the agent will be in state $x_t \in \mathcal{X}$ ¹⁵, and will take an action $a_t \in \mathcal{A}(x_t)$, where available actions may be a function of the current state. In taking the action, the environment *transitions* to a new state $x_{t+1} \in \mathcal{X}$, and the agent receives a reward $r_{t+1} \in \mathcal{R} \subset \mathbb{R}$. When this process is repeated over time, starting at time t and leading up to time T , a trajectory appears with the form:

$$(x_t, a_t, r_t, x_{t+1}, a_{t+1}, r_{t+1}, \dots, x_T, a_T, r_T) \quad (2.21)$$

Through this trajectory, the agent will receive a set of rewards, where the total future reward from time t can generally be denoted by the *return*, $G_t = \sum_{k=0}^T r_{t+k+1}$. As trajectories can be long (or even theoretically infinite in *continuing* tasks) it can be useful to introduce a *discount rate* $\gamma \in [0, 1]$, which discounts future rewards by a factor of γ at each timestep. In the infinite case, this stops the total reward from easily diverging and becoming infinite, which could lead to an agent that has no ‘motivation’ to act in the shorter term. A discount factor of $\gamma = 0$ means that the agent only cares about the immediate reward, whereas $\gamma = 1$ means that the agent cares about all future rewards equally. With γ the future *discounted return* is defined as

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad (2.22)$$

which may also be defined recursively as

$$G_t = r_{t+1} + \gamma G_{t+1} \quad (2.23)$$

The equation can be left unchanged for *episodic* tasks (where $T \neq \infty$) by assuming that $G_T = 0$.

¹⁵In most textbooks, s is typically used to denote a given state, but to stay consistent with the remainder of the thesis, in which $\mathcal{X}$ denotes possible agent *observations*, x is used instead.

Optimisation

The goal of the agent is to learn a policy π that maximises the expected return. A policy is a probability distribution $\pi(a|x) = p_{A|X}(a|x)$ that determines the probability of taking an action when in a particular state. A good policy can be defined as one that maximises the *action-value function*:

$$q_\pi(x, a) = \mathbb{E}_\pi[G_t | X_t = x, A_t = a] \quad (2.24)$$

$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | X_t = x, A_t = a\right]. \quad (2.25)$$

which can be optimised via the optimisation task:

$$\pi^* = \arg \max_{\pi} q_\pi(x, a). \quad (2.26)$$

Chapter 3 will use *temporal difference* learning, which is a popular choice in modern RL applications¹⁶. Using temporal difference learning, the agent can learn from its own experiences, without any background knowledge provided. This is done by gradually incorporating true experienced reward over expected reward after each timestep. This difference is a *temporal difference error*, and is used to update the agent's policy:

$$\delta_t = r_{t+1} + \gamma Q_\pi(x_{t+1}, a_{t+1}) - Q_\pi(x_t, a_t). \quad (2.27)$$

where Q_π is used over q_π to denote the agent's estimate of the action-value function. The choice of the word 'estimate' is deliberate, since in practical scenarios the agent will not be able to observe all possible states, and so will have to generalise from its experiences¹⁷. To manage this estimate, this thesis uses a technique called Q-learning:

$$Q(x_t, a_t) \leftarrow Q(x_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(x_{t+1}, a) - Q(x_t, a_t) \right), \quad (2.28)$$

¹⁶The form of temporal difference reviewed here is known as TD(0), given comparison over a single timestep. For an extensive review of temporal difference and other available optimisation options see (Sutton and Barto, 2018).

¹⁷For smaller sized state spaces, *tabular* methods can be used, where the agent maintains a table of all possible states and their associated value. See (Sutton and Barto, 2018) for more details on tabular methods.

where α is a learning rate. A major breakthrough in RL came about when Q-learning was combined with DL, initiating the branch of so-called DRL, where it was found that through zero-knowledge Q-learning an agent could learn to surpass human performance a variety of Atari games (Mnih et al., 2015). This involves using ANNs (or more specifically CNNs) as function approximators for the Q -values. The same technique will be used in Chapter 3, where an agent will be trained using DRL Q-learning to accomplish a lifelong learning inspired task.

An important distinction between RL and other DL training procedures, is that RL is focused on interactive domains, where an agent is required to learn from its own experience, guided by reward signals. In some ways the reward supervises the agent, but it is not a direct supervision signal, as the agent must learn to associate its actions with the reward it receives. Additionally, RL algorithms must consider the trade-off between exploration and exploitation: exploration is necessary so that the agent may find more rewarding actions, but exploitation is necessary so that the agent may take advantage of the actions it already knows to be rewarding. These two need to be continuously balanced for the agent to arrive at an optimal policy.

2.4 Representation Learning

An important concept in this thesis is *representation learning*, which was briefly referred to with reference to autoencoders in Section 2.2.2 and again in Sections 2.3.1 and 2.3.2. Representation learning differs from other forms of learning as the interest is in the general properties of the data and the usefulness of these representations when considering current and, more importantly, possible future tasks. Representations are broadly important to all forms of AI, but the focus here will be on representation learning in the context of DL, where the representations are learned by ANNs.

An assumption that will be important in this thesis is the notion that observable data is generated by a set of Explanatory Factors (EFs)¹⁸ (Bengio et al., 2013), which can be thought of as independent explanatory properties about the data, whose variation can be considered mutually orthogonal, *i.e.* independent of one another. For instance, when looking at an object,

¹⁸Also commonly referred to as an “explanatory factors of variation”, a “factors of variation”, or a “generative factors” (Locatello et al., 2019; Bengio et al., 2013; Higgins et al., 2017a). When working with generated *visual* domains the the term ‘generative factor’ may lead to assumptions that the factors are strictly perceptive. For this reason, this thesis uses Explanatory Factor to emphasise the fact that factors may not only be perceptive.

the relevant EFs may include its size and colour. According to Bengio et al. (2013) a ‘good’ representations that incorporate these priors are said to *disentangle* the EFs and for an AI to understand the world it inhabits, it must “learn to identify and disentangle the underlying explanatory factors hidden in the observed milieu of low-level sensory data” (Bengio et al., 2013). In the following two subsections, a theoretical formalisation of disentanglement from Higgins et al. (2019) will be outlined, as it is important to the ideas presented in this thesis. Then, an example architecture for practical learning of disentangled representations will be presented, which will be used throughout this thesis¹⁹.

2.4.1 A theoretical definition of disentanglement

The motivation behind representation learning is grounded in the real-world, where the representations that are learned should exhibit properties that make them generally useful for learning the so-called AI-set (Bengio et al., 2013; Bengio and LeCun, 2007). Taking a step further, in disentanglement the goal is to achieve a representation which more readily exposes the underlying EFs of data:

An AI must fundamentally understand the world around us, and we argue that this can only be achieved if it can learn to identify and disentangle the underlying explanatory factors hidden in the observed milieu of low-level sensory data. ... [T]he most robust approach to feature learning is to disentangle as many factors as possible, discarding as little information about the data as is practical. (Bengio et al., 2013)

These notions were formalized in (Higgins et al., 2019) and more recently further supported in (Higgins et al., 2022), drawing inspiration from symmetries in physics. A symmetry is a transformation that can be applied to objects in order to change a specific property and leave all others unchanged. The theoretical foundation of symmetries stems from Noether (1915) who showed that every conservation law in physics is coupled to a corresponding continuous symmetry transformation. For instance, the conservation of momentum arises from the symmetry of spatial translations. One can think of these transformations as the “joints of the world” (Higgins et al., 2022), and if we assume our data behaves according to certain symmetries

¹⁹The content in this section is primarily sourced from (Bengio et al., 2013; Goodfellow et al., 2016; Higgins et al., 2019).

then an opportunity arises in providing architectures with inductive biases that can exploit these symmetries (Goyal and Bengio, 2022). Examples of these include the spatial invariance properties of CNNs (Section 2.2.2), which is intuitively beneficial to the classification of objects since the position of an object in an image should not affect the classification. In the following invariance and symmetry will be used interchangeably.

Symmetries transformations are invertible and compositional, and so can be mathematically abstracted into groups. A group G is defined as a non-empty set $\mathcal{G}$ together with a binary operation on $\mathcal{G}$, denoted by $\cdot$, such that: $\cdot : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$. Moreover $\cdot$ is associative and there exists an identity element $e \in \mathcal{G}$ such that $\forall g \in \mathcal{G}, e \cdot g = g$ holds, and for each $g \in G$ there exists an inverse element $g^{-1} \in G$ such that $g \cdot g^{-1} = e$ holds. For groups representing symmetry transformations, each $g \in \mathcal{G}$ is a symmetry transformation, and the group operation $\cdot$ is the composition of two symmetry transformations. To apply a symmetry transformation g to an object of some set $s \in \mathcal{S}$ a group action α is required: $\alpha : \mathcal{G} \times \mathcal{S} \rightarrow \mathcal{S}$. Throughout the thesis, to keep notation concise and following Higgins et al. (2022), the application of a group action $\alpha(g, s)$ will be expressed as $g \cdot_{\alpha} s$. Group actions have two defining characteristics:

1. compatibility between the group operation and group action: $g_1 \cdot_{\alpha} (g_2 \cdot_{\alpha} s) = (g_1 \cdot g_2) \cdot_{\alpha} s$; $g_1, g_2 \in G$; $s \in \mathcal{S}$.
2. identity: $e \cdot_{\alpha} s = s$; $e \in \mathcal{G}$; $s \in \mathcal{S}$, *i.e.* the group identity acting on s leaves s unchanged.

Symmetries then arise if we take a function, $f : \mathcal{X} \rightarrow \mathcal{Y}$ mapping elements from set $\mathcal{S}$ to set $\mathcal{Y}$, where if $f(g \cdot_{\alpha} s) = f(s)$, $\forall g \in \mathcal{G}, s \in \mathcal{S}$, then f is symmetrical to the transformations that g imposes on s . For our above CNN example, if f is a classifier, s an image of a cat, and g a rotation of the image, then f should be invariant to g as the ‘cat-ness’ of the image should not change through rotation.

Before moving to the final disentanglement definition, let us ground what we have back into the context of real-world representation learning scenarios, the running assumption is that, as discussed in (Bengio et al., 2013), data is generated from a set of EFs. Higgins et al. (2019) formalize this aspect as follows. Let $\mathcal{W}$ be a set of world-states and suppose that there is a generative mapping

$$f_{\mathcal{W} \rightarrow \mathcal{X}} : \mathcal{W} \rightarrow \mathcal{X}, \quad (2.29)$$

where $\mathcal{X}$ is the set of real-world observations, for example pixel representations, audio signals, or any combination of sensory perceptions. Furthermore, let

$$\psi : \mathcal{X} \rightarrow \mathcal{Z} \quad (2.30)$$

be an inference mapping, where $\mathcal{Z}$ can be viewed as the internal representation space of an observer, and let

$$f_{\mathcal{W} \rightarrow \mathcal{Z}} : \mathcal{W} \rightarrow \mathcal{Z} \quad (2.31)$$

be the composition

$$f_{\mathcal{W} \rightarrow \mathcal{Z}} = \psi \circ f_{\mathcal{W} \rightarrow \mathcal{X}}. \quad (2.32)$$

Now moving to Higgins et al. (2019)'s definition itself:

Definition 2.2 (Disentanglement) *Consider a group G that decomposes into subgroups $G = G_1 \times \cdots \times G_n$, with each subgroup G_i acting on a specific facet of the world-state via subgroup action $\cdot_\alpha^i$. Let $\mathcal{W}$ be the world-state space and let $\mathcal{Z}$ be an observer's representation space, and let $f_{\mathcal{W} \rightarrow \mathcal{Z}}$ be defined as in Equation 2.31. Then, $\mathcal{Z}$ is said to be disentangled with respect to the decomposition of G if:*

1. *There exists an action $\cdot_\alpha : \mathcal{G} \times \mathcal{Z} \rightarrow \mathcal{Z}$ with corresponding subgroup actions $\cdot_\alpha^j : \mathcal{G}_j \times \mathcal{Z}_j \rightarrow \mathcal{Z}_j$ for each subgroup $\mathcal{G}_j$, $j \in \{1, \dots, n\}$.*
2. *The function $f_{\mathcal{W} \rightarrow \mathcal{Z}}$ is equivariant between actions on $\mathcal{W}$ and $\mathcal{Z}$, and so satisfies $g \cdot f_{\mathcal{W} \rightarrow \mathcal{Z}}(w) = f_{\mathcal{W} \rightarrow \mathcal{Z}}(g \cdot_\alpha w)$ for all $g \in \mathcal{G}$ and $w \in \mathcal{W}$.*
3. *$\mathcal{Z}$ decomposes into $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_n$, where each $\mathcal{Z}_i$ is invariant under the subgroup action $\cdot_\alpha^j$ for all $j \neq i$.*

When $\mathcal{Z}$ is a vector space, as it will be in this thesis, it decomposes into independent vector subspaces $\mathcal{Z} = \mathcal{Z}_1 \oplus \cdots \oplus \mathcal{Z}_n$, where $\oplus$ denotes the direct sum of vector spaces. Moreover, an additional assumption will be made that $\mathcal{W}$ similarly decomposes into independent subspaces $\mathcal{W} = \mathcal{W}_1 \times \cdots \times \mathcal{W}_n$. As Higgins et al. (2019) state, though this is not required for the definition of disentanglement, it is the natural assumption, as it enforces the notion that the world follows the symmetrical structure defined by G 's symmetry transformations. Here, G intuitively defines

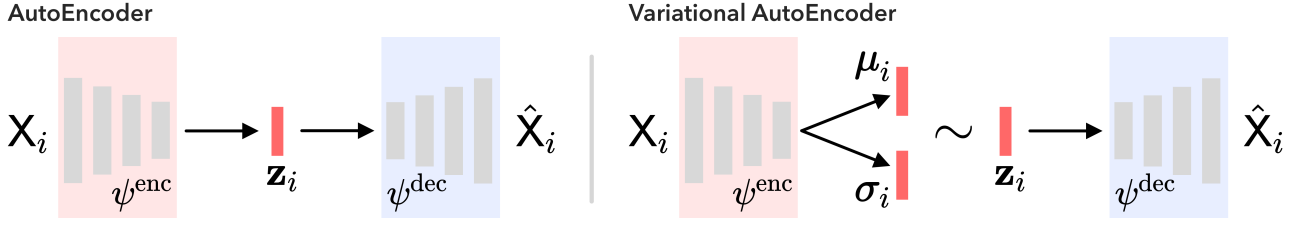


Figure 2.4: **[Left]** a standard AE architecture and **[Right]** a Variational AutoEncoder (VAE). The plot assumes the latent representation is the smallest size layer and that the input is a tensor $\mathbf{X}_i$. In contrast to the AE, the VAE encoder ψ^{enc} outputs distribution parameters from which the latent representation $\mathbf{z}_i$ is sampled.

the fundamental independent changes that govern the construction of set $\mathcal{W}$ (*i.e.* the ‘physically’ allowed world-states), and can be thought of as the ‘natural’ symmetries.

2.4.2 Variational AutoEncoders

VAEs are a popular method for practical learning of disentangled representations (Higgins et al., 2017a; Kingma and Welling, 2014; Eddahmani et al., 2023). A VAE is a form of autoencoder, but one that optimises against the generative model of the data:

$$\theta^* = \arg \min -\ln p_{\mathbf{X}}(\mathcal{X}; \theta), \quad (2.33)$$

where $-\ln p_{\mathbf{X}}(\mathcal{X}; \theta)$ is the negative log-likelihood of the data, and it is assumed that the observable data is generated via a latent variable model,

$$p_{\mathbf{X}}(\mathcal{X}; \theta) = \int_{\mathcal{Z}} p_{\mathbf{X}|\mathbf{Z}}(\mathbf{X}|\mathbf{z}; \theta) p_{\mathbf{Z}}(\mathbf{z}; \theta) d_{\mathbf{z}}. \quad (2.34)$$

However, this integral is intractable, and so the VAE instead optimises against the Evidence Lower Bound (ELBO), which provides a lower bound to the evidence $p_{\mathbf{X}}(\mathcal{X}; \theta)$,

$$\mathcal{L}^{\text{ELBO}}(\mathcal{X}; \omega, \theta) = \sum_{i=1}^N [\mathbb{E}_{\mathbf{z}_i \sim q(\mathbf{z}|\mathbf{X}_i; \omega)} [\log p(\mathbf{X}_i|\mathbf{z}_i; \theta)] - D_{\text{KL}}[q(\mathbf{z}|\mathbf{X}_i; \omega) \parallel p(\mathbf{z}; \theta)]] \quad (2.35)$$

where D_{KL} is the Kullback-Leibler divergence and $q_{\mathbf{Z}|\mathbf{X}}(\mathbf{Z}|\mathbf{X}; \omega)$ is an approximate posterior distribution, with parameters ω , over the latent space $\mathcal{Z}$, given the observable data $\mathcal{X}$. Note that here the ELBO is defined in its stochastically optimisable form following (Kingma and Welling, 2014) (and simultaneously, but separately Rezende et al. (2014)) who showed that optimising the ELBO can be achieved using an autoencoder architecture, where the approximate

posterior $q_{Z|X}(Z|X; \omega)$ is modelled as an encoder and the generative model $p_{X|Z}(X|Z; \theta)p_Z(Z; \theta)$ is modelled as a decoder. However to realise this ambition, an important tweak is required, because the sampling process of the approximate posterior $\mathbf{z} \sim q_{Z|X}(Z|X; \omega)$ is not differentiable as it stands. To circumvent this issue, Kingma and Welling (2014) use a *reparametrization trick*, which offloads the randomness to a noise random variable that can be combined with the approximate posterior parameters to produce a sample from $q_{Z|X}(Z|X; \omega)$, as:

$$\mathbf{z}_i \sim q_{Z|X}(Z|X; \omega) \text{ with } \mathbf{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_i, \quad (2.36)$$

where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a noise random variable, with identity covariance matrix $\mathbf{I}$ and zero vector mean $\mathbf{0}$, $\odot$ is the Hadamard product, and $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are the mean and standard deviation of the approximate posterior $q_{Z|X}(Z|X; \omega)$. All vectors here have the same dimensionality of $\dim(\mathcal{Z})$. In practice, the encoder is used to output $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ for each $\mathbf{X}_i$, and as $\boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}$ is differentiable with respect to these parameters, gradients can be propagated through the full network. This reparametrization assumes that the approximate posterior $q_{Z|X}(Z|X; \omega)$ is Gaussian, which is a common choice. This choice is usually coupled with an additional assumption that $p(X; \theta) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, which makes the D_{KL} term in Equation 2.35 computable in closed form, as:

$$D_{\text{KL}}[\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^{\odot 2} \mathbf{I}) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I})] = \frac{1}{2} \sum_{j=1}^{\dim(\mathcal{Z})} (1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2). \quad (2.37)$$

where $\boldsymbol{\sigma}^{\odot 2}$ is used to denote the Hadamard power of $\boldsymbol{\sigma}$ (element-wise power). Lastly, to compute $p(\mathbf{X}_i | \mathbf{z}_i; \theta)$ it is normally assumed that $p(X|Z; \theta)$ is a factorised Gaussian or Bernoulli distribution, depending on the data type. The overall VAE computational flow is illustrated in Figure 2.4.

Chapter 3

Deep relational representations as transferable knowledge

The goal of this chapter is to set the scene and outline the motivations driving this thesis. At their core, these are towards building human-like learning systems. There are of course many components of human-level intelligence (Newcombe, 2013; Buckner, 2023; Marcus, 2001; Marcus and Davis, 2020; Goyal and Bengio, 2022), but the pivotal trait that this chapter focuses on is our ability to *continuously acquire, reuse, and update knowledge throughout life* Barnett and Ceci (2002); Piaget (2005); Lake et al. (2017). To frame an AI as a more human-like learner, both AI and humans operate in the real-world. This is important as it scopes the tasks an AI could be reasonably expected to perform, which Bengio and LeCun (2007) refer to as the *AI-set*, including tasks such as object perception, language understanding, and motor control. The question is then: what knowledge should an AI acquire that will be useful in the AI-set? One answer is *representation* (Bengio et al., 2013): since all experiences and tasks are grounded in a real-world environment, human-like AI can benefit from representing the world in a way that exposes the representational overlap between experiences and tasks. This is the intuition behind *representation learning* (Bengio et al., 2013), which is the process of learning a representation that is useful across the AI-set, under the assumption that different data is generated from an overlapping set of Explanatory Factors (EFs). Building on (Bengio et al., 2013), this thesis motivates the use of First-Order Logic (FOL)-like *relational representations* that are ‘bound’ to the underlying EFs (Greff et al., 2020). In combining the domain-general EFs with the domain-independence of FOL semantics, the hypothesis is that the resulting FOL-like relational

representation can produce knowledge units, which may be more amenable to transfer.

The present task is to then see if, a relational representation can be learned in a continual manner, starting from low-level stimuli, throughout the lifetime of an AI that operates in a real-world setting, and whether the resulting representation can improve lifelong transfer of knowledge between tasks. This chapter will firstly motivate the use of FOL-like relational representations, leveraging the domain-independence of FOL semantics. This will lead to a modular architectural design that aims to produce this type of representation, starting from low-level perceptive stimuli. Then, a lifelong learning inspired experimental setup will be introduced (Thrun and Mitchell, 1995), wherein a Deep Reinforcement Learning (DRL) agent is trained in a Reinforcement Learning (RL) setting in which the data distribution is gradually widened across task phases, but the reward-salient property is present throughout. The experiment will be conducted using the presented architecture, in which relational inductive biases will be incrementally incorporated so that the impact and limitations of each can be investigated. In doing so, the problem landscape for this thesis will be established, and the following chapters will build upon this foundation.

3.1 Relational abstractions towards reusable knowledge

For AI agents to approach human-like learning capabilities, they must be able to learn and reuse knowledge across tasks (Thrun, 1995; Thrun and Mitchell, 1995). To formalise this discussion, following Thrun (1995), let us frame the human-like AI learner as a lifelong learning robot embedded in the real-world, whose learning can be encapsulated by the following learning problem:

Definition 3.1 (Lifelong Reinforcement Learning) *The Lifelong Reinforcement Learning problem is formally defined as a set of control problems of the form:*

$$\{ \langle \mathcal{X}, \mathcal{A}, W, R_i \rangle \mapsto \pi_i \mid \pi_i : \mathcal{X} \mapsto \mathcal{A} \}$$

such that π_i maximises R_i over time.

where R_i denotes the reward-model for the i th control problem; and π_i is the action policy which

should maximise reward on the i th control problem over time (Thrun and Mitchell, 1995)¹.

The set of control problems denote the tasks that the robot will be expected to solve throughout its lifetime. These may be faced in any order, and a given task may be revisited multiple times. W determines the dynamics of the environment, which for our robot are the dynamics of the real-world. At a given time, t , we say that the world will be in a state $w_t \in \mathcal{W}$, where $\mathcal{W}$ is the set of all possible world-states. However, our robot will be unlikely to perceive the full world-state and will instead only observe what its situation presents and what its sensory capacity can perceive. For this example, let us assume that the robot perceives a visual data stream, represented as images and denoted by $\mathbf{X}_t \in \mathcal{X}$. When an agent takes an action (including the null-action of doing nothing) the state of the world will transition to some new state w_{t+1} according to the dynamics of W and a new observation $\mathbf{X}_{t+1}$ will be made available to the agent, forming a MDP (Section 2.3.3). As such Thrun and Mitchell (1995) take W to be a function of the form $W : \mathcal{W} \times \mathcal{A} \rightarrow \mathcal{W}$.

Importantly, in being associated with natural laws of the real-world, $\mathcal{W}$ will adhere to a certain structure (Higgins et al., 2019, 2022), and is thought that a key capacity of intelligent agents acting in the real-world is their ability to uncover and leverage this structure (Lake et al., 2017; Shanahan et al., 2020a; Goyal and Bengio, 2022). The existence of this structure in $\mathcal{X}$ can be framed under the assumption that $\mathcal{X}$ is generated from a set of underlying EFs. Such an agent, whose perceptions are limited to $\mathcal{X}$, must therefore learn to identify the relevant structure of $\mathcal{W}$ from $\mathcal{X}$ in order to learn to act effectively throughout its lifetime. In effect, the overlapping of relevant EFs between sets of perceptions $\mathcal{X}_i \subseteq \mathcal{X}$ presents an opportunity for a agent to identify regularities between perceptions and *disentangle* these EFs in its internal representation. In doing so, the agent can prepare itself for future perceptions, which may arise from a different set of EFs, but which may share some of the same EFs as those already identified (Bengio et al., 2013; Bengio and LeCun, 2007). Taking this one step further, a core idea in this thesis is that this structure can be, at least in part, represented using FOL. For instance, using FOL we can say

$$\forall A, B, C \text{ (hasProperty}(A, B) \rightarrow \text{isUsefulForTask}(A, C)) \quad (3.1)$$

which states that if an object A has a property B , then it is useful for all tasks. Adopting a FOL

¹This definition is more restrictive than the original definition of Thrun and Mitchell (1995) in that it strictly assumes a single environment W whereas the original definition allows for multiple task-specific environments.

representation is useful as its semantics are domain independent; in the way that Equation 3.1 is defined, it is true for any A , B and C that satisfy the relation definitions. Moreover, FOL representations are compositional, the value of which is best exemplified by natural language, in which a finite set of words can be combined to express an infinite amount of possible sentences (Platts, 1979). Similarly, in learning FOL-like relations provides an opportunity for domain-independent compositional reuse.

To reconcile FOL within the notion of EFs, the key observation is that the relations can be defined in terms of the underlying EF values. For instance, if we wanted to see if one object is bigger than another, we need to look at the scale EF of each object. In associating the relation with its relevant EFs, which must in turn be identified from perceptions, the agent forms a binding between relations and EFs (Greff et al., 2020; Roskies, 1999). This presents an opportunity: if an agent is able to learn relation-decodings via such a binding, then when faced with new experiences it may be better able to delineate new EFs from the existing ones. In this case, the agent may be better able to represent new EFs in a manner that does not interfere with its already established bindings, additionally enabling pre-existing EFs to be exploited with no, or minimal, changes.

For our robot agent to make use of FOL as a representation of knowledge, at a minimum it must be able to perform an *abstraction* from low-level sensory information into FOL-like constructs (Konidaris, 2019)². Let us qualitatively contrast the preference of $\mathcal{X}^\phi$ over $\mathcal{X}$ by considering the following example³.

Example 3.1 *Consider a task where at each timestep t , the agent is presented with two objects, each having an associated reward/penalty that depends on the object shape. The agent’s objective is to select the most rewarding object. For this example let the objects be either squares, hearts, or ovals, where hearts produce a positive reward, squares a penalty, and ovals no reward nor penalty. To theme the example more towards LRL, let us say that: for $t \in \{0, 1, \dots, T_1 - 1\}$ the agent is only shown medium sized objects, in a fixed position, but their rotation may vary; for*

²To benefit from the compositionality of FOL, we might also want to be able to manipulate these symbols into more complex statements, and we would also like to be able to reason over these statements, in order to make decisions about what actions to take. None of this is straightforward and in this chapter the main focus will be on the process of abstraction.

³It should be noted that this example is basic from the point of view of Definition 3.1, in that the agent a single reward model covers the entire task. In extending to a true Lifelong Reinforcement Learning (LRL) setting, the robot may later be expected to subsequently perform different tasks, for instance stacking the objects or clearing them away.

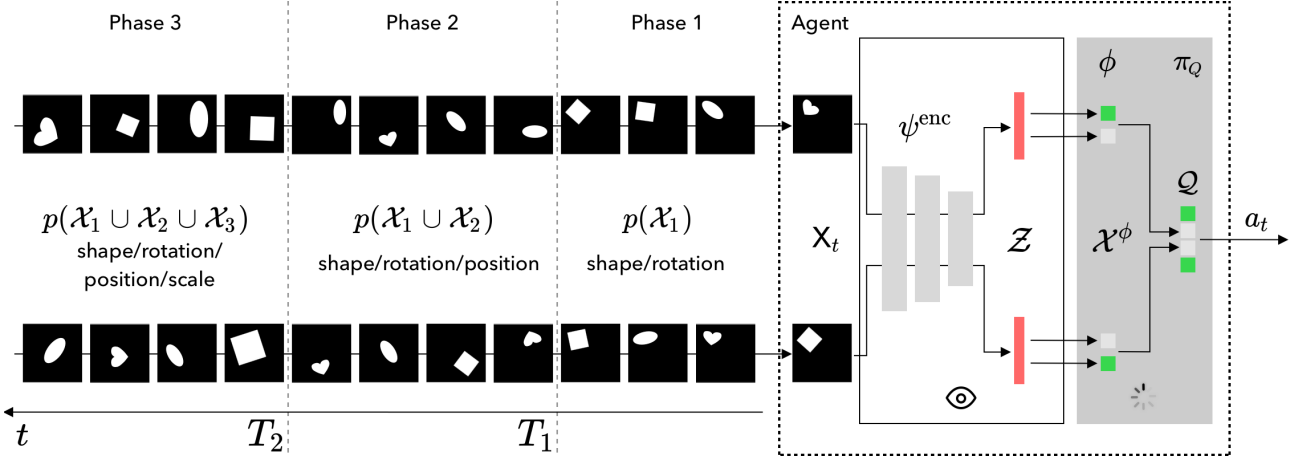


Figure 3.1: A depiction of how an agent may process the environment from Example 3.1, assuming it adopts a relational representation. Note that time is flowing from right to left and that in practice the agent will also receive rewards at each timestep (not shown here). The agent observes two objects at a time, each with an associated intrinsic reward, and encodes them into an internal representation $\mathcal{Z}$ via a shared encoder. These encodings are then passed through a set of relation-decoders to decode characteristics about each object, to produce an abstract representation $\mathcal{X}^\phi$. Here the upper arrow mimics a `isHeart` relation and the lower a `isSquare` relation. Finally, the agent attempts to take the most rewarding action according to $a_t \sim \pi(A|x_t^\phi)$, $x_t^\phi \in \mathcal{X}^\phi$. This process is repeated throughout the agent’s lifetime, during which the data distribution changes and new objects are introduced. A successful agent will learn to align new object representations to its internal relational representation, such that it can continue to identify the rewarding objects from the non-rewarding ones.

$t \in \{T_1, T_1 + 1, \dots, T_2 - 1\}$, the objects’ positions may vary; and for $t \in \{T_2, T_2 + 1, \dots, T_3 - 1\}$, the objects’ sizes may also vary.

Between timesteps $t \in \{0, 1, \dots, T_1 - 1\}$ a fresh agent (no prior experiences) would be forgiven for assuming that $\mathcal{X}$ is the space of all observations of two objects with fixed size and position. In learning to select the hearts, the agent learns to distinguish hearts from other objects, which in FOL may be expressed by the relation `isHeart`(A). Such a relation can be said to depend on a shape EF. We would want that, after T_1 the agent can cope with the new shape positions, and after T_2 the agent can cope with the new shape sizes, such that it can still identify the hearts from the squares/ovals with minimal interference in its reward acquisition. When working directly with $\mathcal{X}$, these observations may appear very different - each observation will be an *entangled* scene in which object boundaries and higher level properties may not be readily apparent. On the other hand, when abstracted to $\mathcal{X}^\phi$ the agent may be able to *disentangle* the shapes of each individual object as distinct relations and filter out the noise in the representation it uses to select actions. In which case, when the agent suddenly experiences a new shape size, a property that is fundamentally *invariant* to the shape identity and thus the reward, it should more easily cope with these new states by assimilating size into its representation. The task is divided into a base representation, a relational abstraction, and then a reward

prediction component, as shown by Figure 3.1. To acquire a FOL-like representation directly from low-level observations means learning the mapping $f : \mathcal{X} \rightarrow \mathcal{X}^\phi$. The following experiment will test whether an agent, with an architecture that incorporates various degrees of relational inductive bias, can learn such a mapping purely from reward, and whether this mapping can be maintained as the agent experiences new observations. The details of this inductive bias is described in more detail in the next section.

3.2 End-to-end Relational Reinforcement Learning

For $\mathcal{X}^\phi$ to be FOL-like it must carry FOL-like characteristics. In FOL, and more broadly in Logic, knowledge is represented in terms of entities⁴ and relations between them (Garnelo and Shanahan, 2019), which has been highlighted as a desirable property for DL agents (Battaglia et al., 2018). In being FOL-like, the relations should at a minimum be explicit and modular, but should also interact in a logically consistent way, where they can be expressed in terms of one another in a compositional manner (Garnelo and Shanahan, 2019). In the extreme, abstracting into a true FOL representation can be desirable. This is the basis of Relational Reinforcement Learning (RRL) (Dzeroski et al., 2001) where the environment is represented according to symbolic relational components, towards improved generalisation and planning. But in RRL states and actions must be expressed with relations, and the policy is learned using logical decision trees with carefully provided background knowledge. In practical scenarios, this is unscalable, as for even small sized environments (in terms of numbers of objects and relation symbols) exhaustively assigning each relation symbol grounding for each state becomes intractable (Thrun and Mitchell, 1995). Modern works that can be viewed as bringing RRL into the DRL era either require problem-specific heuristics (Garnelo et al., 2016), rely on pretrained relation-decoders (Badreddine and Spranger, 2019), or do not offer an explicit FOL-like representation and instead the relational representation is implicit to the architecture (Zambaldi et al., 2019). Though these may be considered FOL-like, a fully end-to-end implementation that produces explicit FOL-like relations that can be reasoned over is still missing. The following experiments will explore such an implementation, by incrementally incorporating relational inductive biases into a standard DRL end-to-end architecture.

⁴Normally referred to as objects in FOL literature, this should not be confused with the physical objects we had in Example 3.1.

Before outlining the architectures, let us first consider the relevant spaces and mappings between them, as these will help motivate and differentiate the architectures and their inductive biases. We will firstly assume that the agent observes its environment as set of low-level states $\mathbf{X} \in \mathcal{X}$, where each $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ is a (H)eight $\times$ (W)idth $\times$ (C)olour-Channels image tensor, and that each image is mapped into an intermediary latent representation via the mapping,

$$\psi^{\text{enc}} : \mathcal{X} \rightarrow \mathcal{Z} \quad (3.2)$$

where for a given image $\mathbf{X}$, ψ^{enc} produces a $\mathbf{z} \in \mathcal{Z}$ vector. The latent space is a lower dimensional representation of the input image, and is intended to capture the relevant information for the task at hand. From here, the latent representation is either mapped to $\mathcal{X}^\phi$ via the mapping,

$$\phi : \mathcal{Z} \rightarrow \mathcal{X}^\phi, \quad (3.3)$$

or may be directly mapped to a vector of Q -values, $\mathbf{q} \in \mathcal{Q}$, where $\dim(\mathcal{Q}) = |\mathcal{A}|$, so that the problem can be trained using Q-learning⁵. Abusing notation, π is used to refer to either mapping into $\mathcal{Q}$, depending on the context, *i.e.*

$$\pi : \mathcal{Z} \rightarrow \mathcal{Q} \quad \text{or} \quad \pi : \mathcal{X}^\phi \rightarrow \mathcal{Q}. \quad (3.4)$$

To test performance the action corresponding with the maximum Q -value is then selected. However, during training the action is randomly selected to encourage exploration. The overall mapping flow is depicted in Figure 3.3, assuming ϕ is used.

With the mappings defined, the three levels of relational inductive bias are as follows⁶:

CNN (baseline with no relational inductive bias): The baseline architecture uses a simple CNN model for ψ^{enc} and an MLP for π with ReLU activations. Here observations $\mathbf{X}_i \in \mathcal{X}$ are passed to ψ^{enc} as a single image to produce a $\mathbf{z}_i \in \mathbb{R}^{20}$ vector encoding. The single image is constructed by concatenating two images side-by-side, *i.e.* $\mathbf{X}_i = [\mathbf{X}_i^1 \mid \mathbf{X}_i^2]$ (width-wise concatenation). This is then passed to π to produce two outputs, one per each action Q -value.

EW-CNN (entity-wise inductive bias): The first level of relational inductive bias involves

⁵See Section 2.3.3 on Q-learning.

⁶Additional experimental details are provided in Appendix A.

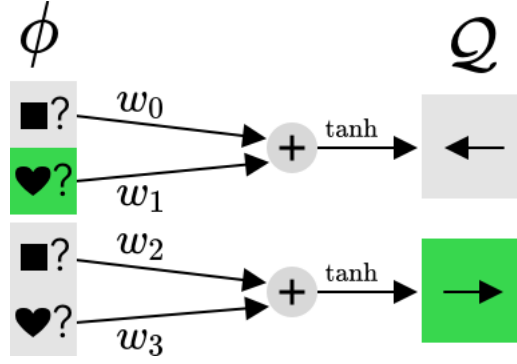


Figure 3.2: Illustration of π used in the EW-CNN- ϕ architecture, where $w_0, \dots, w_3$ are scalar weights. The policy is extremely simple, relying on the FOL-like representation obtained by ϕ . By using a tanh activation, Q -values are bound to $(-1,1)$ which is appropriate for the reward model used in these experiments, as rewards are either -1, 0, or +1.

the splitting up of each $\mathbf{X}_i \in \mathcal{X}$ into the individual constituent $\mathbf{X}_i^1$ and $\mathbf{X}_i^2$ image entities. A smaller shared CNN encoder is then applied to each entity image to produce two vectors $\mathbf{z}_i^1, \mathbf{z}_i^2 \in \mathbb{R}^{10}$. These are concatenated and passed forward to the same π function as the baseline model.

EW-CNN- ϕ (relational $\mathcal{X}^\phi$ layer): The second and final relational inductive bias is the inclusion of an explicit relational layer, $\mathcal{X}^\phi$. Here the encodings produced by the EW-CNN encoder are passed to a set of shared unary relation-decoders, ϕ_{r_1} and ϕ_{r_2} , which together perform the mapping ϕ . Each produces a single $(0,1)$ value, *i.e.*

$$\phi_r : \mathcal{Z} \rightarrow (0, 1). \quad (3.5)$$

and are implemented as Neural Tensor Network⁺ (NTN⁺) functions, which were used as FOL-like relation-decoders in (Donadello et al., 2017)⁷. The output of the relational layer is a fixed-order $(0, 1)^4$ vector (2 relation-decoders applied to 2 image entities, with a fixed order to preserve position). This is then passed through a simplified π to output the final Q -values. As shown by Figure 3.2, unlike the π of the baseline and EW-CNN models, this π applies a single layer MLP without biases and with tanh activations, that splits weights between each action Q -value, so that each Q -value is produced by looking only at the groundings of a specific entity position. The use of tanh activations is to ensure that the output of π is bounded between $(-1, 1)$, which is appropriate for the reward function used in these experiments as rewards are either -1, 0, or +1. Overall π is heavily constrained and biased - since no supervision is used to guide the relation-decoders towards classifying the desired

⁷NTN⁺ model details are provided in Appendix B.

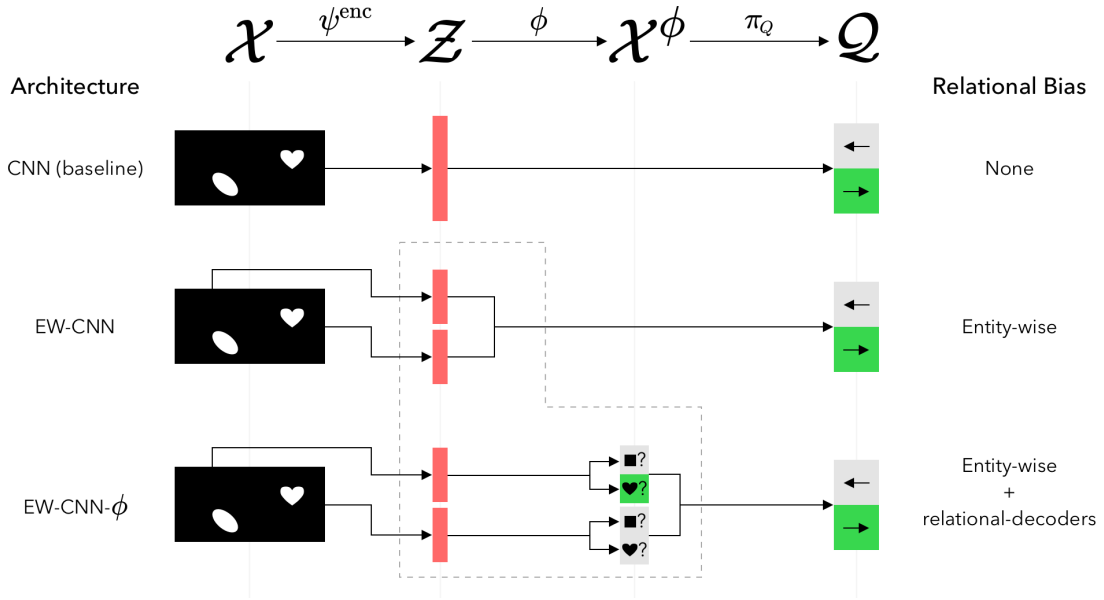


Figure 3.3: Breakdown of end-to-end DRL architectures that incorporate increasing degrees of relational inductive bias. The top row firstly provides the space mappings, where a low-level scene in $\mathcal{X}$ is mapped to an intermediary representation $\mathcal{Z}$ via ψ^{enc} , then to $\mathcal{X}^\phi$ via ϕ , and finally to Q -values in $\mathcal{Q}$ via π . In the greedy case, the action corresponding with the maximum Q -value is then selected. Alternatively, in the stochastic case, the Q -values are passed through a softmax function to produce a probability distribution over actions, from which an action is sampled. The bottom row depicts the architectures, where the baseline CNN does not incorporate any bias and omits usage of ϕ , instead directly mapping $\mathcal{Z}$ to $\mathcal{Q}$. EW-CNN incorporates an entity-wise bias, where the low-level scene is split into two entities, but again ϕ is omitted. In the final EW-CNN- ϕ version, the full degree of relational inductive bias is incorporated, where each entity latent representation is passed through a set of shared relation-decoders to produce a lower dimensional relational representation. These are then passed through π to produce Q -values. Since hearts produce reward and ovals do not, a successful agent should learn to select the heart for this timestep.

EFs, imposing architectural constraints can narrow the hypothesis space and improve the likelihood that relations are learned.

Each architecture (illustrated by Figure 3.3) is trained using Q-learning. For the relation-decoding architecture, the salient EF to decode is shape type, since this the only property that determines rewards. In which case, a desirable learning outcome could be learning $\phi_{\text{isHeart}}(\mathbf{z}_i^1) \approx 1$, where $\mathbf{z}_i^1 = \psi^{\text{enc}}(\mathbf{X}_i^1) \in \mathcal{Z}$, suggesting that the first entity image is a heart. However, there is no guarantee that the relation-decoders will learn to decode the desired shape EFs, as no direct supervision is used. Although there are three shapes, only two relation-decoders are used. From a FOL perspective the third shape can be identified by the absence of the other two shapes, *e.g.* via the rule:

$$\forall X \left((\text{isOval}(A) \leftrightarrow \neg (\text{isHeart}(A) \vee \text{isSquare}(A))) \right). \quad (3.6)$$

This can be facilitated by ϕ by *e.g.* $\phi_{\text{isHeart}}(\mathbf{z}_i^1) \approx 0$ and $\phi_{\text{isSquare}}(\mathbf{z}_i^1) \approx 0$, in which case the Q -value activation provides $\tanh(0) = 0$. In theory having three relational-decoders should be

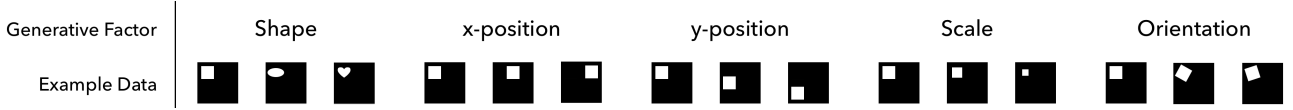


Figure 3.4: Example **dSprites** data for each EF that has more than one attribute value: shape, scale, orientation, x -position, and y -position.

suitable, but initial experimentation showed that including the additional relation-decoder in a fully unsupervised setting led to loss often reaching bad local minima, and not showing FOL-like outputs that converged to deterministic 0 or 1 outputs.

3.3 Experimental setup

In this section, Example 3.1 will be formalised into a test-bed experimental setup that will then be used to evaluate the impact of entity-wise and relational inductive biases towards knowledge transfer in an LRL-themed setting. Let $\mathcal{X} = \bigcup_{i=1}^3 \mathcal{X}_i$ be a dataset of images, where $\mathcal{X}_1 \subset \mathcal{X}_2 \subset \mathcal{X}_3$. The experimental procedure is split up into phases $\{P_1, P_2, P_3\}$, where at each phase, starting from P_1 , two images are uniformly sampled from either each corresponding domain, *e.g.* $\mathbf{X}_t^1, \mathbf{X}_t^2 \sim \mathcal{X}_i$ during phase P_i . Each phase lasts for 150k timesteps, where 100 evaluation timesteps are performed after every 250 timesteps of training. Following a similar approach taken in representation learning studies (Locatello et al., 2019), during evaluation time image samples are drawn from the same distribution as the training data. An extension to this work could include a distribution shift between training and evaluating, but the interest here is in seeing how well the agent can cope with a changing data distribution during the phases of training. In effect, in the evaluation it is desirable that the cause for possible performance failures is not due to a failure to generalise the representation learning component to a new data distribution, but rather due to a failure to adapt to the changing distribution that is fully exposed during training.

In this experiment the **dSprites** synthetic dataset (Matthey et al., 2017) is used, which comprises 737,280 procedurally generated $64 \times 64 \times 1$ greyscale images. Therefore there are $|\mathcal{X}| = 737280^2$ possible unique states that can be constructed. Matthey et al. (2017) constructed the **dSprites** dataset by generating one image for each configuration of 6 independent EFs: colour, shape, scale, orientation, x -position, and y -position. Respective numbers of EF values are provided in Table 3.1 and data examples are provided by Figure 3.4. From these EFs, three data subsets are

Table 3.1: Breakdown of the dSprites dataset EFs.

Explanatory Factor	# Values	Range/Description
Colour	1	Fixed white colour value.
Shape	3	Square, ellipse, or heart.
Scale	6	Linearly spaced in $[0.5, 1]$.
Orientation	40	Uniformly distributed in $[0, 2\pi]$.
X-Position	32	Uniformly distributed in $[0, 1]$.
Y-Position	32	Uniformly distributed in $[0, 1]$.

created in accordance with Example 3.1: $\mathcal{X}_1$ includes all orientation variations and shapes, but keeps all other properties fixed to a specific value; $\mathcal{X}_2$ includes $\mathcal{X}_1$ and furthermore lets x and y position vary; and $\mathcal{X}_3$ includes $\mathcal{X}_2$ and allows scale, the final property, to vary across its range. The corresponding cardinalities are $|\mathcal{X}_1| = 14400$, $|\mathcal{X}_2| = 1.5 \times 10^4$, and $|\mathcal{X}_3| = 5.4 \times 10^4$. A common hindrance for lifelong learning with DL is *catastrophic forgetting*, in which learning new information can cause the agent to forget previously learned information (McCloskey and Cohen, 1989). This is mitigated by including the previous data subset in the next one at each phase and ensuring the reward model, R , stays fixed. The reward model is

$$R(\mathbf{X}_i^j) = \begin{cases} 1 & \text{if } \mathbf{X}_i^j \text{ is a heart} \\ -1 & \text{if } \mathbf{X}_i^j \text{ is an square} \\ 0 & \text{if } \mathbf{X}_i^j \text{ is an oval} \end{cases}$$

where $\mathbf{X}_i^j$ is the j th image in the i th timestep. At each timestep and state the agent can take one of two actions, $\mathcal{A} = \{\leftarrow, \rightarrow\}$, corresponding to an image selection. The RL problem is structured as a 1-step episodic RL setting⁸ where the agent’s goal is to choose the most rewarding image at each timestep, based on the individual image’s internal properties. This is a heavily simplified RL problem that requires no future reward discounting and no locomotive decision making. Both are desirable as there are fewer hyperparameters to train, meaning evaluation can be more focused towards the effect of each relational inductive biases, and the agent does not need to learn to perform distance-based routing between rewarding objects. Moreover, the agent does need to balance exploration versus exploitation in order to learn a policy that maximises long-term reward. Instead, during training the agent randomly takes actions, and

⁸See Section 2.3.3 for RL background

during evaluation the agent chooses the action with the highest predicted reward so that we can assess the agent’s ability to learn the reward model. Since the training and evaluation data distributions are the same, the reported evaluation loss is a 100-step batched average of random actions. Taking the reward prediction loss from greedy action selection is less insightful, as the agent will learn to only pick the most rewarding object and as such the loss will be biased towards higher reward objects.

Since the reward model is fixed across phases and the data always exposes all shapes, the agent should in theory be able to learn to maximise reward in the first phase of learning. Moreover, the first phase only has 120 unique images to sample from, compared to 1.2×10^6 and 7.4×10^6 in the second and third phases, respectively, learning should be easier in the first phase. If FOL-like relations are indeed learned properly, this would mean that the ϕ to π can be established in the first phase, essentially learning a FOL-like π of the form “if $\mathbf{X}_i^1$ is a heart and $\mathbf{X}_i^2$ is a square then choose $\mathbf{X}_i^1$ ”, or “if $\mathbf{X}_i^1$ is a square and $\mathbf{X}_i^2$ is a oval then choose $\mathbf{X}_i^2$ ”, and so on. When new data is then introduced in the later phases, the agent should only need to align the new representations so they can be assimilated into the existing FOL-like representation, such that each shape type is correctly identified. In theory, ϕ and π can therefore be fixed after successful learning in the first phase, and the agent should only need to learn ψ^{enc} in the later phases. In the following, along with baselines, results will be given in both cases, where EW-CNN- ϕ (P123) denotes that the full EW-CNN- ϕ architecture is trained throughout each phase and EW-CNN- ϕ (P1) where ϕ and π are fixed after the first phase.

3.4 Results

In this section, results for the LRL-themed experiment are presented. During training, the agent is exposed to random pairs of images from each data subset, corresponding with the training phase. It is possible that ovals are shown next to ovals, or ovals with squares, and even sometimes squares next to squares. In these scenarios, the best the agent can do is to pick the least bad object in terms of reward. This means that looking at accumulated reward over the testing phase can be less insightful than knowing whether the agent picked the best available choice. Therefore, the following results show the *rank reward*, which is the percentage of timesteps in which the agent selected the best object.

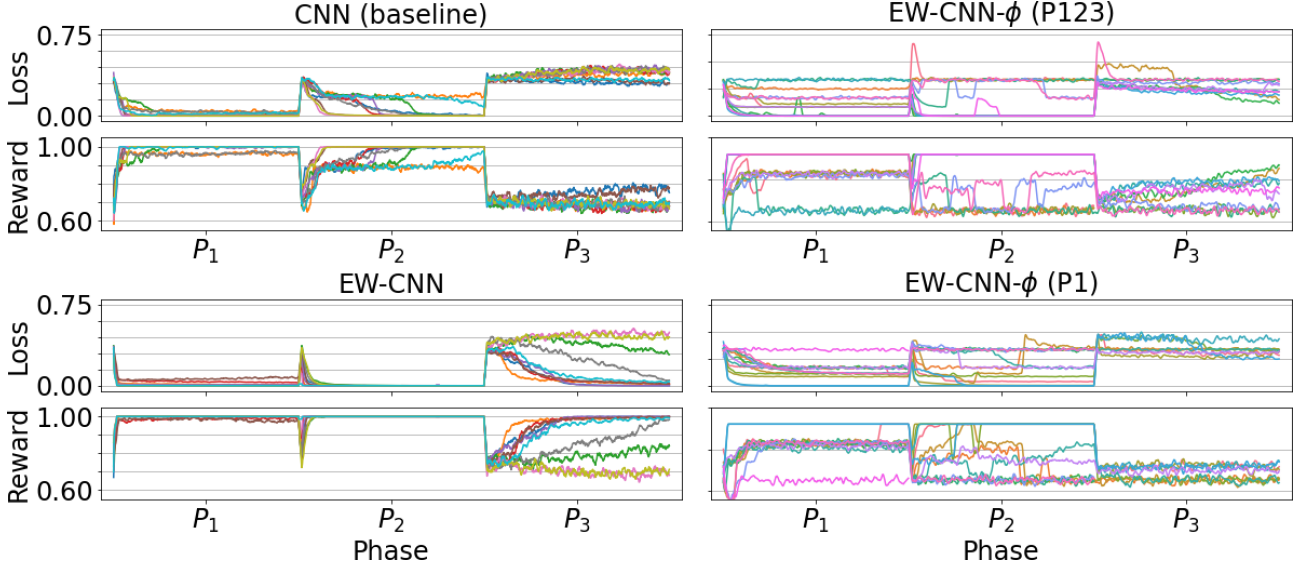


Figure 3.5: Evaluation L^1 reward classification loss (5 evaluation step moving average) and rank reward (10 evaluation step moving average) for each experimental setting described in Section 3.3. The x -axis shows each phase. The loss value gives the average difference between predicted action reward versus the true reward across 100 timesteps of evaluation, and the rank reward gives the percentage of timesteps where the agent selects the best object during the same period. Results include 10 random restarts for CNN (baseline) and EW-CNN, and 14 for each EW-CNN- ϕ setting. If a learning run fails to reach a rank reward greater than or equal to 0.9 in any phase, it is pruned and so not trained in later phases. Note that rank reward can be maximised before losses reach good values, since the agent only needs to identify the object with the better relative reward. Key observations are that the baseline case fails to optimise loss or rank reward in the final phase, and sometimes in the second. Adding in the entity-wise bias leads to robust optimisation in the second phase, and generally good performance in the final phase. On the other hand, results are generally far worse when incorporating relation-decoders. Both EW-CNN- ϕ (P123) and EW-CNN- ϕ (P1) can fail to converge to an optimal solution in the first and second phases. Final phase convergence is more apparent for EW-CNN- ϕ (P123), but is slow and generally unstable. As for EW-CNN- ϕ (P1), there is very slight indication that is able to converge but is even slower.

Figure 3.5 shows the L^1 loss and rank rewards averaged over 100 evaluation steps, throughout training for each experimental setting described in Section 3.3. Evaluations are performed every 250 timesteps, and training phases change every 150k timesteps. Firstly, in the baseline scenario, the agent fails to optimize both loss and rank reward in the final phase, with occasional difficulties also seen in the second phase. In contrast, adding in the entity-wise bias leads to robust optimisation in the second phase, and generally good performance in the final phase, where generally the rank reward is quickly maximised, or convergence towards optimal results is slower but still visible. Here, the rank reward is often quickly maximized, although in some instances, convergence to the optimal results is slower but still evident. However, it is noteworthy that in a few cases, the agent converges to a sub-optimal solution, albeit rarely. The incorporation of relation-decoders, interestingly, results in markedly worse outcomes, where evaluation performance is generally far less stable. In many instances, the agent settles on a sub-optimal solution in the initial phase. Since these are pruned, later phase results will

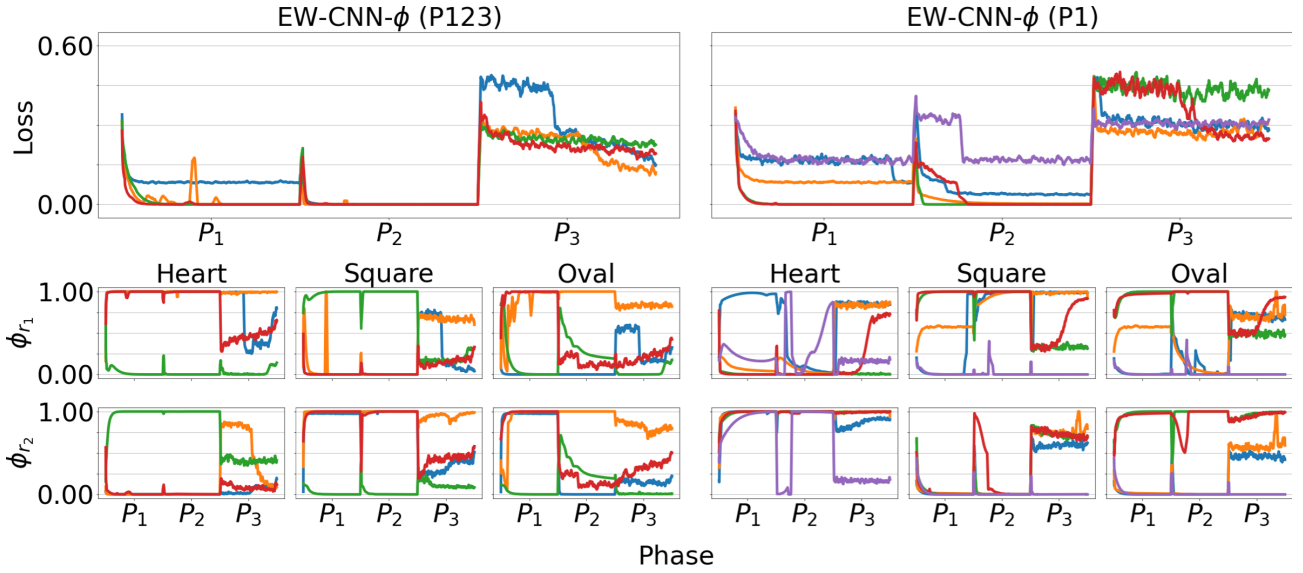


Figure 3.6: Illustration of evaluation performance for a collection of training runs. **(Top)** Evaluation L^1 reward classification loss (5 evaluation step moving average), as in Figure 3.5. **(Bottom)** Average evaluation ϕ_{r_1} and ϕ_{r_2} relation-decoder classifications for each object type (10 evaluation step moving average) for each EW-CNN- ϕ setting. The x -axes show each phase. EW-CNN- ϕ (P123) annotates the setting where the relation-decoder and policy are trained throughout, and EW-CNN- ϕ (P1) annotates the setting where the relation-decoder and policy are fixed after the first phase. Looking for instance at the red EW-CNN- ϕ (P123) plot, good losses during P_1 and P_2 phases coincides with a ϕ_{r_1} classification that is close to 1 for hearts and close to 0 for squares, and vice versa for ϕ_{r_2} classifications. However, transitioning from P_1 to P_2 , both ϕ_{r_1} and ϕ_{r_2} oval classifications go from close to 1 to closer to 0. When entering P_3 ϕ_{r_1} heart classifications and ϕ_{r_2} square classifications drop. Similar events can be seen for other plots. Overall this indicates training instability, where shape EF representations are not being maintained throughout training.

only show runs which obtained at least 0.9 rank reward in the previous phase. Despite this, performance often deteriorates and fails to reach good levels in subsequent phases. Comparing the two EW-CNN- ϕ settings, we can see that neither maximises rank reward in the final phase. However, EW-CNN- ϕ (P123) does trend towards a good outcome, but this is slow and generally unstable. As for the EW-CNN- ϕ (P1) setting, there is very slight indication that the agent is able to converge to an optimal solution, but requires even longer.

Next, Figure 3.6 presents results that explore the underlying relation-decoder classifications to try and better understand the behaviour of each EW-CNN- ϕ setting. Good loss performance coincides with relation-decoders that resemble FOL-like relation behaviour, in which they produce mutually exclusive classifications. For instance, good performance is observed up to P_3 for the green EW-CNN- ϕ (P123) plot, where ϕ_{r_1} behaves like an `isSquare` relation and ϕ_{r_2} like an `isHeart` relation. For EW-CNN- ϕ (P123), the better runs are given by the green and red plots, where ϕ_{r_1} resembles an `isNotHeart` relation and ϕ_{r_2} an `isNotSquare` relation. However, degradation generally occurs in the final phase, where classification strengths deteriorate to the point where it is not clear what each relation-decoder is classifying. Results are generally less

stable for the EW-CNN- ϕ (P_1) case, indicating that ψ^{enc} is unable to align representations to the fixed relation-decoder decision boundaries after P_1 .

3.5 Discussion

The results in the previous section firstly demonstrate that even in a simple RL setting, where the agent is only required to learn to identify the reward-salient property and faced with two images at each timestep, the agent is unable to learn to maximise reward in the final phase when using the baseline standard CNN architecture. However, entity-wise separation leads to a significant performance gain, where the agent is then able to learn to maximise reward in the final phase. Naturally this makes sense, since success in this task relies on differentiating objects associating Q -values with each. This supports the notion that object-ness is an important inductive bias for intelligent agents (Goyal and Bengio, 2022; Lake et al., 2017).

Surprisingly, results are less stable when using the relation-decoder architecture. Firstly, freezing the relation-decoder and policy layers after the first phase generally shows worse performance. There are slight indications that convergence to a low loss can happen eventually. However, this is far from the desired performance gains that could be expected from a FOL perspective. Even if relation-decoders do resemble the expected FOL-like relation behaviour in P_1 , as in the case of the red and green EW-CNN- ϕ (P_1) plots, this does not lead to a faster return to solving the RL task in subsequent phases.

An interesting property, and possible area for improvement in the architecture, is the freedom that the relation-decoders have in their classification of ovals. Since ovals have no reward, and given the simplicity of the utilized policy (as illustrated by Figure 3.2), as long as both relation-decoders produce the same classification value for an oval, and assuming that the policy learns an opposite sign weight for ϕ_{r_1} and ϕ_{r_2} 's output, respectively, then the agent will be able to learn to maximise reward. This is observed in the red EW-CNN- ϕ (P_{123}) plot, where during P_2 the oval classification flips from close to 1, to closer to 0 for both relation-decoders, without deteriorating the overall loss. This makes the overall relation-decoder behaviours less FOL-like - for instance where in P_1 of the same plot, ϕ_{r_1} resembles an **isNotHeart** relation and ϕ_{r_2} an **isNotSquare** relation, in P_2 they ϕ_{r_1} instead resembles an **isHeart** relation and ϕ_{r_2} an **isSquare** relation. In P_2 this does not affect the overall loss, but if we are to say the relation-decoders are

FOL-like then such a change in behaviour is not desirable.

In terms of demonstrating the value of a relational representation, the results in this chapter firstly demonstrate that taking an entity-wise inductive bias for relational tasks is a clear win, over a naïve end-to-end CNN approach, represented by EW-CNN. Results also show that relation-decoders are able to learn to mimic FOL-like relation behaviour when trained in an end-to-end and unsupervised, where they produce mutually exclusive classifications for given shape types. However, contrary to expectations, maintaining a fixed relationship between the relation-decoder (ϕ) and policy mapping (π) after the initial phase, assuming that the abstract relational reward model is successfully learned, actually leads to a decline in performance. This degradation comes from the mis-alignment between ψ^{enc} and ϕ , where ψ^{enc} is unable to align representations to the fixed relation-decoders. It is less surprising that this is the case in P_3 , as CNNs are themselves inductively biased towards spatial translation invariance (as discussed in Section 2.2.2) but they are not invariant to scale change. However, despite this translational invariance, the results in P_2 show that ψ^{enc} is not always able to align representations to the fixed relation-decoders. It is possible that this is because the decision boundaries learned by each relation-decoder are too unforgiving (perhaps through overfitting the much narrower P_1 data-distribution) and in freezing their weights alignment is left to ψ^{enc} alone. Nonetheless, to further investigate the FOL-like properties of relation-decoders coupled with lower-level representation learning components and the transferability of the resulting relational representations, the next chapter will explore the coupling of relation-decoders with a greedier AE-based representation learner. In contrast to the architecture employed in this architecture, where reward is all that matters and irrelevant EFs can be ignored, the next chapter will aim to learn a complete data representation, in which the latent representations are encouraged to retain all input information whilst encoding the EFs in a manner that is amenable to relation-decoding.

Chapter 4

Improving representation learning with relation-decoders

In Chapter 3, the pivotal capability for an agent to perform well at the LRL task is its ability to delineate shape from irrelevant Explanatory Factors (EFs), and to then cope with being exposed to new ‘nuisance’ EFs which are irrelevant to the task, such as position and scale. From a FOL perspective, if unary shape classification were successful, future performance should be maintainable providing new observations do not confuse the agent’s ‘understanding’ of the shape factor. The fact that this was not possible in the fully relational case, where instead significant training instability was observed, suggests that the agent was not able to align the new information to its already established shape relation-decoding and corresponding policy, which had already solved the task in the first phase of training. Moreover, what if the additional properties of the data are not nuisance EFs (Achille and Soatto, 2018) when considering possible future tasks (Achille et al., 2018)? For instance, if the agent had to also stack objects, then the position and scale EFs would be salient. This motivates a closer look at the representation learning process, where we would want the representation to greedily represent each EF, in a manner that supports both current and future task performance. *Disentanglement*¹ has been demonstrated to offer such a representation and indeed improve transfer (Bengio et al., 2013; Higgins et al., 2019; Eddahmani et al., 2023; Locatello et al., 2019; Higgins et al., 2017a,b; Tschannen et al., 2018; Steenbrugge et al., 2018; van Steenkiste et al., 2019; Dittadi et al., 2021), even in lifelong setting due to their compositional structure (Achille et al., 2018).

¹See Section 2.4.1 for background on disentanglement.

From an LRL perspective, a pairing between relation-decoders and disentangled representation learning is desirable: whilst supervision over relation-decoders can target task-specific EFs, the backbone disentangled representation learning task can encourage the agent to ‘greedily’ capture the possibly future relevant EFs in composite² representations. This chapter investigates the compatibility of incorporating relation-decoders into a disentangled representation learning backbone as a form of weak relational supervision. Departing from the full LRL setting, a series of scoped transfer experiments will be conducted in order to explore the quality of the resulting relational representation. In particular, results will explore the basic FOL-like behaviour of the representation and will show that including relational semantic bias can both improve learning disentangled representations over multiple EFs, but also be *necessary* in situations where disentanglement alone is not enough.

4.1 Disentanglement and relational representations

The standard approach to obtaining a disentangled representation involves unsupervised generative modelling of the data distribution, typically following one of two approaches: Generative Adversarial Network (GAN) based methods (Goodfellow et al., 2014; Chen et al., 2016; Radford et al., 2016; Jeon et al., 2021) and VAE based methods (Kingma and Welling, 2014; Rezende et al., 2014; Higgins et al., 2017a; Burgess et al., 2017; Chen et al., 2018; Kumar et al., 2018; Kim and Mnih, 2018; Shao et al., 2020; Eddahmani et al., 2023)³. However, choosing a fully unsupervised approach can have shortcomings. In particular Locatello et al. (2019) proved that unsupervised disentanglement is impossible without inductive biases and advocated for the crucial role that supervision can play in obtaining disentangled representations. Moreover, Träuble et al. (2021) showed that unsupervised disentanglement is susceptible to spurious correlations and Higgins et al. (2017a) showed that “data continuity” is also critically important, in which dense variation of the possible EFs values is required in the data, in order to obtain disentanglement. To combat this, a trend of weak supervision⁴ has emerged, where the backbone unsupervised disentanglement task is supplemented with partial data labelling (Kingma et al., 2014; Narayanaswamy et al., 2017; Locatello et al., 2020b; Khemakhem et al.,

²Composite here meaning that each EF is captured independently of others, so that they can be combined to describe a combinatorial number of possible entities.

³Background for VAEs, which are employed throughout the remainder of this thesis, is provided in Section 2.4.2.

⁴Background on weak supervision is provided in Section 2.2.4.

2020; Locatello et al., 2020a; Hosoya, 2019; Bouchacourt et al., 2018; Chen and Batmanghelich, 2020, 2019; Karaletsos et al., 2016; Dittadi et al., 2021). Building on this trend, this section will motivate the use of FOL-like relation-decoding as a form of weak supervision over the backbone unsupervised disentanglement task. The general strategy is to train the VAE to reconstruct the input space $\mathcal{X}$ from the latent space $\mathcal{Z}$, and, using an auxiliary decoding objective, to train the relation-decoders to predict the FOL-like relations on or between latent representations of input objects (Chen and Batmanghelich, 2020, 2019; Karaletsos et al., 2016). The intuition for this coupling relies on the compositionality of EFs, and how this relates to the relevant EFs for a given relationship on/between objects. Following Higgins et al. (2019), let us assume there to be a *disentangled* transformation, which, in being disentangled, decomposes into symmetry transformations $\mathcal{G} = \mathcal{G}_1 \times \mathcal{G}_2 \times \dots \times \mathcal{G}_n$, where each $\mathcal{G}_i$ corresponds to an EF ν_i . Let us also assume that $\mathcal{Z}$ is a vector space that is similarly disentangled and decomposed into $\mathcal{Z} = \mathcal{Z}_1 \oplus \mathcal{Z}_2 \oplus \dots \oplus \mathcal{Z}_n$ where each $\mathcal{Z}_i$, $i \in \{1, 2, \dots, n\}$, component is a 1-dimensional axis aligned (and so orthogonal) vector subspace, and $\oplus$ denotes the direct sum of vector spaces. Then we can define an n -ary relation-decoder ϕ_r such that all of the following holds:

$$\begin{aligned}
 \phi_r(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n) &= \phi_r(g \cdot_{\alpha}^i \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n) \\
 &= \phi_r(\mathbf{z}_1, g \cdot_{\alpha}^i \mathbf{z}_2, \dots, \mathbf{z}_n) \\
 &\dots \\
 &= \phi_r(\mathbf{z}_1, \mathbf{z}_2, \dots, g \cdot_{\alpha}^i \mathbf{z}_n),
 \end{aligned} \tag{4.1}$$

where $g \in \mathcal{G}_i$ and $\cdot_{\alpha}^i$ is a subgroup action as provided in Section 2.4.1. Equation 4.1 tells us that the output of ϕ_r is symmetrical/invariant to the symmetry transformation g applied to its arguments, *i.e.* transforming any input argument using g will not change the output of ϕ_r . As an illustrative example, we could take g to be a “shift x -position by 1” transformation, and so $\nu_i = \text{position}$. In which case ϕ_r could be a relation-decoder that successfully approximates the FOL `isAbove`(A, B) relation, where A and B are FOL variables. Given our above assumptions about $\mathcal{Z}$, we can similarly take the i th component of $\mathcal{Z}$ to encode x -position and the j th component to encode y -position. Then, with subgroup action $+$, we can define $\mathbf{g} = [g_1, g_2, \dots, g_n]$ to be a vector where $g_i = 1$ and $g_j = 0$ for all $j \neq i$ (*i.e.* a one-hot vector where only the i th dimension is 1 and all others are 0). In this case $\mathbf{g} \cdot_{\alpha}^i \mathbf{z}_1 = \mathbf{z}_1 + \mathbf{g}$ and only the i th dimension of $\mathbf{z}_1$ is incremented. It is clear that transforming $\mathbf{z}_1$ in this way should not change the output of ϕ_r , since the

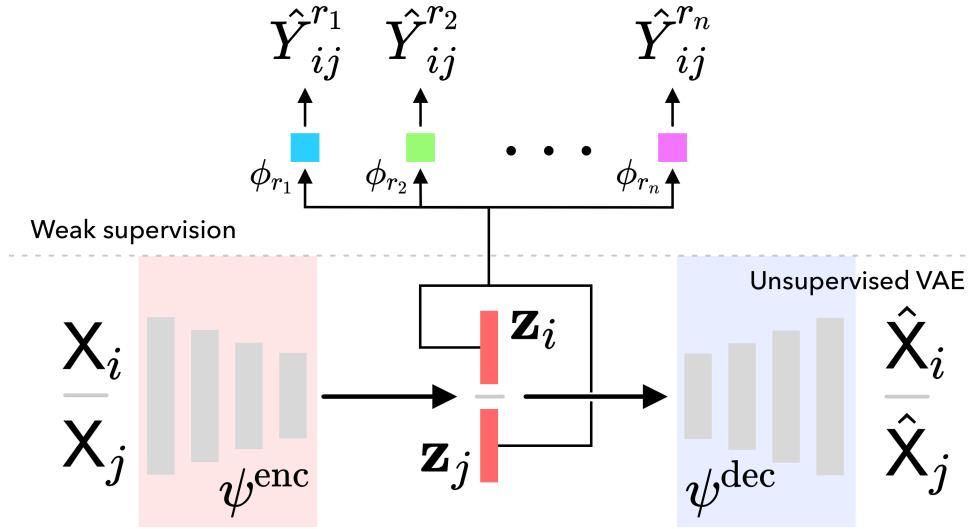


Figure 4.1: VAE and relation-decoder combined architecture. Observable (image) data $\mathbf{X}_i, \mathbf{X}_j \in \mathcal{X}$ are passed through a shared VAE, comprised of ψ^{enc} and ψ^{dec} , using the unsupervised ELBO objective. Their latent encodings $z_i, z_j \in \mathcal{Z}$ are used to reconstruct the inputs, giving $\hat{\mathbf{X}}_i, \hat{\mathbf{X}}_j \in \mathcal{X}$, but are also passed through additional shared relation-decoders ϕ_{r_k} to produce relational decodings $\hat{Y}_{ij}^{r_k}$ for each $r_k \in \sigma$, which are trained using $\mathcal{L}^\sigma(\mathcal{Y})$. In practice, not all relations will be trained on each $\mathbf{X}_i, \mathbf{X}_j \in \mathcal{X}$ pair, as each set of relation labels $\mathbf{Y}^{r_k}$ is sparsely populated.

x -position is not relevant to the truth of the `isAbove` relation. This leads a promising pairing between relation-decoder learning together with disentangled representation learning, where an appropriate choice of relations to approximate could aid in obtaining a disentangled latent representation. Moreover, as in Chapter 3, if the learned relation-decoders are truly FOL-like, then we should expect the resulting representation to be more transferable, either (or both) at the representational level, which will be examined in the following, or at the relation-decoder level, which will be explored in Chapter 6.

4.2 A straightforward relational representation learning architecture

As discussed in the previous section, numerous relations can be defined over a single EF, and, in being invariant to other EFs, including relation-decoder learning alongside a backbone disentangled representation task may help induce disentanglement over the relevant EFs for the relation that each relation-decoder approximates. In the following experiments, the disentangled representation learning element will be facilitated by a VAE, which is a popular choice for disentangled representation learning in the literature and are relatively straightforward to

train in comparison with GANs (Goodfellow, 2017; Nie and Patel, 2019). The weak relational supervision will be provided by a set of relation-decoders, each of which will be trained to approximate a corresponding FOL-like relation. Building on Chapter 3, each experiment will use NTN⁺ relation-decoders⁵. The overall architecture is shown in Figure 4.1. Given the use of multiple relation-decoders that are trained jointly, the architecture can be considered to be a MTL architecture (Goodfellow et al., 2016; Caruana, 1997). A similar architecture was utilized in (Chen and Batmanghelich, 2020), but only incorporated a single relation-decoder with a hard-coded latent subspace domain. In contrast, the architecture in this chapter is more general, allowing for multiple relation-decoders which can define their input domains through training. Additionally Chen and Batmanghelich (2020) did not use NTN⁺ relation-decoders and instead used a simple function specifically suited to similarity relations, whereas the NTN⁺ relation-decoders are more expressive and have been shown to model a variety of relations (Donadello et al., 2017; Socher et al., 2013).

In the general n -ary case, for each relation-decoder ϕ_r corresponding to $r \in \sigma$, where σ is a signature of relations as defined in Section 2.1.2, we can define an n -dimensional tensor $\mathbf{Y}^r \in \{0, 1\}^{|\mathcal{X}| \times \dots \times |\mathcal{X}|}$ of ground-truth relation labels which can be used to train the relation-decoder. Here, the entry $Y_{i_1 i_2 \dots i_n}^r$ is 1 if the relation r holds for the ordered n -tuple $(i_1, i_2, \dots, i_n)$ of data instances, and 0 otherwise. However, when $|\mathcal{X}|$ is large combinatorial explosion means it is impractical to train a relation-decoder for every possible relation label. Instead, the tensor $\mathbf{Y}^r$ will be sparsely populated, where $\mathcal{I}^r$ denotes the set of populated n -tuple indices in $\mathbf{Y}^r$, according to a sparsity rate, δ , which provides the ratio $\frac{|\mathcal{I}^r|}{|\mathcal{X}|^n}$, *i.e.* the ratio of available labels versus all possible labels. To train the combined relational representation learning architecture, the VAE will be trained using the unsupervised ELBO objective $\mathcal{L}^{\text{ELBO}}$ described in Section 2.4.2, and the relation-decoders using an aggregate loss $\mathcal{L}^\sigma$ over all relations in σ and their available relation labels. The joint relational representation learning objective is formulated as:

$$\mathcal{L}^{\text{joint}}(\mathcal{X}, \mathcal{Y}; \psi^{\text{enc}}, \psi^{\text{dec}}, \phi) = \mathcal{L}^{\text{ELBO}}(\mathcal{X}; \psi^{\text{enc}}, \psi^{\text{dec}}) + \lambda \mathcal{L}^\sigma(\mathcal{X}, \mathcal{Y}, \mathcal{I}; \psi^{\text{enc}}, \phi) \quad (4.2)$$

where $\mathcal{Y} = \{ \mathbf{Y}^r \mid r \in \sigma \}$, $\mathcal{I} = \{ \mathcal{I}^r \mid r \in \sigma \}$, and $\lambda \in \mathbb{R}^+$ is a positive scaling factor that balances the contribution of the two terms. Though Equation 4.2 specifies the general objective, each experiment will explore different choices of VAE, $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}^\sigma$ aggregation methods.

⁵NTN⁺ implementation details are provided in Appendix B.

4.3 Experiments

In the following, two sets of experiments will be introduced, each split into two phases: a source representation learning phase (source phase), and a target transfer phase (target phase). All experiments employ the architecture described in the previous Section 4.2 for the source phase, but differ in their downstream transfer tasks. Source phase analysis will focus on the compatibility of combining multiple relation-decoders in conjunction with an unsupervised disentangled representation learning backbone, in terms of the resulting source phase relation-decoding accuracy and disentanglement performance. Using different target phase tasks, experiments will then assess the impact of including relation-decoders under two transfer scenarios. The first compares the downstream learnability of a set of relations that are exact negations of the relations approximated during the source task, against their learnability when no weak relational supervision is included in the source task (*i.e.* vanilla VAE training). The second explores the possible transfer benefits of relational representation learning from a different angle, in which the transferability of the learned representation will be assessed on a downstream abstract visual reasoning task that relies on arithmetic properties of handwritten digits.

4.3.1 Relation-decoding transfer and disentangled representations

In this experiment, two signatures of relations are defined with respect to the `dSprites` dataset EFs, which were outlined in Table 3.1. Relation-decoders that approximate the first signature of relations are used to weakly supervise a VAE during the source phase of the experiment. In the task phase, the VAE is frozen and a new set of relation-decoders are trained to approximate the second signature of relations. The first signature of relations are:

$$\sigma^+ = \{\text{isLeft}(A, B), \text{isAbove}(A, B), \text{isBiggerThan}(A, B), \text{isSameShape}(A, B)\} \quad (4.3)$$

with FOL variables A and B . The second signature of relations is then constructed by defining the negation of each relation in σ^+ as a new relation:

$$\sigma^- = \{\text{isNotLeft}(A, B), \text{isNotAbove}(A, B), \text{isNotBiggerThan}(A, B), \text{isNotSameShape}(A, B)\}. \quad (4.4)$$

At first glance this may seem to be a trivial task, as if the learned relation-decoders are accurate, then approximations for the second signature can be obtained by simply taking the negation of the approximations of the first signature. For instance, taking $\phi_{\text{isNotRight}}(\mathbf{z}_i, \mathbf{z}_j) = 1 - \phi_{\text{isRight}}(\mathbf{z}_i, \mathbf{z}_j)$. The question of interest is whether this semantic is readily apparent when only having access to the latent representations $\mathbf{z}_i, \mathbf{z}_j$. Put differently, *is the semantics of the relations captured in the learnt latent space?* To evaluate this question, the experimental setup will consider a range of weak supervision scalar values $\lambda \in \{0, 25, 100, 200, 300, 400\}$, where for the baseline case $\lambda = 0$ the VAE is trained without weak relational supervision and so all relation-decoders are trained as a downstream task. Lastly, as an additional point of interest, the final part of the experiment will consider an additional signature of relations in the target phase, composed of unary relations over the shape EF:

$$\sigma^u = \{\text{isOval}(A), \text{isHeart}(A), \text{isSquare}(A)\}. \quad (4.5)$$

In contrast to σ^- , though σ^u contains relation-symbols which have a relationship with those in σ^+ , for instance, through sentences such as $\forall A, B, (\text{isOval}(A) \wedge \text{isOval}(B) \rightarrow \text{isSameShape}(A, B))$, their logical associations to σ^+ are more complex than simple negations and so may be more difficult to approximate from the latent space.

The VAE backbone used in these experiments is a variant called the β -TCVAE (Chen et al., 2018), which decomposes the D_{KL} term in the ELBO (Equation 2.35), under the assumption that $p_Z(\mathbf{z})$ is factorised, to expose additional independently tunable parameters. The resulting ELBO is:

$$\begin{aligned} \mathcal{L}_{\beta\text{-TCVAE}}^{\text{ELBO}} = & \mathbb{E}_{\mathbf{z}_i \sim q(\mathbf{Z}|\mathbf{x}_i)} [\log p(\mathbf{X}_i|\mathbf{z}_i)] & (4.6) \\ & - \alpha I(\mathbf{Z}; \mathbf{X}) & (\text{Index-Code MI}) \\ & - \beta D_{\text{KL}} \left[q(\mathbf{z}_i) \parallel \prod_j q(z_{i,j}) \right] & (\text{Total Correlation}) \\ & - \gamma \sum_j D_{\text{KL}} \left[q(z_{i,j}) \parallel p(z_{i,j}) \right] & (\text{Dimension-wise KL}) \end{aligned}$$

where $\alpha, \beta, \gamma \in \mathbb{R}^+$ are scalar weightings and the distributions q and p are computed by encoder ψ^{enc} and decoder ψ^{dec} networks respectively. When using the β -TCVAE, $\mathcal{L}^{\text{ELBO}}$ in Equation 4.2 is taken to be $\mathcal{L}_{\beta\text{-TCVAE}}^{\text{ELBO}}$. In decomposing the D_{KL} ELBO term, three components are exposed:

the Index-Code Mutual Information, which gives the mutual information between latent random vector $\mathbf{Z}$ and observable random (image) tensor $\mathbf{X}$; the Total Correlation, a generalisation of mutual information to more than two random variables, which measures the statistical dependence between the marginal distributions of the components of $\mathbf{Z}$ and will only be zero when the components are independent; and finally the Dimension-wise Kullback-Leibler (KL), which provides a penalty if individual components deviate from their corresponding priors. An initial hyperparameter tuning phase was conducted to determine appropriate $\mathcal{L}_{\beta\text{-TCVAE}}^{\text{ELBO}}$ parameters for the following experiments, which were found to be $\alpha = 0$, $\beta = 4$ and $\gamma = 0$. With these parameters it was possible to achieve reliably strong disentanglement, whilst achieving a reasonable reconstruction quality, which suggests input information is being retained in the latent representation. To measure disentanglement strength, the Mutual Information Gap (MIG) score from (Chen et al., 2018) is used, which measures the mean difference between the first and second top-most informative latent dimensions with respect to a given ground truth EF. For K EFs, the MIG score is computed as follows:

$$\frac{1}{K} \sum_{k=1}^K \frac{1}{H(\nu_k)} (I(\mathbf{Z}_{j_k}; \nu_k) - \max_{j \neq j_k} I(\mathbf{Z}_j; \nu_k))$$

where ν_k represents a random variable associated with the k th EF, $\mathbf{Z}_j$ is a random variable associated with the j th component of $\mathbf{Z}$, and $j_k = \arg \max_j I(\mathbf{Z}_j; \nu_k)$ is the index of the component that has the highest mutual information with the k th EF. Typically, when applying the MIG score, the components of $\mathbf{Z}$ are taken to be single dimensions, in which case it measures axis-aligned disentanglement, but in principle they could be higher dimensional vector subspaces. In the following experiments, mutual information will be calculated on a per dimension basis. Lastly, to train the relation-decoders, the following aggregation is used:

$$\mathcal{L}^\sigma(\mathcal{X}, \mathcal{Y}, \mathcal{I}; \psi^{\text{enc}}, \phi) = -\frac{1}{|\sigma|} \sum_{r \in \sigma} l(\mathbf{Y}^r, \mathcal{I}^r; \psi^{\text{enc}}, \phi_r) \quad (4.7)$$

where

$$l(\mathbf{Y}^r, \mathcal{I}^r; \psi^{\text{enc}}, \phi_r) = \frac{|\mathcal{I}^r|}{\sum_{(i_1, i_2, \dots, i_n) \in \mathcal{I}^r} \frac{1}{\text{Ber}(\mathbf{Y}_{i_1 i_2 \dots i_n}^r; \hat{\mathbf{Y}}_{i_1 i_2 \dots i_n}^r)}} \quad (4.8)$$

specifies the loss for an n -ary relation. Here $\text{Ber}(\mathbf{Y}_{i_1 i_2 \dots i_n}^r; \hat{\mathbf{Y}}_{i_1 i_2 \dots i_n}^r)$ is a Bernoulli distribution with

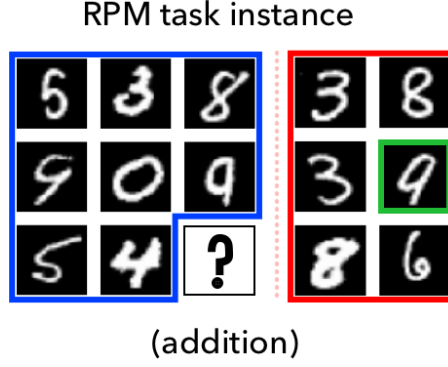


Figure 4.2: An example arithmetic RPM instance: the blue panel digits give the context that a learner must use to select an answer panel (red set) that replaces the ? panel. The arithmetic RPM task is either row-wise subtraction or addition. In the illustrated case it is addition and so the correct answer (green panel) is the digit 9.

parameter $\hat{Y}_{i_1 i_2 \dots i_n}^r$, which is the estimated relation label for the tuple $(i_1, i_2, \dots, i_n)$, and $Y_{i_1 i_2 \dots i_n}^r$ is the true relation label. The estimated relation labels are produced by relation-decoders, where for instance $\hat{Y}_{i_1 i_2 \dots i_n}^r = \phi_r(\mathbf{z}_{i_1}, \mathbf{z}_{i_2}, \dots, \mathbf{z}_{i_n})$. Equation 4.8 uses the harmonic mean to aggregate the loss over the populated indices following (Donadello et al., 2017). The harmonic mean is chosen over a standard arithmetic mean as it disproportionately penalises more incorrect predictions. Additional experimental details are provided in Appendix A.

4.3.2 Abstract visual reasoning over non-visual semantics

In this experiment the transferability of disentangled representations is explored in a different setting. In the source phase, the VAE component used in these experiments is a β -VAE, which is again trained with or without weak relational supervision. The β -VAE objective is similar to that of the β -TCVAE, but less complex, as it does not decompose the D_{KL} term in the ELBO, resulting in the following ELBO:

$$\mathcal{L}_{\beta\text{-VAE}}^{\text{ELBO}} = \mathbb{E}_{\mathbf{z}_i \sim q(\mathbf{z}|\mathbf{x}_i)} [\log p(\mathbf{X}_i|\mathbf{z}_i)] - \beta D_{\text{KL}} [q(\mathbf{z}_i) \parallel p(\mathbf{z}_i)]. \quad (4.9)$$

where again q and p are computed by ψ^{enc} and ψ^{dec} , respectively. In this experiment β and λ hyperparameter settings are explored in the range $\beta \in \{4, 8, 12\}$ and $\lambda \in \{0, 1000\}$, where $\lambda = 0$ means that the β -VAE is trained without weak relational supervision.

The source task training procedure is equivalent that described in Section 4.3.1, but the target task is now a Raven’s Progressive Matrices (RPM) task. This is following work by van Steenkiste

et al. (2019) and Steenbrugge et al. (2018), which showed that disentangled representations can boost performance in downstream RPM tasks. A RPM is an abstract visual reasoning task, where a learner is presented with a set of context image panels structured in rows, with the final panel in the final row missing. The learner must discover/infer a rule that explains how each row of the context panels is constructed in order to select the appropriate answer panel from a set of potential answer panels. In contrast to (van Steenkiste et al., 2019) and (Steenbrugge et al., 2018), in which the rules depended on a set of visual properties, in this experiment the rules depend on the numeric properties of handwritten digits, using the MNIST dataset (LeCun, 1998). The expectation is that disentanglement may not help in this case, as the numeric semantics are not native to the visual features of digit symbols. By including a numeric semantic bias, through appropriate weak relational supervision, the hope is that the representation will be encouraged to capture the non-visual semantics of the digit symbols, which will then be useful for downstream arithmetic reasoning. To do so experiments will incorporate weak relational supervision over the following signatures:

$$\sigma^E = \{\text{isEqual}(A, B)\} \quad (4.10)$$

$$\sigma^{E+G} = \{\text{isEqual}(A, B), \text{isGreater}(A, B)\} \quad (4.11)$$

$$\sigma^{E+G+S} = \{\text{isEqual}(A, B), \text{isGreater}(A, B), \text{isSuccessor}(A, B)\}. \quad (4.12)$$

Both σ^{E+G} and σ^{E+G+S} provide background knowledge that can bias the representation towards encoding the ordinality of the digit, which would otherwise not be associated with any visual property of a digit symbol. However, using `isEqual` alone is not sufficient to induce a semantic bias over the ordinality of the digits, as it can be confused with a visual `isSameSymbol` relation. This is due to the fact that each digit has a 1-1 correspondence with its shape, and so equality can be achieved by matching digits based on their shape. Including the σ^E means the relative effects of digit symbol clustering, versus biasing towards ordinality, can be compared. In contrast to the experiment described in Section 4.3.1, relation-decoding loss is implemented using a binary cross entropy loss, which also disproportionately penalises lower accuracy predictions.

Following (van Steenkiste et al., 2019; Steenbrugge et al., 2018), the downstream RPM learner is a Wild Relation Network (WRen) (Barrett et al., 2018), which is a Relation Network (RN) (Santoro et al., 2017) adapted to RPM tasks. Each RPM task instance consists of a set of 6 answer panels $\mathcal{A} = \{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_6\}$ and a set of 8 context panels $\mathcal{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_8\}$, where

all panels are image tensors with dimensions Height $\times$ Width $\times$ Colour-Channels. For each instance, the WReN selects the answer panel as follows:

$$\text{WReN}(\mathcal{A}, \mathcal{C}) = \arg \max_{\mathbf{A} \in \mathcal{A}} \overbrace{f\left(\sum_{\mathbf{o}_i, \mathbf{o}_j \in \mathcal{O}} h(\mathbf{o}_i, \mathbf{o}_j)\right)}^{\text{Relation Network}}$$

where $\mathcal{O} = \{\psi^{\text{enc}}(\mathbf{C}_1), \psi^{\text{enc}}(\mathbf{C}_2), \dots, \psi^{\text{enc}}(\mathbf{C}_8), \psi^{\text{enc}}(\mathbf{A})\}$. Concretely, the underlying RN computes a score for each answer panel $\mathbf{A} \in \mathcal{A}$ using ANNs f and h , and then the highest scoring answer panel is selected as the final answer. The experimental procedure involves switching the choice of ψ^{enc} , which is taken from either a vanilla β -VAE or a β -VAE trained with weak relational supervision. The data is split using the standard MNIST train 60,000 : 10,000 test split (LeCun, 1998). The test-set is used to test relation-decoder prediction accuracy during the source task and is used to create RPM instances consisting of data not observed during the source task. Overloading the test-set in this way is desirable, as it enables easier comparison between relation-decoding performance and downstream RPM performance. The abstract visual reasoning target task follows the same setup as in (van Steenkiste et al., 2019; Steenbrugge et al., 2018). The WReN is trained for 100,000 steps, where in each step the WReN is shown an unseen RPM instance constructed from the test MNIST set. Every 1000 steps the test WReN performance is evaluated on 100 new RPM instances. As MNIST is low dimensional compared to the datasets used in (van Steenkiste et al., 2019), in these experiments the number of neurons in the downstream WReN model is divided by a factor of 4. Additional experimental details are provided in Appendix A.

4.4 Results

The results presented in this section will investigate the interplay between disentanglement and weak relational supervision, by exploring the impact of weak relational supervision on the resulting latent representations and on downstream performance. The downstream impact will be evaluated in two ways: firstly, in held out relation approximation on relations that are new but closely related, from a semantic perspective, to the already approximated relations; secondly, by evaluating performance on a different downstream task that depends on semantically related concepts, in this case the ability to perform basic arithmetic versus the ability to approximate

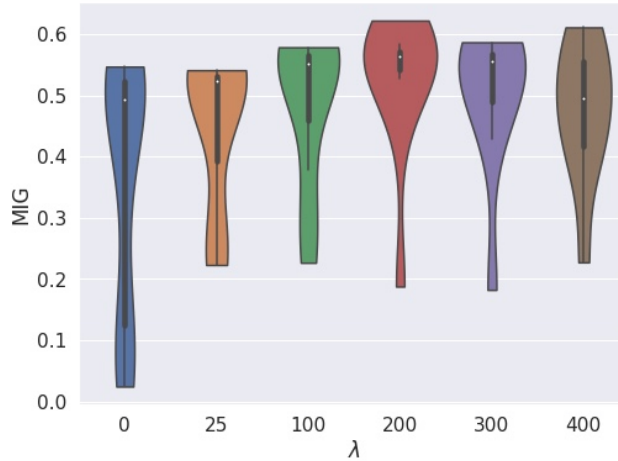


Figure 4.3: MIG score (estimated) distributions resulting from running the **dSprites** source representation learning task, with weak relational supervision over the σ^+ relations, with different λ settings and $\beta = 4$ throughout. Note that $\lambda = 0$ means the β -TCVAE is trained in the vanilla case, without any weak relational supervision. In which case we can see that the median MIG score is close to 0.5 and can be considered high (Chen et al., 2018). However the distribution of scores is very wide, indicating that the disentanglement is not reliable. By incorporating weak relational supervision, we see the the low MIG score tail push upwards. Moreover the MIG score median and maximum rise until $\lambda = 200$, at which point median scores appear to slightly worsen. Overall including weak relational supervision can be seen to improve disentanglement as measured by MIG, and that this improvement is most pronounced at $\lambda = 200$. These results are generated using 20 random restarts in each setting.

the relations the describe ordinality.

Interplay between disentanglement and weak relational supervision

Figure 4.3 presents violin plots that illustrate the disentanglement effects, as measured by the MIG score, of increasing the strength of weak relational supervision using σ^+ over the **dSprites** dataset. Violin plots are useful as they estimate the empirical distribution of the value of interest, by combining a box plot with a kernel density estimation procedure which applies a Gaussian kernel to each data point. The overall distribution is illustrated by the coloured envelope, where wider regions indicate a higher probability density, and is supplemented with the median (white dot), interquartile range (black bar), and the remaining value range (thin black whisker line). If the envelope extends beyond the whisker line, this indicates that there are data points considered to be outliers according to the interquartile range. The MIG scores are estimated using 20 random restarts for each λ setting. In general we can see that incorporating weak relational supervision improves the disentanglement of the learned representations, as measured by the MIG score.

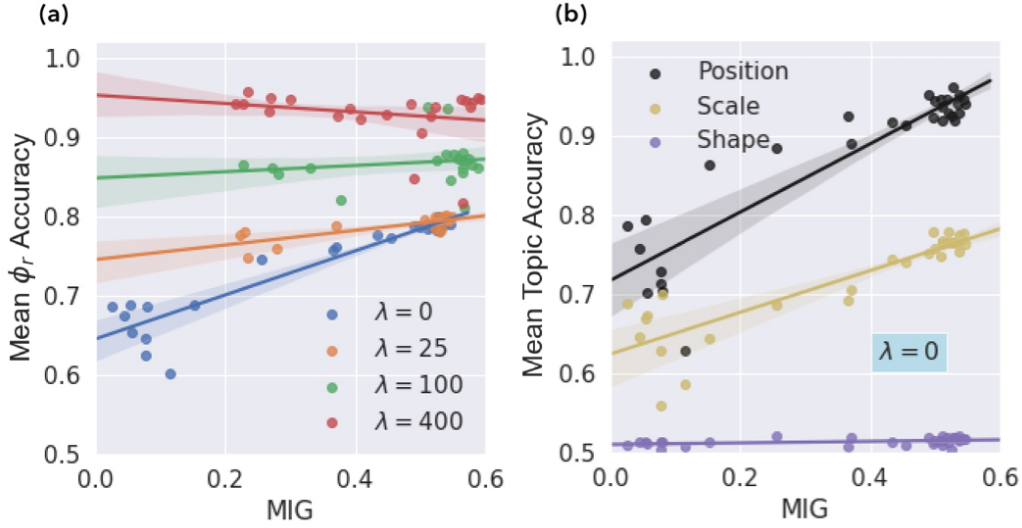


Figure 4.4: **(a)** Relationship between overall relation-decoding prediction accuracy and MIG scores, across σ^+ . A clear positive correlation between MIG and relation-decoding accuracy is observed at $\lambda = 0$, which is diminished at higher values of λ and eventually destroyed. **(b)** a break-down of relation-prediction accuracy at $\lambda = 0$ with respect to each EF dedicated relation. We see that for $\lambda = 0$ (*i.e.* vanilla β -TCVAE training case) shape relation approximation is not achievable from the learned representations, regardless of their associated disentanglement score. However, a positive correlation is observed for other EF relations.

The second set of results, illustrated by Figure 4.4, examine the correlation between test accuracy for the relation-decoders that approximate σ^+ and MIG. Here, the scatter points represent a single training run, and the line is a linear regression fit to the data, with the shaded region indicating the 95% confidence interval. Figure 4.4a shows that mean relation-decoder accuracy is higher with weak relational supervision (as expected) and these improvements continue to occur for the higher values of λ , even up to $\lambda = 400$. Moreover, there is a distinct positive correlation between MIG scores and relation-decoding performance for non-supervised encodings ($\lambda = 0$) which is diminished or destroyed at higher values of λ . Figure 4.4b, shows MIG score versus EF grouped relation-decoding performance for $\lambda = 0$. This provides a breakdown values contributing to the blue points/regression line in Figure 4.4a, and shows that certain relation-decoders are disproportionately influencing the aggregate relation-decoding accuracies, as well as the correlation between MIG and performance. In particular, we can see that the relation-decoding accuracy pertaining to the shape EF is very low regardless of MIG, but position and scale relation-decoding is positively correlated with MIG score.

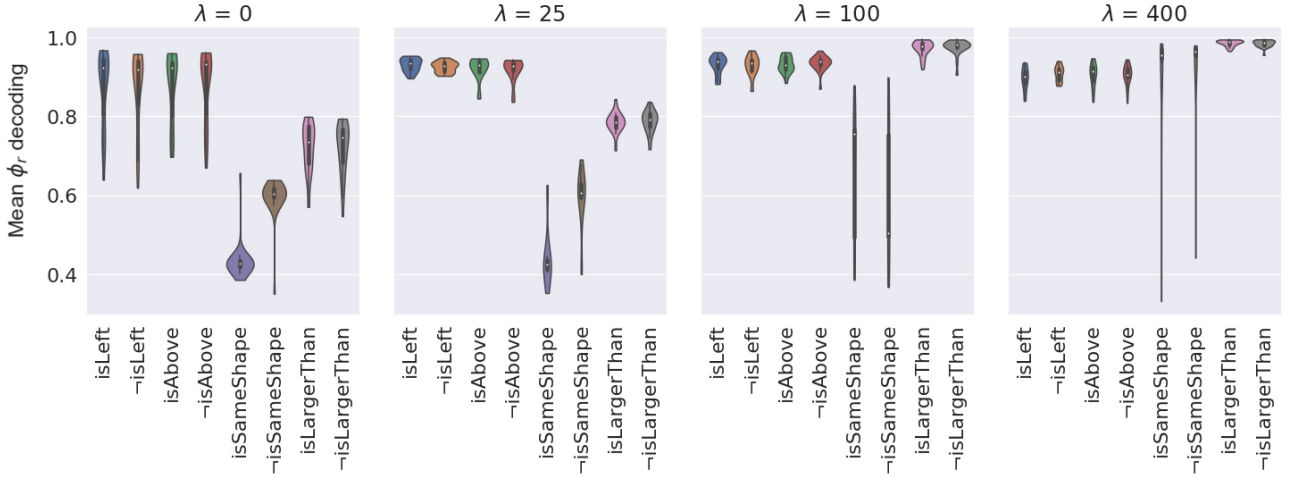


Figure 4.5: Violin plots showing the estimated relation-decoder prediction distribution for each relation in σ^+ , as λ is increased, where for $\lambda = 0$ no weak relational supervision is applied in the representation learning phase. We can see that at $\lambda = 0$ relation-decoders fail to reliably approximate shape and scale relations. At $\lambda = 100$ scale relation-decoder performance increases rapidly, but a slower performance increase is observed for shape relation-decoders. At $\lambda = 400$ improvements can be seen for both shape and scale relation-decoders, whilst position-based relation-decoding accuracy degrades slightly. Note that the $\neg$ cases should be semantically equivalent to the σ^- relations, but values are obtained using the σ^+ relation-decoders. Figure 4.6 provides the same plot for σ^- relation-decoder performance and enables evaluation as to whether the negation semantics is as expected from a FOL standpoint.

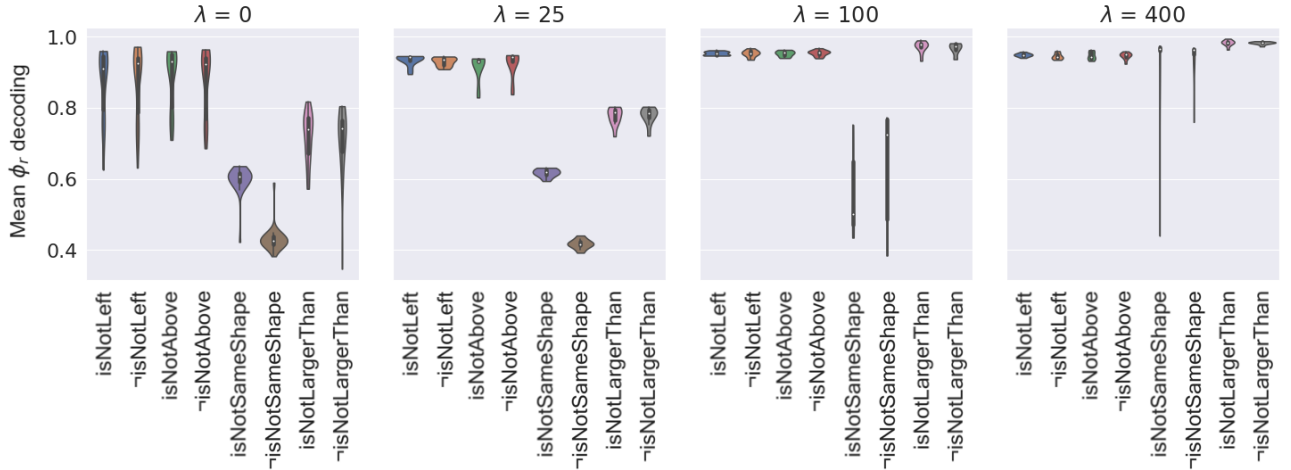


Figure 4.6: Violin plots showing the estimated relation-decoder prediction distribution for each relation in σ^- , as λ is increased, where for $\lambda = 0$ no weak relational supervision is applied in the representation learning phase. These results are largely the same as Figure 4.5 which is to be expected. Most interestingly, shape relation-decoding performance is opposite to that of σ^+ , where *e.g.* at $\lambda = 25$, $\neg$ isNotSameShape performance is approximately equivalent to isSameShape, and similarly for isNotSameScale and $\neg$ isSameScale.

Interplay between source relation-decoding and downstream relation-decoding performances

Figure 4.5 plots the mean relation-decoding performance on a per σ^+ relation basis, split between true label case and false label case prediction accuracy, for multiple training runs and values of λ . This allows us to look at how each EF topic is individually affected by increases of λ and by looking at each label case separately (as opposed to viewing the mean accuracy across both label cases) we can see how the relation-decoders perform on the *deduced* σ^- (via *e.g.* $\phi_r(\mathbf{z}_i, \mathbf{z}_j) = 1 - \phi_r(\mathbf{z}_i, \mathbf{z}_j)$). In other words, each $\neg$ case can suggest how well relation-decoders for σ^- should perform in the downstream task. The key observation here is that information necessary to learn positional concepts is most readily extracted, followed by scale, with the shape EF most resistant to encoding. This order of difficulty is interesting, as an argument can be made that shape relation-decoding is only possible once scale relation-decoding is successful, indicating a possible correspondence between scale and shape features. The same behaviour is seen in the downstream task, where the shape-type classification is the most difficult, followed by scale-type classification, with the positional-type classification being the easiest. Figure 4.6 provides the complementary plot to Figure 4.5, illustrating per relation downstream relation-decoding performance for the σ^- complement relations. By comparing between `isSameShape` versus `isNotSameShape`, the performances appear to be opposite, where `isNotSameShape` seems to correspond with $\neg$ `isSameShape` and vice versa. This outcome is as expected, indicating that the resultant latent structure consistently encodes information in a specific manner. It suggests that this encoding allows for the straightforward recovery of relation negation using new NTN⁺ relation-decoders. However, it is important to note that this recovery is limited in performance, constrained to a level similar to that of the σ^+ relation-decoders.

The second set of results in this part of the investigation looks at the possible correspondence between scale and shape from a different angle. Figure 4.7a and Figure 4.7b respectively illustrate the change in performance in σ^u relation-decoding accuracy versus `isSameShape` and `isLargerThan` σ^+ relation-decoding accuracy. Interestingly, the majority of the gains in σ^u relation-decoding accuracy occurs with minimal increases in the `isSameShape` relation-decoding accuracy, *i.e.* it is possible to achieve high unary shape relation-decoding accuracy without having high `isSameShape` relation-decoding accuracy. This is unexpected, as `isSameShape` can

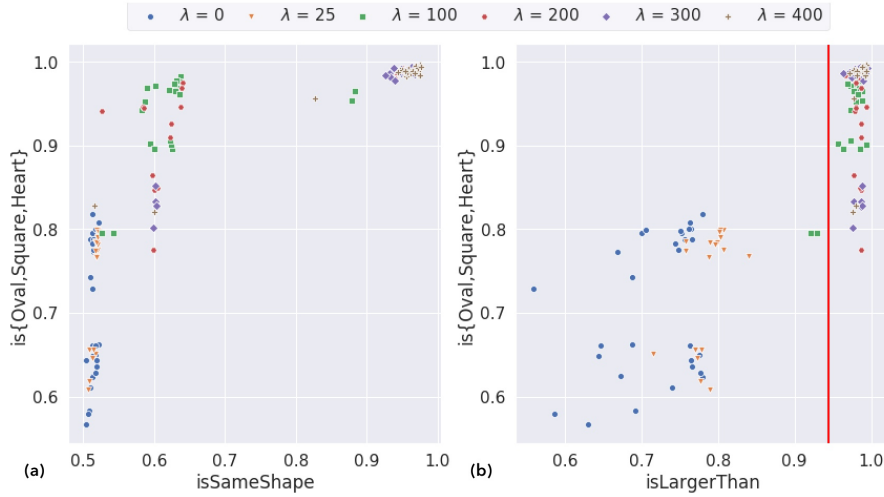


Figure 4.7: Scatter plots demonstrating how performance of (a) shape and (b) scale relation-decoding (for relations in σ^+) correlates with downstream σ^u unary shape relation prediction (effectively shape type classification). A positive correlation is observed between unary shape relation-decoding performance and source task binary shape similarity prediction. However, it appears that unary shape relation-decoding performance can perform comparatively well even at lower `isSameShape` performance and that performance seems capped until `isLargerThan` relation-decoding performance reaches ≈ 0.95 . The same can be said about `isSameShape`, which is again only reliably decoded when `isLargerThan` accuracy reaches above 0.95.

be logically deduced from unary shape classification, *e.g.* via:

$$\forall A, B \text{ (isOval}(A) \wedge \text{isOval}(B) \rightarrow \text{isSameShape}(A, B)) \quad (4.13)$$

or

$$\forall A, B \text{ (isOval}(A) \wedge \neg \text{isOval}(B) \rightarrow \neg \text{isSameShape}(A, B)). \quad (4.14)$$

Figure 4.7b may offer an explanation, as close to optimal σ^u relation-decoding accuracy occurs when `isLargerThan` relation-decoding accuracy is above roughly 0.95. Looking across the plots, it is also clear that `isSameShape` relation-decoding accuracy is only reliably achieved when `isLargerThan` relation-decoding accuracy is high. This result further supports the notion that the ability to compare scale is a necessary precursor to learning shape relations.

Abstract visual reasoning over non-visual semantics

In this final investigation, results will be presented that evaluate how adding weak relational supervision as a semantic bias can help in situations where the downstream learning task relies on abstract visual reasoning over inherently non-visual semantics, namely completing a RPM task based on basic subtraction and addition arithmetic, as described in Section 4.3. Recall

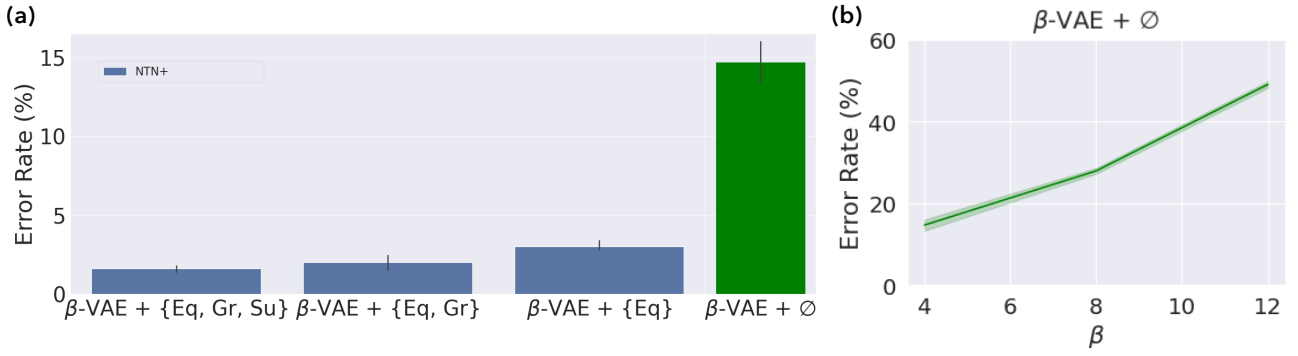


Figure 4.8: Illustration of the target phase RPM task error rates obtained with and without weak relational supervision during the source phase. In (a) results are shown for each weak relational supervision signature option, with $\beta = 4$ and $\lambda = 1000$, where signatures are indicated in the parenthesis, with abbreviations: is(Gr)eater, is(Eq)ual, and is(Su)ccessor, and β -VAE + $\emptyset$ gives results for the vanilla β -VAE training case. In addition (b) shows the effect of β in the vanilla β -VAE training case. It is clear that the RPM error rate is positively correlated with β increase in the vanilla β -VAE training. Moreover, even for the lowest β setting, the best vanilla β -VAE error rate is $\approx 15\%$. In contrast, for the weak relational supervision cases, the error rate is markedly lower, where supervising over σ^E leads to a significantly reduced error rate. Supervising over σ^{E+G} and σ^{E+G+S} leads to further, but more modest, improvements.

that for these experiments the MNIST dataset is used, which contains images of handwritten digits, and instead opt for a β -VAE in place of the previously used β -TCVAE. The results are illustrated by Figure 4.8, where the y -axis gives the minimum 5 test-step moving average test error rate ($1 - \text{accuracy}\%$) of the downstream WReN. Figure 4.8a shows the minimum error rate obtained for each weak relational supervision signature option, with $\beta = 4$ and $\lambda = 1000$, where signatures are indicated in the parenthesis and β -VAE + $\emptyset$ gives results for the vanilla β -VAE training case. We can see that incorporating weak relational supervision over σ^E leads to a significant reduction in error rate, to below 5%, versus the vanilla β -VAE training case which has an error rate of $\approx 15\%$. When incorporating weak relational supervision over the strictly ordinal relations, more modest reductions are observed. In Figure 4.8b the effect of β in the vanilla β -VAE training case is shown. It is clear that the RPM error rate is positively correlated with β increase in the vanilla β -VAE training, meaning that an increase in β leads to worse downstream RPM performance.

To gain an understanding of the underlying disentanglement performance, Figure 4.9 shows the Mutual Information (MI) statistics for the best and worst case downstream RPM performance when weak supervision is applied, where values are ordered from lowest MI (left) to highest (right). The MIG score can be estimated by looking at the gap between the first and second highest MI values in each plot. Interestingly, better WReN results are achieved when the MI spectrum is more uniform. This suggests that the WReN learner is better able to exploit the

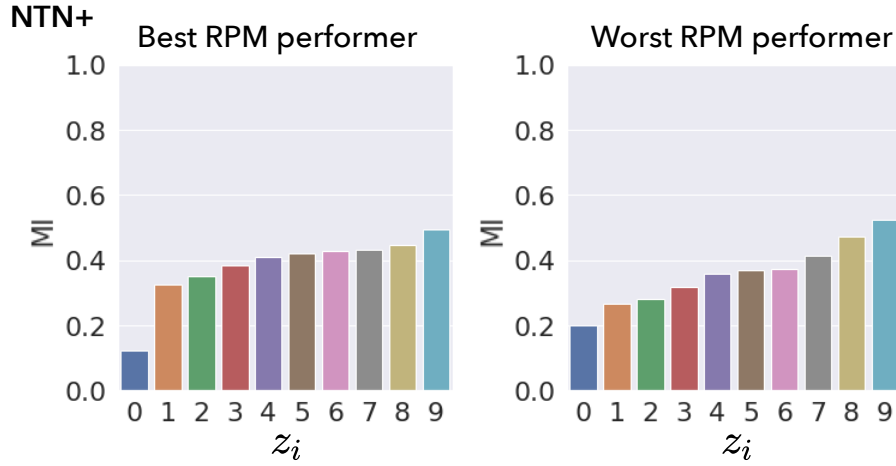


Figure 4.9: MI statistics for each latent dimension, with respect to the digit ground truth, for best (left) and worst (right) case downstream MNIST RPM performance, with dimensions ordered in ascending MI from left to right in each plot. A high MI here means that knowing the corresponding latent dimension values can provide some indication as to the input digit value. For disentanglement, the MIG score can be estimated by looking at the difference between the largest and next largest MI values. Under this lens, it seems that greater disentanglement with respect to the digit factor occurs when worse downstream performance is observed, as the MI spectrum is more uniform when the RPM performance is better.

latent space structure when the latent space is *less* disentangled with respect to the digit factor, suggesting that, in this case, disentanglement alone is insufficient for downstream performance improvements. It is worth noting that these results are dependent on the inductive-biases of the WReN and perhaps a different downstream learner may indeed benefit from a more disentangled structure with respect to the digit factor.

Lastly, Figure 4.10 shows a TSNE plot of the MNIST dataset, with perplexity 30, where each point is coloured according to its digit label, with darker values corresponding with larger digit values. The top row corresponds with the maximum MIG scores and the bottom row with the minimum downstream RPM error rates. Clustering is clearly more distinct as a result of the weak relational supervision. Interestingly, a higher β in the instance where no weak relational supervision is used shows more distinct separation of digits, compared to a higher β . This suggests that disentanglement actually obfuscates digit identity (when no additional supervision is used) and may explain why a higher β leads to worse downstream WReN performance for this task.

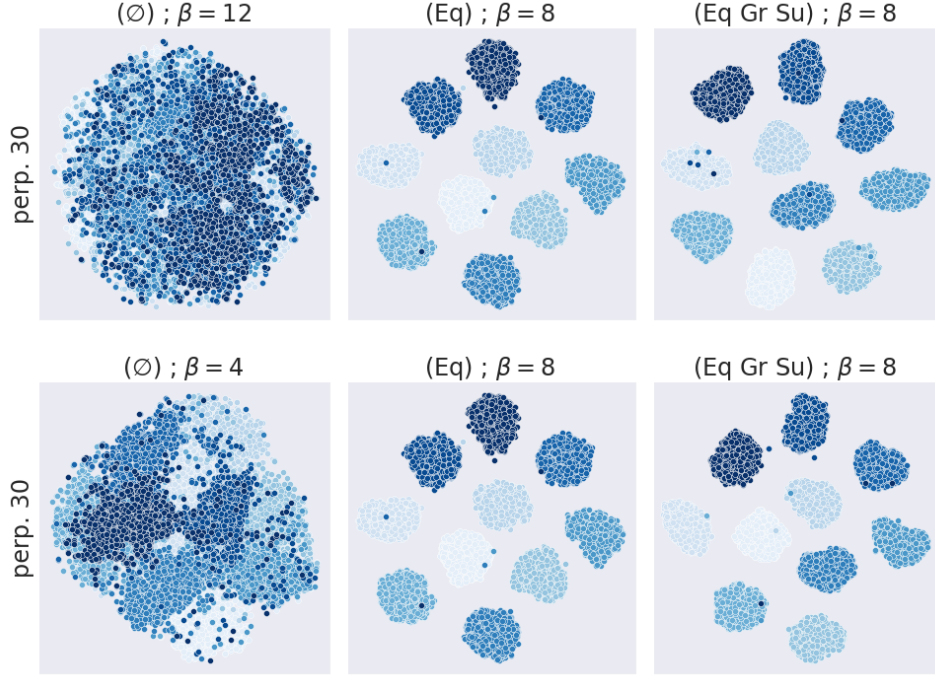


Figure 4.10: TSNE plot of the MNIST dataset, with perplexity 30, where each point is coloured according to its digit label. Darker points correspond with larger digit values. For each signature, the top row corresponds with maximum MIG scores and the bottom with minimum downstream RPM error rates. β values are shown in the plot titles, together with the level of included relational inductive bias: $\emptyset$ no bias, Eq σ^E bias, and Eq, Gr, Su σ^{E+G+S} bias. A clear clustering effect is observed when including σ^E and appears similar for σ^{E+G+S} , suggesting clustering matters most for the downstream WReN.

4.5 Discussion

Relation-decoding transfer and disentangled representations

One of the questions that the dSprites experiments sought to answer was whether relational semantics are captured in the learnt latent space, insofar as learning to approximate σ^+ relations enables downstream learnability of σ^- relations. From a FOL perspective this is expected, since by definition knowing that a relation holds between two objects implies that the negation of that relation does not hold and vice versa. The answer to this is yes. By comparing Figures 4.5 and 4.6 it is clear that in all λ cases, σ^+ and σ^- relation-decoding performances are in agreement. In particular, by comparing the accuracies of `isSameShape` (in Figure 4.5) versus `isNotSameShape` (in Figure 4.6), we can see that the performance of the downstream `isNotSameShape` relation-decoder matches the negation of the `isSameShape` relation-decoder and the reverse is also true. Overall this means that the resulting latent structure robustly exposes semantics that enables the downstream relation-decoder to ‘find’ and that this holds when weak relational supervision is

incorporated into the source task. This is a foundational result that provides a possible basis for FOL-like relational representations. In FOL, the interaction between a relation and its negation — where exactly one of them can be true in any given model — is a reflection of the principle of bivalence in Logic, and is a fundamental aspect of classical FOL semantics. It seems reasonable to expect that FOL-like relational representations would make use of this property, and that a FOL-like relation-decoder implementation will at least be able to recover negation semantics from the latent space. However, more intricate relationships between relations can exist and it is interesting to consider whether learnt latent structure is amenable to learning relations that are not so closely related to source task relations. In this regard, the results illustrated by Figure 4.7 are particularly interesting. From a logical perspective, `isSameShape` and the unary shape type relations of σ^u are closely related, through rules such as,

$$\forall A, B, (\text{isOval}(A) \wedge \text{isOval}(B) \rightarrow \text{isSameShape}(A, B)). \quad (4.15)$$

However, counter-intuitively, it is possible to achieve a downstream unary relation type decoding accuracy that is significantly higher than the source `isSameShape` relation-decoder accuracy. Instead, we see that σ^u relation-decoding accuracies seem to be better explained by improvements in source `isLargerThan` relation-decoding performance, where high σ^u is only achieved when `isLargerThan` relation-decoding performance is above roughly 0.95. It therefore seems there is more to FOL-like semantic transfer at the level of the latent representation.

Returning back to Figures 4.5 and 4.6, when weak relational supervision is not incorporated into the source task the NTN⁺ relation-decoders are unable to fully shape and scale relations. Both of these EFs have the smallest variation in the dataset, and so it is unsurprising that they are the most difficult to learn, as unsupervised disentanglement is known to be susceptible to ambiguities in the dataset (Higgins et al., 2017a; Träuble et al., 2021). Weak relational supervision is therefore beneficial, as adding a strong enough semantic bias appears to resolve these ambiguities. The order of learnability with respect to λ relational supervision strength also agrees with their relative variability in the dataset, where, at lower values, scale can be learned, but only the largest values allow shape relations to be learnt. On top of the importance of EF variability in the dataset, some EFs are more abstract than others. Whereas ordinal properties, such as position and scale, can be varied in a continuous manner, shape is a more discrete property. It is reasonable that an additional bias may be required to ‘inform’ the encoder ψ^{enc}

about these semantics so that they can be encoded appropriately.

Abstract visual reasoning over non-visual semantics

Surprisingly, the results of the RPM experiments showed that disentanglement can in fact be detrimental to downstream learning under certain conditions. In this case, though the RPM task is a visual one, the semantics of the reasoning are not inherently visual and instead rely on the numeric properties of the handwritten digits. Interestingly, a significant increase in downstream performance is observed when weak relational supervision over the σ^E signature, but supervising over additional relations produces incremental improvements. Without additional relations, `isEqual` can be solved as a visual similarity task, essentially equivalent to applying an `isSameDigitSymbol` relation-decoding. It is only when `isGreaterThan` or `isSuccessor` are included that the ordinality of the digits is likely to be captured. Looking at the TSNE plot in Figure 4.10, it seems that at large β values, although visual properties are likely to be disentangled, the data does not tightly cluster according to numeric identity. In contrast, when adding the weak relational supervision, plotting with the same TSNE perplexity clearly shows digit identity data clustering. Together these results suggest that the downstream WReN primarily relies on clear separation between digit identity. This furthermore suggests the WReN may be completing the task by memorising digit subtraction/addition pairings, instead of making use of their intrinsic magnitude values. It is unclear whether biasing for ordinality is the reason for the incremental improvement over σ^E , or whether additional clustering effects are what is most useful.

The negative impact of disentanglement on downstream RPM performance is further demonstrated in Figure 4.9, which shows the MI statistics (with respect to the digit ground-truth EF) for the best and worst case downstream RPM performance when weak supervision is applied. The representation can be considered more disentangled when the spectrum is less uniform, indicating that certain dimensions are more specialised towards the digit EF. It is clear that the spectra is more uniform in the best performing case, further supporting the notion that disentanglement alone is not always beneficial for downstream learning. It should be noted, however, that lower overall disentanglement with respect to the digit factor is not entirely unsurprising, as digit identity is a visual property and the visual variation between digits can overlap with the numeric identity.

To conclude, the results presented provide a number of insights into the interplay between disentanglement and downstream learning. We saw that disentanglement and weak relational supervision can be beneficial as overall higher disentanglement was observed when weak relational supervision was applied over the `dSprites` dataset EFs. However, the results show that sometimes disentanglement is not enough; in situations where the downstream task relies on an inherent non-visual abstract data property (*e.g.* numeric identity of handwritten `MNIST` digits) which cannot reasonably be disentangled from purely visual signal, some amount of weak relational supervision is required to provide the missing semantics. The results also illustrated interesting properties about the FOL-like behaviour of the overall architecture. We saw that the downstream relation-decoders are able to learn the negation of weakly supervised relations, which is a necessary property for FOL-like relational representations. However, we also saw that in certain situations, what should be logically deducible from the source task semantics is not necessarily the case, suggesting that there is more to FOL-like semantic transfer at the level of the latent representation. The next chapter will build on this theme and provide theoretical foundations for what it means for a relational representation to be FOL-like, and how this might be achieved.

Chapter 5

Defining coherence and consistency of Soft-Structures

The goal of this chapter is to deepen our understanding of relational representations and disentanglement. Whilst the previous chapters employed various architectures that combine relation-decoders with encoders in some manner, to leverage relational representations for transfer we need to have a formal underpinning of when these representations truly capture general relational semantics (*i.e.* are FOL-like) and offer the purported transferability of RRL in Chapter 3. The key property of interest in being FOL-like is that the semantics of the relations are general and can be applied in new ML tasks. Even when domains differ. This chapter addresses this need by proposing a framework for formally expressing and measuring logical consistency of deep relational representations, such as those obtained in Chapters 3 and 4, with respect to logical theories that constrain the semantics of the relations. Specifically, this chapter first introduces a ‘soft’ counterpart of a model theoretic structure (Hodges, 1993), referred to as a *soft-structure*, which formalises the architecture employed in Chapter 4 and allows us to draw parallels between the semantics of FOL and the semantics of deep relational representations. Equipped with this new construct, consistency with regards to structures will be contrasted with soft-structures, leading to formal definitions of ϵ -consistency and ϵ -coherence of soft-structures, which respectively express the degree to which relation-decoder beliefs are consistent with a given theory and the transferability of this consistency with regards to new domains. From here a theoretical study will be conducted to explore necessary conditions for achieving ϵ -consistency and ϵ -coherence. Finally, the chapter will conclude with a return

back to disentanglement, extending the soft-structure framework to incorporate disentangled representations, and exploring the impact of disentanglement on ϵ -consistency and ϵ -coherence.

5.1 From logical structures to soft-structures

An important property of Logic (Section 2.1) is the formal definition of semantics, *i.e.* how relations and objects obtain their *meaning*. To ground this discussion in a concrete example, this chapter will consider a specific logical theory defining the general notion of ordinality¹. We can describe ordinality using a FOL language $\mathcal{L}^{\text{FOL}}$ with variables $\mathcal{V} = \{A, B\}$ and a signature consisting of the following relation symbols

$$\sigma = \{\text{isGreater}(A, B), \text{isLess}(A, B), \text{isEqual}(A, B), \text{isSuccessor}(A, B), \text{isPredecessor}(A, B)\}, \quad (5.1)$$

where no constants or function symbols are considered. In Logic, the semantics of each relation are obtained from an interpretation function $I_{\mathcal{S}_\sigma}$, which defines the meaning of each symbol in the signature σ over a given domain $\mathcal{S}$. In other words, the interpretation function tells us which relations hold between objects in $\mathcal{S}$, and which do not, and the meaning of each relation is derived from this². Using the language we can express a theory $\mathcal{T}$ which specifies rules that constrain the relations. For instance, a key characteristic of ordinality is the notion that any two numbers cannot be both greater than and less than each other, which as a FOL sentence can be expressed as

$$\forall A, B \ (\neg (\text{isGreater}(A, B) \wedge \text{isLess}(A, B))).$$

Though a theory does not give us the semantics of relations, it does provide a set of constraints that any interpretation must satisfy, offering a semantic blueprint that determines which interpretations make sense according to the theory. Here lies a special level of generalisability that the semantics of FOL relations can offer - the theory is not tied to any particular domain, and so the same theory can be used to describe relations and their properties over any domain. However, semantics are not as clearly defined for the relational representations that we learned in Chapter 4. Often, we are content with the fact that the network can perform a task and

¹The full theory specification is provided in Appendix C.

²From a formal FOL standpoint, the notion of semantics ends here; to go further means dealing with difficult problems of “symbol grounding” (Harnad, 1990) and “binding” (Treisman, 1996; Roskies, 1999; Feldman, 2013; Greff et al., 2020)

less interested in how it does it, so long as it generalises to our needs. In disentanglement, in particular when using VAEs, semantics are often qualitatively queried using latent traversals (Higgins et al., 2017a), where the effects of independently varying each latent dimension on output reconstructions are observed. Disentanglement metrics offer a quantitative perspective in a similar vein, by producing values that try to determine the relationship between latent representation properties and underlying EFs when they are known. However, as datasets and tasks get more complex and the number of latent dimensions increases, it becomes increasingly difficult or impossible to reliably interpret the semantics of the latent space. What is missing is an evaluation of semantics as a function of both the latent representations of data and the relation-decoders used to compare them. Such an evaluation could allow us to make statements as to whether the learned relation-decoders correspond with the true FOL relation semantics that they are intended to approximate.

To establish this evaluation, we will turn our attention to the challenge of learning to approximate a particular structure over a real-world domain $\mathcal{S}$ given a signature σ . Here, a learner must determine an appropriate interpretation over real-world data, such as images or other perceptions. This can be challenging for a number of reasons. First, we may only have a partial description of the interpretation, as was the case in Chapter 4 where we did not have labels for every data pairing and relation. Second, data may be noisy and contain information that is not relevant to a given relation or theory. For instance, the handwritten digits in the MNIST dataset contains stylistic details such as line thickness and digit skew that are irrelevant to the ordering of the digits, and so learning ordinal relation-decoders would require them to filter out this noise in their predictions. And last, but not least, though relations may be general in FOL with respect to a variety of domains, different datasets can appear vastly different from each other, even if they share some of the same underlying semantics, and this presents an encoding challenge.

Following the convention of the autoencoder disentanglement literature (Bengio et al., 2013; Kingma and Welling, 2014; Higgins et al., 2017a, 2019), let us make the assumption that real-world observations S are drawn from some conditional distribution $p_{S|Z}$, where Z is a latent random variable, itself drawn from prior p_Z . It is therefore useful to define a domain *encoding* of the form:

$$\psi_S : \mathcal{S} \rightarrow \mathcal{Z}, \tag{5.2}$$

tasked with approximating the conditional expectation of the posterior, *i.e.* $\psi_S(s) \approx \mathbb{E}_{p_{Z|S}}[Z|s]$.

Since obtaining an interpretation from domain encodings for a given signature may require dealing with noise, we express the interpretation of relations over real-world data by belief functions (Paris, 1994; Paris and Vencovská, 2015; Shafer, 1976; Dempster, 2008) over the space $\mathcal{Z}$:

$$\phi_r : \mathcal{Z}^{\text{ar}(r)} \rightarrow (0, 1), \quad (5.3)$$

where $\text{ar}(r)$ was defined in Section 2.1 and denotes the arity of relation r . It is helpful to also denote the full set of belief functions for a given signature as $\phi^\sigma = \{ \phi_r \mid r \in \sigma \}$, which will be referred to interchangeably as a set of relation-decoders. Concretely, for a binary relation r and an ordered pair $(s_i, s_j) \in \mathcal{S}_2$, $\phi_r(\psi_{\mathcal{S}}(s_i), \psi_{\mathcal{S}}(s_j))$ describes the belief that $(s_i, s_j) \in I_{\mathcal{S}_\sigma}(r)$. The belief $\phi_r(\psi_{\mathcal{S}}(s_i), \psi_{\mathcal{S}}(s_j)) \approx 1$ signifies a strong belief that $(s_i, s_j) \in I_{\mathcal{S}_\sigma}(r)$ and $\phi_r(\psi_{\mathcal{S}}(s_i), \psi_{\mathcal{S}}(s_j)) \approx 0$ signifies a strong belief that $(s_i, s_j) \notin I_{\mathcal{S}_\sigma}(r)$. Note that contrary to general Dempster-Schafer theory belief functions, this thesis takes the common belief simplification $\text{Bel}(\neg A) = 1 - \text{Bel}(A)$ and assumes that each relation-decoder outputs a single belief score, as opposed to a belief and plausibility (Shafer, 1976; Dempster, 2008). As such, maximum uncertainty is modelled at a belief of 0.5. Together, $\psi_{\mathcal{S}}$ and ϕ^σ allow us to define a belief-based analogue to a model-theoretic structure.

Definition 5.1 (Soft-structure and soft-substructure) *Given a signature σ , a (possibly infinite) set $\mathcal{Z}$ and relation-decoders $\phi^\sigma = \{ \phi_r \mid r \in \sigma \}$, a soft-structure is a tuple $\tilde{\mathcal{Z}}_\sigma = (\mathcal{Z}, \phi^\sigma)$. For a (finite) domain $\mathcal{S}$ and encoding $\psi_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{Z}$, $\tilde{\mathcal{S}}_\sigma = (\psi_{\mathcal{S}}(\mathcal{S}), \phi^\sigma)$ is called a (finite) soft-substructure of $\tilde{\mathcal{Z}}_\sigma$, with a subdomain $\psi_{\mathcal{S}}(\mathcal{S}) = \{ \psi_{\mathcal{S}}(s) \mid s \in \mathcal{S} \} \subseteq \mathcal{Z}$.*

Note that $\psi_{\mathcal{S}}$ is not necessarily required for a soft-substructure, as a soft-substructure is effectively a tuple $\tilde{\mathcal{S}}_\sigma = (\mathcal{Z}', \phi^\sigma)$ where $\mathcal{Z}' \subseteq \mathcal{Z}$ domain. However, $\psi_{\mathcal{S}}$ is necessary to connect the soft-substructure beliefs to a real-world domain.

In the following we will mostly work with soft-substructures, since we will focus on observational domains which may not be surjectively mapped to $\mathcal{Z}$. But it should be noted that a finite soft-substructure can be viewed as a soft-structure whose domain is limited to the domain of the soft-substructure, *i.e.* where $\mathcal{Z} \doteq \psi_{\mathcal{S}}(\mathcal{S})$. In other words, a soft-substructure is a soft-structure with respect to itself. A soft-substructure can be used to learn a model theoretic structure over a real-world domain through learning $\psi_{\mathcal{S}}$ and ϕ^σ , such that ϕ^σ 's beliefs are in agreement with the interpretation $I_{\mathcal{S}_\sigma}$ of a given structure $\mathcal{S}_\sigma$. In this way, soft-substructure learning can be seen as

a formalisation of the relational representation learning that was done in Chapter 4, where the target structure’s interpretation was partially provided through weak relational labelling, $\psi_{\mathcal{S}}$ was provided by the VAE, and ϕ^{σ} was provided by the relational auxiliary layer. But now that we are equipped with this formalisation we can conduct a deeper exploration into the semantics of relational representations and how they relate to the semantics of FOL structures. In particular, if a theory $\mathcal{T}$ is provided not all structures may be equal and some may be more desirable than others. In which case it can be particularly interesting to consider soft-substructures that are ‘consistent’ with $\mathcal{T}$, but it is not yet clear what this formally means. With Definition 5.1 in place, the next section will introduce the notion of ϵ -consistency of soft-substructures and how it can be used to evaluate the quality of a soft-substructure with respect to a given theory.

5.1.1 Defining consistency of soft-substructures

Consistency is an important property in Logic, as it helps us determine which truths and falsehoods make sense with respect to a governing theory $\mathcal{T}$.

A set of beliefs is called consistent if these beliefs could all be true together in some possible situation. The set of beliefs is called inconsistent if there is no possible situation in which all the beliefs are true. A single belief is called consistent if it could be true in some possible situation. An inconsistent belief is said to be self-contradictory and a contradiction (Hodges, 2001).

This can be similarly beneficial in DL, as often our data will adhere to certain rules (*e.g.* cats do not have fins, dogs can bark, *etc.*) (Nye et al., 2021). Therefore it can be useful to ensure that the beliefs of our relation-decoders are consistent with these rules. Towards this end, we must first determine the degree to which a soft-substructure *supports* any given structure.

Definition 5.2 (Structure support and interpretation label) *Given a structure $\mathcal{S}_{\sigma}$ and a soft-substructure $\tilde{\mathcal{S}}_{\sigma}$, the structure support of $\tilde{\mathcal{S}}_{\sigma}$ for $\mathcal{S}_{\sigma}$ is*

$$p(\mathcal{S}_{\sigma}|\psi_{\mathcal{S}}, \phi^{\sigma}) = \prod_{r \in \sigma} \prod_{O \in \mathcal{S}^{\text{ar}}(r)} f(\phi_r, \psi_{\mathcal{S}}, O, \gamma_{\mathcal{S}_{\sigma}}^{r,O}) \quad (5.4)$$

with

$$f(\phi_r, \psi_{\mathcal{S}}, O, \gamma_{\mathcal{S}_\sigma}^{r,O}) = (\phi_r(\psi_{\mathcal{S}}(O)))^{\gamma_{\mathcal{S}_\sigma}^{r,O}} \cdot (1 - \phi_r(\psi_{\mathcal{S}}(O)))^{1-\gamma_{\mathcal{S}_\sigma}^{r,O}}, \quad (5.5)$$

where the interpretation label $\gamma_{\mathcal{S}_\sigma}^{r,O} = 1$ if $O \in I_{\mathcal{S}_\sigma}(r)$, and 0 otherwise; and $\phi_r(\psi_{\mathcal{S}}(O))$ is, by abuse of notation, shorthand for $\phi_r(\psi_{\mathcal{S}}(s_1), \dots, \psi_{\mathcal{S}}(s_n))$ with $n = \text{ar}(r)$.

Equation 5.4 assumes that, given a soft-substructure, the beliefs in what constitutes the interpretations of the different relations of a signature σ are independent of one another. For notational brevity, let us define

$$p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) \doteq p(\mathcal{S}_\sigma | \psi_{\mathcal{S}}, \phi^\sigma), \quad (5.6)$$

and use this to express the structure support of $\tilde{\mathcal{S}}_\sigma$ with respect to $\mathcal{S}_\sigma$. It is straightforward to show that $\sum_{\mathcal{S}_\sigma} p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) = 1$ (summed over all possible structures with domain $\mathcal{S}$ and signature σ): The easiest way to see this is to note that, by Equation 5.4, each $p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma)$ forms a product over independent Bernoulli random variables, and each structure will have the same terms but will differ in their unique $\gamma_{\mathcal{S}_\sigma}^{r,O}$ assignments. Therefore, the sum of all $p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma)$ will be the sum of all possible combinations of $\gamma_{\mathcal{S}_\sigma}^{r,O}$, which is equal to 1. A short worked example makes this clear: let σ be a signature containing just one binary relation, $>$, and let the domain consist of two objects, $\mathcal{S} = \{s_1, s_2\}$. Ignoring any theory, as we are at this point only interested in combining structure support for each structure, there are 4 binary pairs over $\mathcal{S}$, $\{(s_1, s_1), (s_1, s_2), (s_2, s_1), (s_2, s_2)\}$, and therefore 16 possible unique structures. To make common terms easy to identify, and for the sake of brevity, let us label each $\phi_{>}(\psi_{\mathcal{S}}(s_i), \psi_{\mathcal{S}}(s_j)), i, j \in \{1, 2\}$ by ϕ_{ij} . The structure support of a soft-substructure for all logical structures is then defined in full as follows:

$$\begin{aligned}
\sum_{\mathcal{S}_\sigma} p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) &= \phi_{12}\phi_{21}\phi_{11}\phi_{22} + \phi_{12}\phi_{21}\phi_{11}(1 - \phi_{22}) \\
&\quad + \phi_{12}\phi_{21}(1 - \phi_{11})\phi_{22} + \phi_{12}\phi_{21}(1 - \phi_{11})(1 - \phi_{22}) \\
&\quad + \phi_{12}(1 - \phi_{21})\phi_{11}\phi_{22} + \phi_{12}(1 - \phi_{21})\phi_{11}(1 - \phi_{22}) \\
&\quad + \phi_{12}(1 - \phi_{21})(1 - \phi_{11})\phi_{22} + \phi_{12}(1 - \phi_{21})(1 - \phi_{11})(1 - \phi_{22}) \\
&\quad + (1 - \phi_{12})\phi_{21}\phi_{11}\phi_{22} + (1 - \phi_{12})\phi_{21}\phi_{11}(1 - \phi_{22}) \\
&\quad + (1 - \phi_{12})\phi_{21}(1 - \phi_{11})\phi_{22} + (1 - \phi_{12})\phi_{21}(1 - \phi_{11})(1 - \phi_{22}) \\
&\quad + (1 - \phi_{12})(1 - \phi_{21})\phi_{11}\phi_{22} + (1 - \phi_{12})(1 - \phi_{21})\phi_{11}(1 - \phi_{22}) \\
&\quad + (1 - \phi_{12})(1 - \phi_{21})(1 - \phi_{11})\phi_{22} + (1 - \phi_{12})(1 - \phi_{21})(1 - \phi_{11})(1 - \phi_{22}) \quad (5.7)
\end{aligned}$$

Each summand can be viewed as a joint probability, for instance $\phi_{12}\phi_{21}\phi_{11}\phi_{22}$ is the probability $p(s_1 > s_2, s_2 > s_1, s_1 > s_1, s_2 > s_2)$. As all combinations are covered, picking any summand will mean that the remaining summands together form its complement. Therefore the sum simplifies to:

$$\begin{aligned}
\sum_{\mathcal{S}_\sigma} p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) &= p(s_1 > s_2, s_2 > s_1, s_1 > s_1, s_2 > s_2) + \neg p(s_1 > s_2, s_2 > s_1, s_1 > s_1, s_2 > s_2) \\
&= \phi_{12}\phi_{21}\phi_{11}\phi_{22} + (1 - \phi_{12}\phi_{21}\phi_{11}\phi_{22}) \\
&= 1, \quad (5.8)
\end{aligned}$$

and so the sum of the structure support for all structures here is 1. This can be demonstrated for any number of relations and domain elements, as this only affects the number of summands and the number of product terms in each summand. Equation 5.4 can therefore be treated as a probability measure, where $p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) \approx 1$ means that there is a high probability that the interpretation sampled from $\tilde{\mathcal{S}}_\sigma$ will be $I_{\mathcal{S}_\sigma}$. If we have a theory $\mathcal{T}$ over a signature σ then it is natural to ask with what weight $\tilde{\mathcal{S}}_\sigma$ supports any given structure that is a model of $\mathcal{T}$. We use *model-weight*, $\Gamma_{\mathcal{M}_S^\mathcal{T}}^{\tilde{\mathcal{S}}_\sigma}$, to describe the support given by $\tilde{\mathcal{S}}_\sigma$ to models of $\mathcal{T}$:

$$\Gamma_{\mathcal{M}_S^\mathcal{T}}^{\tilde{\mathcal{S}}_\sigma} = \sum_{\mathcal{S}_\sigma \in \mathcal{M}_S^\mathcal{T}} p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) \quad (5.9)$$

where $\mathcal{M}_S^\mathcal{T}$ is the set of all structures with domain $\mathcal{S}$ that are models of $\mathcal{T}$. This lets us compare

soft-substructures, for which a good soft-substructure will be one that has a high model-weight.

Definition 5.3 (ϵ -Consistency of soft-substructures) *Given a finite soft-substructure $\tilde{\mathcal{S}}_\sigma$ and a positive number $\epsilon \in (0, 1)$, if $\Gamma_{\mathcal{M}_S^\tau}^{\tilde{\mathcal{S}}_\sigma} \geq 1 - \epsilon$ then we say that $\tilde{\mathcal{S}}_\sigma$ is ϵ -consistent with theory $\mathcal{T}$.*

As with the definition for structure support, given a soft-substructure is a soft-structure, ϵ -consistency can be equivalently defined for soft-structures. When the context is clear, a soft-structure that is ϵ -consistent with $\epsilon \approx 0$ will be referred to as being *consistent*. This thesis proposes ϵ -consistency as an appropriate measure of the notion of consistency presented in (Nye et al., 2021). Taking the probabilistic view, a low ϵ indicates that the structure that one samples from a soft-structure/substructure is likely to be a model of the governing theory. At the limit

$$\lim_{\epsilon \rightarrow 0} \Gamma_{\mathcal{M}_S^\tau}^{\tilde{\mathcal{S}}_\sigma} = 1, \quad (5.10)$$

and so, for an infinitely small ϵ , a structure sampled from $\tilde{\mathcal{S}}_\sigma$ will be a model of $\mathcal{T}$ with probability ≈ 1 . If all structures are models, then model-weight becomes a sum of structure support for all structures, as we had in Equation 5.7, and is trivially maximised by $\sum_{\mathcal{S}_\sigma} p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma)$. However, this may not be the case - it is possible that only a subset of the possible structures are models of a given theory $\mathcal{T}$. It is therefore necessary to see whether it is possible to maximise model-weight in these cases. If $\mathcal{M}_S^\tau \neq \emptyset$, a soft-structure can always be made arbitrarily ϵ -consistent (for any $\epsilon > 0$) by appropriately supporting a single model. However, it is not immediately clear that arbitrary support is possible when support is to be shared across models. Looking at the sum over all structures in Equation 5.7, we can see that certain groups of summands appear with useful *pseudo-complimentary* properties. For instance, we can see the first two summands fit this pattern:

$$c \cdot \phi_{22} + c \cdot (1 - \phi_{22}), \quad (5.11)$$

where $c = \phi_{12}\phi_{21}\phi_{11}$. For these two summands, in the limit of $c \rightarrow 1$

$$\lim_{c \rightarrow 1} (c \cdot \phi_{22} + c \cdot (1 - \phi_{22})) = 1, \quad (5.12)$$

as the pairs become complimentary with respect to ϕ_{22} . Given these are only complimentary in the limit $c \rightarrow 1$, these patterns are referred to as *pseudo-compliments*. Similarly we can

observe the same pseudo-complimentary pattern for longer sums, for instance taking the first four summands from Equation 5.7:

$$c \cdot \phi_{11}\phi_{22} + c \cdot (1 - \phi_{11})\phi_{22} + c \cdot \phi_{11}(1 - \phi_{22}) + c \cdot (1 - \phi_{11})(1 - \phi_{22}), \quad (5.13)$$

where $c = \phi_{12}\phi_{21}$, again giving

$$\lim_{c \rightarrow 1} (c \cdot \phi_{11}\phi_{22} + c \cdot (1 - \phi_{11})\phi_{22} + c \cdot \phi_{11}(1 - \phi_{22}) + c \cdot (1 - \phi_{11})(1 - \phi_{22})) = 1. \quad (5.14)$$

In fact, for any fixed choice of c , providing that all combinations of the remaining terms are covered, the sum will always be 1 in the limit $c \rightarrow 1$.

What about other arbitrary choices of summands which do not form a pseudo-compliment, can we still distribute support across each structure? For instance, let us select the second and third summand in Equation 5.7 and assume,

$$\phi_{12}\phi_{21}\phi_{11}(1 - \phi_{22}) + \phi_{12}\phi_{21}(1 - \phi_{11})\phi_{22} \approx 1. \quad (5.15)$$

In which case, the support for all other structures must be 0, which would mean that any other summand is close to 0, *e.g.*

$$\phi_{12}\phi_{21}(1 - \phi_{11})(1 - \phi_{22}) \approx 0. \quad (5.16)$$

We can see that for Equation 5.16 to be satisfied, either $\phi_{12} \approx 0$, $\phi_{21} \approx 0$, $\phi_{11} \approx 1$, or $\phi_{22} \approx 1$, but each choice means zeroing support for one of the summands in Equation 5.15 and so it does not appear possible to satisfy Equation 5.15 without having zero support for one of the summands. Next, let us look at a length 3 selection. Take, for instance

$$c \cdot \phi_{11}\phi_{22} + c \cdot (1 - \phi_{11})\phi_{22} + c \cdot \phi_{11}(1 - \phi_{22}) \approx 1. \quad (5.17)$$

Once again looking at remaining structure supports, which must all be 0, we can see that $c \cdot (1 - \phi_{11})(1 - \phi_{22}) \approx 0$ but since Equation 5.16 implies that $c \approx 1$, this means that either $\phi_{11} \approx 1$ or $\phi_{22} \approx 1$. Each of these choices effectively zeros the support for a structure in Equation 5.17 and so neither suffices and hence Equation 5.16 cannot be satisfied.

To summarise the worked example so far, it appears that non-trivial solutions to maximising

model-weight are possible if the set of structures to support form a pseudo-compliment. However, the likelihood that a selected subset of structures will form a pseudo-compliment becomes vanishingly small as the domain size and/or signature grows. The effect is that for a given domain $\mathcal{S}$, signature σ , and theory $\mathcal{T}$, it is unlikely that $\mathcal{M}_{\mathcal{S}}^{\mathcal{T}}$ will form a pseudo-compliment, unless specific conditions are placed on $\mathcal{T}^3$, in which case model-weight can only be maximised by maximising structure support for a single model of $\mathcal{M}_{\mathcal{S}}^{\mathcal{T}}$. To see this, let us extend the above $|\mathcal{S}| = 2$, $\sigma = \{>\}$, worked example by introducing a theory of strict total ordering, defined by the following axioms:

$$\forall A (\top \rightarrow \neg(A > A)) \quad (\text{Irreflexivity}) \quad (5.18)$$

$$\forall A, B ((A > B) \rightarrow \neg(B > A)) \quad (\text{Antisymmetry}) \quad (5.19)$$

$$\forall A, B, C (((A > B) \wedge (B > C)) \rightarrow (A > C)) \quad (\text{Transitivity}) \quad (5.20)$$

$$\forall A, B (A \neq B \rightarrow (A > B) \vee (B > A)) \quad (\text{Totality}) \quad (5.21)$$

With two domain elements and one binary relation, we can construct 16 possible structures. However, only 2 of these are models of the governing theory, *i.e.* $\mathcal{M}_{\mathcal{S}}^{\mathcal{T}} = \{\mathcal{S}_{\sigma}^1, \mathcal{S}_{\sigma}^2\}$, with interpretations:

$$I_{\mathcal{S}_{\sigma}^1}(>) = \{(s_1, s_2)\} \quad (5.22)$$

$$I_{\mathcal{S}_{\sigma}^2}(>) = \{(s_2, s_1)\}. \quad (5.23)$$

Using our previous shorthand notation, we can compute the model-weight as

$$\Gamma_{\mathcal{M}_{\mathcal{S}}^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}} = p(\mathcal{S}_{\sigma}^1 | \tilde{\mathcal{S}}_{\sigma}) + p(\mathcal{S}_{\sigma}^2 | \tilde{\mathcal{S}}_{\sigma}) \quad (5.24)$$

$$= \phi_{12}(1 - \phi_{21})(1 - \phi_{11})(1 - \phi_{22}) + (1 - \phi_{12})\phi_{21}(1 - \phi_{11})(1 - \phi_{22}), \quad (5.25)$$

which does not form a pseudo-complement. For this sum, only solutions that favour a single model can maximise the model-weight. Indeed, looking at Figure 5.1 we can see that it is only possible to achieve $\Gamma_{\mathcal{M}_{\mathcal{S}}^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}} \approx 1$ when the belief is concentrated on a single model.

Since a soft-substructure can always maximise consistency via unitary solutions, where unitary means with belief focused on a single structure. Moreover, given that non-trivial solutions may

³The significance of such theories is an open question for future work.

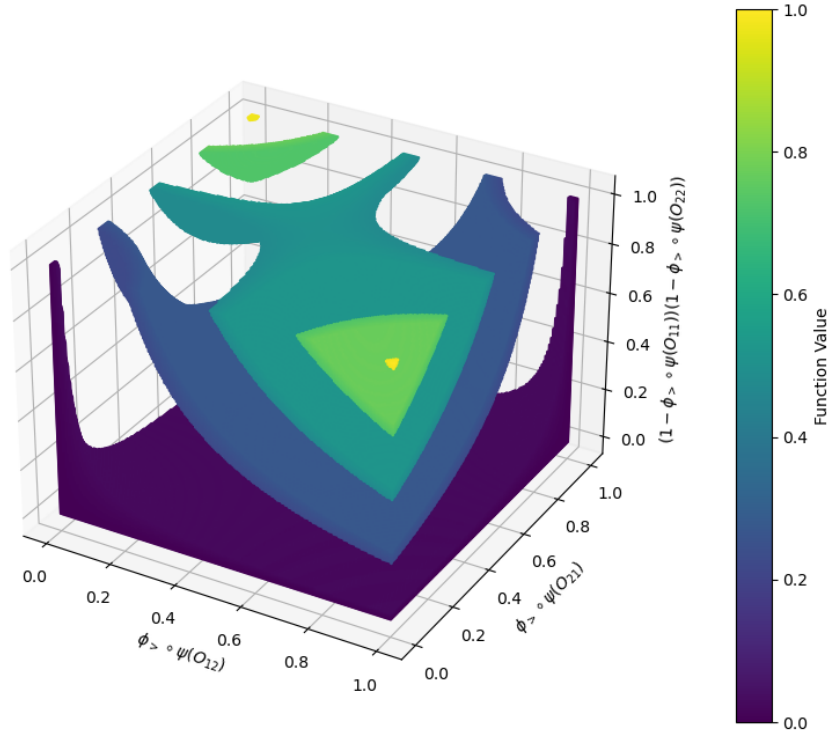


Figure 5.1: Plotting $\Gamma_{\mathcal{M}_S^T}^{\tilde{\sigma}}$ for different ground atom beliefs. The colour of each region indicates the model-weight for different ground atom beliefs. To make the plot more legible only certain model-weight value bands are plotted. These are, from dark blue to yellow: $0 \leq \Gamma_{\mathcal{M}_S^T}^{\tilde{\sigma}} \leq 0.025$, $0.225 \leq \Gamma_{\mathcal{M}_S^T}^{\tilde{\sigma}} \leq 0.275$, $0.475 \leq \Gamma_{\mathcal{M}_S^T}^{\tilde{\sigma}} \leq 0.525$, $0.725 \leq \Gamma_{\mathcal{M}_S^T}^{\tilde{\sigma}} \leq 0.775$, and $0.975 \leq \Gamma_{\mathcal{M}_S^T}^{\tilde{\sigma}} \leq 1$. It is clear that, if we want model-weight to be above some threshold, *e.g.* 0.8 (lighter green contours), then the belief must be concentrated on one of the two models.

not be possible for certain models, it is quite possible that to maximise consistency with different models, new soft-substructures/structures have to be learned.

So far we have only explored conditions over single soft-substructures, but in the next section our focus will move to characteristics shared between sets of soft-substructures and define the notion of *coherence* between soft-substructures.

5.1.2 Defining coherence of soft-substructures

In the previous section we saw that it may not be possible to achieve a high model-weight by supporting more than one model of a given theory and domain, but, in contrast, it is always possible to achieve a high model-weight by supporting a single model. As such, in

situations where we must consider multiple models, it may be necessary to learn multiple soft-substructures/structures. If models have commonalities, either in terms of domain, signature, or theory, we do not want to have to learn everything from scratch and would desire that in learning one soft-substructure, a learning agent can generalise the appropriate knowledge to other soft-substructures. In the context of lifelong learning agents, which this thesis is motivated by, the overarching ambition here is that achieving a high model-weight for one soft-substructure would lend itself to achieving high model-weight for other soft-substructures in the case of the same theory, and, together, learning each soft-substructure leads to a general understanding of the world that the agent exists in. Firstly, let us clarify this desire, starting with an intuitive example. Consider a human that learns the concept of ordinality, which we explored in Section 5.1. If a human has successfully learned this concept they should be able to apply it to any domain that has a suitable ordering, such as the ordering of numbers, or the ordering of events in time; or even performing an ordering over a domain of abstract objects (ordering for the sake of ordering), as we saw in Section 5.1.1. To do so, some degree of knowledge transfer is required.

Returning back to our soft-structure formalism, each model ordering from Section 5.1 can be supported using a different soft-substructure, where each soft-substructure concentrates its belief in just one model ordering, but the same signature and theory apply. If each soft-substructure supports a single model with high model-weight, meaning the ordering is ‘successful’, then each is consistent with the same theory of ordinality. Provided that these soft-substructures are said to have truly captured (with high model-weight) the concept of ordinality, then we would expect that the knowledge transfer between soft-substructures is such that the ϵ -consistency of one soft-substructure implies the ϵ -consistency of the other soft-substructures. More formally, assume we have two soft-substructures, $\tilde{\mathcal{S}}_\sigma^1$ and $\tilde{\mathcal{S}}_\sigma^2$, that share a signature σ , for which a single theory $\mathcal{T}$ applies. Let us make no further assumptions, other than that $\Gamma_{\mathcal{M}_{\tilde{\mathcal{S}}_1}^{\tilde{\mathcal{S}}_\sigma^1}} \approx 1$. The question of interest is then: under what conditions does the ϵ -consistency of $\tilde{\mathcal{S}}_\sigma^1$ imply ϵ -consistency of $\tilde{\mathcal{S}}_\sigma^2$? Concretely, we are interested in specifying when the following ϵ -coherence condition holds.

Definition 5.4 (ϵ -Coherence across soft-substructures) *Two soft-substructures, $\tilde{\mathcal{S}}_\sigma^1$ and $\tilde{\mathcal{S}}_\sigma^2$, that share the same signature σ , are said to be ϵ -coherent with respect to a theory $\mathcal{T}$, if and only if $\tilde{\mathcal{S}}_\sigma^1$ is ϵ_1 -consistent with $\mathcal{T}$, $\tilde{\mathcal{S}}_\sigma^2$ is ϵ_2 -consistent with $\mathcal{T}$, and $\epsilon_1 \leq \epsilon$, and $\epsilon_2 \leq \epsilon$.*

This thesis proposes ϵ -coherence as an appropriate measure of the notion of coherence presented

in (Nye et al., 2021) and will loosely refer to a soft-substructure that is ϵ -coherent with $\epsilon \approx 0$ as being *coherent*. Having ϵ -coherence between two soft-substructures means that each is at least ϵ -consistent with respect to a given $\mathcal{T}$. In terms of ordinality, if $\epsilon \approx 0$, this would mean both soft-substructures have a high belief in structures that are valid orderings. As we saw in the previous section, this can always be achieved if each soft-substructure supports a single model over their respective domains. To summarise, it is desirable for a lifelong learning agent to generalise consistency, such that achieving consistency of one soft-substructure can imply consistency of another. We now have a formal definition for this, but we have not yet specified how this might be achieved.

5.2 Exploring when we can obtain coherence

Previously we saw an example of a theory, namely that of strict ordering, which permitted two models, $\mathcal{S}_\sigma^1$ and $\mathcal{S}_\sigma^2$, for a fixed domain $\mathcal{S} = \{s_1, s_2\}$, signature σ , and theory $\mathcal{T}$. We also saw that it was only possible to achieve maximum model-weight by supporting one or the other. Building upon this example, take soft-substructure $\tilde{\mathcal{S}}_\sigma^1$ such that $p(\mathcal{S}_\sigma^1 | \tilde{\mathcal{S}}_\sigma^1) \approx 1$ and similarly soft-substructure $\tilde{\mathcal{S}}_\sigma^2$ such that $p(\mathcal{S}_\sigma^2 | \tilde{\mathcal{S}}_\sigma^2) \approx 1$. Each soft-substructure can differ in their domain (influenced via their respective encoders) or belief functions. They could differ in both, but as knowledge transfer is required we will assume that they share one or the other. The choice of which to pick is not obvious: fixing $\phi_>$ means fixing beliefs over $\mathcal{Z}$ and fitting the encodings to these beliefs, whereas fixing the encoder means fixing the mapping from $\mathcal{S}$ to $\mathcal{Z}$ and modifying beliefs over $\mathcal{Z}$, essentially reinterpreting these points. Though each may have merit, this thesis will explore ϵ -coherence between soft-substructures that have the same relation-decoders ϕ^σ , but may change encoder ψ_* . Although this thesis opts for fixed belief functions, the broader practical and theoretical implications of choosing to fix either the encoder or belief functions is a very rich topic for future research.

So let us now explore when ϵ -coherence may be implied between soft-substructures that share relation-decoders but may have different encoders. Firstly, we need to define a *substructure* and an *extension* (Hodges, 1993).

Definition 5.5 (Substructure and extension) Let $\mathcal{S}_\sigma^1$ and $\mathcal{S}_\sigma^2$ be structures that have signature σ . We say that $\mathcal{S}_\sigma^2$ is a substructure⁴ of $\mathcal{S}_\sigma^1$, denoted $\mathcal{S}_\sigma^2 \subseteq \mathcal{S}_\sigma^1$, if:

1. $\mathcal{S}_2 \subseteq \mathcal{S}_1$, i.e. the domain of $\mathcal{S}_2$ is a subset of the domain of $\mathcal{S}_1$.
2. For every relation symbol r of arity m in σ , and for all $s_1, \dots, s_m \in \mathcal{S}_2$:

$$(s_1, \dots, s_m) \in I_{\mathcal{S}_\sigma^2}(r) \leftrightarrow (s_1, \dots, s_m) \in I_{\mathcal{S}_\sigma^1}(r)$$

Similarly if $\mathcal{S}_\sigma^2 \subseteq \mathcal{S}_\sigma^1$ then $\mathcal{S}_\sigma^1$ is an extension of $\mathcal{S}_\sigma^2$.

With this definition we can introduce the following theorem.

Theorem 5.1 (Substructure support) Let $\mathcal{S}_\sigma = (\mathcal{S}, I_{\mathcal{S}_\sigma})$ be a structure over domain $\mathcal{S}$ and $\tilde{\mathcal{S}}_\sigma = (\psi_{\mathcal{S}}(\mathcal{S}), \phi^\sigma)$ be a soft-substructure over domain $\psi_{\mathcal{S}}(\mathcal{S})$. For any substructure, $\mathcal{S}_\sigma^* = (\mathcal{S}_*, I_{\mathcal{S}_\sigma^*})$, $\mathcal{S}_* \subseteq \mathcal{S}$, of $\mathcal{S}_\sigma$, with corresponding soft-substructure $\tilde{\mathcal{S}}_\sigma^* = (\psi_{\mathcal{S}}(\mathcal{S}_*), \phi^\sigma)$ over domain $\psi_{\mathcal{S}}(\mathcal{S}_*)$,

$$p(\mathcal{S}_\sigma^* | \tilde{\mathcal{S}}_\sigma^*) \geq p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma).$$

(Proof) Recall that, from Equation 5.4, structure support $p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma)$ is calculated as a product over $r \in \sigma$ and $O \in \mathcal{S}^{\text{ar}(r)}$ terms that, by definition of ϕ_r (Equation 5.3), will take values in the $(0, 1)$ open interval. This means that structure support is a monotonically decreasing function in the number of multiplicative terms. Given this, the proof follows from the fact that since $\tilde{\mathcal{S}}_\sigma^*$ and $\tilde{\mathcal{S}}_\sigma$ share the same σ , $\psi_{\mathcal{S}}$, and ϕ^σ , then: (1) the number of terms in $p(\mathcal{S}_\sigma^* | \tilde{\mathcal{S}}_\sigma^*)$ will be less than or equal to the number of terms in $p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma)$, (2) all of the terms in $p(\mathcal{S}_\sigma^* | \tilde{\mathcal{S}}_\sigma^*)$ will be in $p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma)$, and (3) in supporting $\mathcal{S}_\sigma$, $\tilde{\mathcal{S}}_\sigma$ will also support $\mathcal{S}_\sigma^*$. To show that these points hold, let us first introduce a simplification to the notation. Let $\mathcal{K}_\sigma^{\mathcal{S}}$ be the set of (r, O) tuples, corresponding to each $r \in \sigma, O \in \mathcal{S}^{\text{ar}(r)}$ combination. We can then write the structure support $p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma)$ as

$$p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) = \prod_{(r, O) \in \mathcal{K}_\sigma^{\mathcal{S}}} (\phi_r(\psi_{\mathcal{S}}(O)))^{\gamma_{\tilde{\mathcal{S}}_\sigma}^{r, O}} \cdot (1 - \phi_r(\psi_{\mathcal{S}}(O)))^{1 - \gamma_{\tilde{\mathcal{S}}_\sigma}^{r, O}}, \quad (5.26)$$

⁴This definition is scoped to signatures comprising only relation symbols and no functions nor constants. However the definition can be extended to include these, and the following theorems will still hold.

and similarly for $p(\mathcal{S}_\sigma^*|\tilde{\mathcal{S}}_\sigma^*)$,

$$p(\mathcal{S}_\sigma^*|\tilde{\mathcal{S}}_\sigma^*) = \prod_{(r,O) \in \mathcal{K}_\sigma^{\mathcal{S}_\sigma^*}} (\phi_r(\psi_{\mathcal{S}}(O)))^{\gamma_{\mathcal{S}_\sigma^*}^{r,O}} \cdot (1 - \phi_r(\psi_{\mathcal{S}}(O)))^{1-\gamma_{\mathcal{S}_\sigma^*}^{r,O}}. \quad (5.27)$$

Point (1) comes from the fact that through $\mathcal{S}_* \subseteq \mathcal{S}$, then $|\mathcal{S}_*^{\text{ar}(r)}| \leq |\mathcal{S}^{\text{ar}(r)}|$ for each $r \in \sigma$, and so $\mathcal{K}_\sigma^{\mathcal{S}_*} \subseteq \mathcal{K}_\sigma^{\mathcal{S}}$. Point (2) is due to the fact that ϕ^σ and $\psi_{\mathcal{S}}$ are shared between $\tilde{\mathcal{S}}_\sigma$ and $\tilde{\mathcal{S}}_\sigma^*$, ensuring that all relation-decodings for a given (r, O) tuple will be equal. Finally, point (3) is due to the substructure relationship $\mathcal{S}_\sigma^* \subseteq \mathcal{S}_\sigma$ - in being a substructure, Definition 5.5 tells us that $O \in I_{\mathcal{S}_\sigma^*}(r) \leftrightarrow O \in I_{\mathcal{S}_\sigma}(r)$, meaning $I_{\mathcal{S}_\sigma^*}(r) \subseteq I_{\mathcal{S}_\sigma}(r)$ for all $r \in \sigma$, $O \in \mathcal{S}_*^{\text{ar}(r)}$. This ensures that $\gamma_{\mathcal{S}_\sigma^*}^{r,O}$ and $\gamma_{\mathcal{S}_\sigma}^{r,O}$ are the same for each (r, O) tuple. The net effect of (1-3) is that all $(r, O) \in \mathcal{K}_\sigma^{\mathcal{S}_*}$ tuples in Equation 5.27 occur in Equation 5.26 and have the same relation-decoding and labels, and so all terms in Equation 5.27 will be equal to the counterpart (r, O) term in Equation 5.26. As $p(\mathcal{S}_\sigma^*|\tilde{\mathcal{S}}_\sigma^*)$ is an equal length or shorter product than $p(\mathcal{S}_\sigma|\tilde{\mathcal{S}}_\sigma)$, but both are monotonically decreasing functions, then $p(\mathcal{S}_\sigma^*|\tilde{\mathcal{S}}_\sigma^*) \geq p(\mathcal{S}_\sigma|\tilde{\mathcal{S}}_\sigma)$. This completes the proof. It also follows that if $\mathcal{S}_* \subset \mathcal{S}$ then $p(\mathcal{S}_\sigma^*|\tilde{\mathcal{S}}_\sigma^*) > p(\mathcal{S}_\sigma|\tilde{\mathcal{S}}_\sigma)$.

Without additional assumptions, Theorem 5.1 does not imply ϵ -consistency nor ϵ -coherence between soft-substructures, but it does present opportunities for each. Specifically, the following corollary for ϵ -consistency can be obtained.

Corollary 5.1 (Substructure consistency) *Given a theory $\mathcal{T}$ over signature σ , let $\mathcal{S}_\sigma = (\mathcal{S}, I_{\mathcal{S}_\sigma})$ be a structure with domain $\mathcal{S}$, and let $\tilde{\mathcal{S}}_\sigma = (\psi_{\mathcal{S}}(\mathcal{S}), \phi^\sigma)$ be a soft-substructure over domain $\psi_{\mathcal{S}}(\mathcal{S})$. Let $\mathcal{S}_\sigma^* = (\mathcal{S}_*, I_{\mathcal{S}_\sigma^*})$, $\mathcal{S}_* \subseteq \mathcal{S}$, be a substructure of $\mathcal{S}_\sigma$ with corresponding soft-substructure $\tilde{\mathcal{S}}_\sigma^* = (\psi_{\mathcal{S}}(\mathcal{S}_*), \phi^\sigma)$ over domain $\psi_{\mathcal{S}}(\mathcal{S}_*)$. If $\mathcal{S}_\sigma^* \in \mathcal{M}_{\mathcal{S}_*}^{\mathcal{T}}$ and $p(\mathcal{S}_\sigma|\tilde{\mathcal{S}}_\sigma) = 1 - \epsilon$, then $\tilde{\mathcal{S}}_\sigma^*$ will be at least ϵ -consistent.*

(Proof Sketch) Assume $\mathcal{S}_\sigma^* \in \mathcal{M}_{\mathcal{S}_*}^{\mathcal{T}}$ and $p(\mathcal{S}_\sigma|\tilde{\mathcal{S}}_\sigma) = 1 - \epsilon$. By Theorem 5.1 we have that $p(\mathcal{S}_\sigma^*|\tilde{\mathcal{S}}_\sigma^*) \geq 1 - \epsilon$. Therefore $\Gamma_{\mathcal{M}_{\mathcal{S}_*}^{\mathcal{T}}}^{\tilde{\mathcal{S}}_\sigma^*} \geq 1 - \epsilon$. Hence, $\tilde{\mathcal{S}}_\sigma^*$ will be at least ϵ -consistent.

The outcome of Corollary 5.1 is that the support $p(\mathcal{S}_\sigma|\tilde{\mathcal{S}}_\sigma)$ acts as a lower bound on ϵ -consistency for soft-substructures that support sub-models (substructures that are models) of $\mathcal{S}_\sigma$. Notice that if $\mathcal{S}_\sigma$ is also a model, then Corollary 5.1 could be expressed in terms of ϵ -coherence. But we can also derive ϵ -coherence claims with respect to two different sub-models of $\mathcal{S}_\sigma$, which has more utility for our purposes.

Corollary 5.2 (ϵ -Coherence between soft-substructures) *Given a theory $\mathcal{T}$ over signature σ , let $\mathcal{S}_\sigma = (\mathcal{S}, I_{\mathcal{S}_\sigma})$ be a structure over domain $\mathcal{S}$ and $\tilde{\mathcal{S}}_\sigma = (\psi_{\mathcal{S}}(\mathcal{S}), \phi^\sigma)$ be a soft-substructure over domain $\psi_{\mathcal{S}}(\mathcal{S})$. Let structures $\mathcal{S}_\sigma^1 = (\mathcal{S}_1, I_{\mathcal{S}_\sigma^1})$, $\mathcal{S}_1 \subseteq \mathcal{S}$, and $\mathcal{S}_\sigma^2 = (\mathcal{S}_2, I_{\mathcal{S}_\sigma^2})$, $\mathcal{S}_2 \subseteq \mathcal{S}$, be substructures of $\mathcal{S}_\sigma$ and similarly let $\tilde{\mathcal{S}}_\sigma^1 = (\psi_{\mathcal{S}}(\mathcal{S}_1), \phi^\sigma)$ and $\tilde{\mathcal{S}}_\sigma^2 = (\psi_{\mathcal{S}}(\mathcal{S}_2), \phi^\sigma)$ be soft-substructures over domains $\psi_{\mathcal{S}}(\mathcal{S}_1)$ and $\psi_{\mathcal{S}}(\mathcal{S}_2)$, respectively. If $\mathcal{S}_\sigma^1 \in \mathcal{M}_{\mathcal{S}_1}^{\mathcal{T}}$ and $\mathcal{S}_\sigma^2 \in \mathcal{M}_{\mathcal{S}_2}^{\mathcal{T}}$, and $p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) = 1 - \epsilon$, then $\tilde{\mathcal{S}}_\sigma^1$ and $\tilde{\mathcal{S}}_\sigma^2$ will be at least ϵ -coherent.*

(Proof Sketch) The proof follows directly from Corollary 5.1, which tells us that both $\tilde{\mathcal{S}}_\sigma^1$ and $\tilde{\mathcal{S}}_\sigma^2$ will be at least ϵ -consistent. Thus, by Definition 5.4, we have that they are at least ϵ -coherent with each other.

According to Corollary 5.2, if we can find a common extension of two models, and providing we have learned a soft-substructure that supports this extension, then we can obtain coherence guarantees between soft-substructures applied to each subdomain by using the same encoder and relation-decoder ‘hardware’ in their implementation. The question that we must now address is: when can we expect to find such substructure-extension relationships? Connecting back to the original motivation behind this thesis, the next section will explore this question in the context of lifelong learning agents acting in the real-world.

5.3 Coherence in real-world domains

The overarching motivation of this thesis is to take steps towards artificial lifelong learning agents operating in the real-world, so let us now investigate Corollary 5.2 within this context. The goal of this section is to draw connections between the theoretical foundations that have been established thus far and practical real-world lifelong learning. In the spirit of Chapter 4, these connections will be heavily inspired by disentanglement, as defined by Higgins et al. (2019)⁵.

A lifelong learning agent operating in the real-world will perceive real-world observations through its sensory capacity. As we did in Section 2.4.1, let us denote the space of real-world observations by $\mathcal{X}$. During a given task, the agent will observe a subset of task-relevant observations (*i.e.* observations made during the task) which we can denote by $\mathcal{X}_n \subseteq \mathcal{X}$ for the n th task, $n \in \mathbb{Z}_0^+$.

⁵Higgins et al. (2019)’s definition of disentanglement is provided by Definition 2.2

As we saw in Chapter 4, the components that are used to define a soft-substructure were applied in a general relational representation learning setup, in which a ML agent obtains a representation of its observations that is amenable to positive transfer on downstream tasks. In a similar vein, we can frame soft-substructure learning as a relational representation learning problem, in which a lifelong learning agent, equipped with a vocabulary σ , seeks to learn an internal representation for its observations such that its beliefs match the ground-truth state of its observations. In doing so the agent may attempt to uncover the regularities that underpin its observations and, in the process, learn a representation that is amenable to positive transfer on downstream tasks. This thesis frames this learning process in terms of learning to support models of theories that describe these underlying regularities. Let us define $\mathcal{X}_\sigma = (\mathcal{X}, I_{\mathcal{X}_\sigma})$ to be a structure over $\mathcal{X}$, and let us assume that this structure will be a model of some latent theory⁶ and that it has substructures. The task-structure that an agent has access to during the n th task can be similarly defined as a substructure $\mathcal{X}_\sigma^n = (\mathcal{X}_n, I_{\mathcal{X}_\sigma^n})$, with $\mathcal{X}_n$ as defined earlier. The relational representation learning challenge that the agent faces is to maximise the structure support $p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n)$, where $\tilde{\mathcal{X}}_\sigma^n$ is a soft-substructure $\tilde{\mathcal{X}}_\sigma^n = (\psi_{\mathcal{X}}(\mathcal{X}_n), \phi^\sigma)$, $\phi^\sigma = \{ \phi_r \mid r \in \sigma \}$. The forward looking agent would want the resulting learning process to improve support for other task-structures that it may be faced with in the future. Towards this end, Corollary 5.2 tells us that when maximising $p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n)$, our agent should aim to also maximise $p(\mathcal{X}_\sigma | \tilde{\mathcal{X}}_\sigma)$, in order to ensure its internal representation is amenable to new task-substructures $\mathcal{X}_\sigma^m$, $m \neq n$, $m \in \mathbb{Z}_0^+$. However, this means attempting to generalise to *all possible observations*, but, as discussed in Section 2.4.1, $\mathcal{X}$ can be a high dimensional and complex space in which the underlying regularities are heavily entangled and therefore difficult to decipher. In contrast, adopting Higgins et al. (2019)’s disentangled view, we can hypothesize the existence of a latent world-state space, $\mathcal{W}$, which is disentangled and more clearly exposes the Explanatory Factors (EFs) from which $\mathcal{X}$ is generated. From this viewpoint, a better option for our agent is to learn to represent $\mathcal{W}_\sigma = (\mathcal{W}, I_{\mathcal{W}_\sigma})$ through soft-substructure $\tilde{\mathcal{W}}_\sigma = (\psi_{\mathcal{W}}(\mathcal{W}), \phi^\sigma)$, by instead maximising $p(\mathcal{W}_\sigma | \tilde{\mathcal{W}}_\sigma)$. But $\mathcal{W}_\sigma$ is latent and our agent still only has access to observations from $\mathcal{X}_\sigma^n$ and $\mathcal{X}_\sigma^m$, so how can $p(\mathcal{W}_\sigma | \tilde{\mathcal{W}}_\sigma)$ be maximised? This relies on existence of mappings from the observational structures, assumed to be substructures of a broader structure over all possible observations, into corresponding substructures of a latent world-state space, that

⁶Some of the properties of this theory will be explored in Section 5.4 under the lens of disentanglement. But proper investigation into what this theory might be and whether it is reasonable to assume it exists may be best suited to metaphysical debate and merits its own thesis (or theses) (Domingos, 2015).

preserve the original semantics. Such a mapping is called an *elementary embedding* (Hodges, 1993).

Definition 5.6 (Elementary embedding) *An elementary embedding⁷ from a logical structure $\mathcal{A}_\sigma$ to a logical structure $\mathcal{B}_\sigma$ is an injective mapping $e_{\mathcal{B}_\sigma}^{\mathcal{A}_\sigma} : \mathcal{A}_\sigma \rightarrow \mathcal{B}_\sigma$ such that for every formula $\Box(V_1, \dots, V_m)$, with variables $(V_1, \dots, V_m)$, in the signature σ and for any $(a_1, \dots, a_m) \in \mathcal{A}_m$:*

$$\mathcal{A}_\sigma \models \Box(a_1, \dots, a_m) \leftrightarrow \mathcal{B}_\sigma \models \Box(e_{\mathcal{B}_\sigma}^{\mathcal{A}_\sigma}(a_1), \dots, e_{\mathcal{B}_\sigma}^{\mathcal{A}_\sigma}(a_m)). \quad (5.28)$$

If an elementary embedding $e_{\mathcal{W}_\sigma}^{\mathcal{X}_\sigma}$ exists between $\mathcal{X}_\sigma$ and $\mathcal{W}_\sigma$, then we can be sure that both structures are equivalent from a semantic perspective. However, as it stands, we cannot employ Corollary 5.2 to provide coherence guarantees between observational structures. This is because we have introduced a new domain and so cannot assume that all structures share a single domain encoder, as was required in Theorem 5.1. Our aim is to show that an elementary embedding can still enable us to employ Corollary 5.2 for coherence guarantees. Towards this end, we must show that given a suitable mapping, structure support can be transferred between domains.

Theorem 5.2 *Let $\mathcal{X}_\sigma^n$ be a structure and $e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$ be a surjective elementary embedding from $\mathcal{X}_\sigma^n$ to $\mathcal{W}_\sigma^n$, and let $\mathcal{W}_\sigma^n$ be a substructure of a structure $\mathcal{W}_\sigma$. With $\phi^\sigma = \{ \phi_r \mid r \in \sigma \}$, let $\tilde{\mathcal{X}}_\sigma^n = (\psi_{\mathcal{X}}(\mathcal{X}_n), \phi^\sigma)$ be a soft-substructure over domain $\psi_{\mathcal{X}}(\mathcal{X}_n)$, $\tilde{\mathcal{W}}_\sigma^n = (\psi_{\mathcal{W}}(\mathcal{W}_n), \phi^\sigma)$ be a soft-substructure over domain $\psi_{\mathcal{W}}(\mathcal{W}_n)$, and $\tilde{\mathcal{W}}_\sigma = (\psi_{\mathcal{W}}(\mathcal{W}), \phi^\sigma)$ be a soft-substructure over domain $\psi_{\mathcal{W}}(\mathcal{W})$. If $\psi_{\mathcal{X}} = \psi_{\mathcal{W}} \circ e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$, where $\circ$ denotes function composition, then*

$$p(\mathcal{W}_\sigma | \tilde{\mathcal{W}}_\sigma) = 1 - \epsilon \rightarrow p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n) \geq 1 - \epsilon.$$

(Proof) By Theorem 5.1 we know that $p(\mathcal{W}_\sigma^n | \tilde{\mathcal{W}}_\sigma^n) \geq p(\mathcal{W}_\sigma | \tilde{\mathcal{W}}_\sigma)$. The proof then follows from showing that through the substitution $\psi_{\mathcal{X}} = \psi_{\mathcal{W}} \circ e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$, $p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n) = p(\mathcal{W}_\sigma^n | \tilde{\mathcal{W}}_\sigma^n)$. Let us start

⁷As in Definition 5.5, this definition is scoped to signatures comprising only relation symbols and no functions nor constants, but can be extended to include these, and the following theorems will still hold.

by writing out the structure supports for $\mathcal{W}_\sigma$ and $\mathcal{X}_\sigma^n$,

$$p(\mathcal{W}_\sigma | \tilde{\mathcal{W}}_\sigma) = \prod_{r \in \sigma} \prod_{O \in \mathcal{W}^{\text{ar}}(r)} (\phi_r(\psi_{\mathcal{W}}(O)))^{\gamma_{\mathcal{W}_\sigma}^{r,O}} \cdot (1 - \phi_r(\psi_{\mathcal{W}}(O)))^{1 - \gamma_{\mathcal{W}_\sigma}^{r,O}} \quad (5.29)$$

$$p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n) = \prod_{r \in \sigma} \prod_{O \in \mathcal{X}_n^{\text{ar}}(r)} (\phi_r(\psi_{\mathcal{X}}(O)))^{\gamma_{\mathcal{X}_\sigma^n}^{r,O}} \cdot (1 - \phi_r(\psi_{\mathcal{X}}(O)))^{1 - \gamma_{\mathcal{X}_\sigma^n}^{r,O}}. \quad (5.30)$$

We can see the differences that must be accounted for between Equation 5.29 and Equation 5.30 are in: the encoders $\psi_{\mathcal{W}}$ and $\psi_{\mathcal{X}}$, the interpretation labels $\gamma_{\mathcal{W}_\sigma}^{r,O}$ and $\gamma_{\mathcal{X}_\sigma^n}^{r,O}$, and the domains $\mathcal{W}$ and $\mathcal{X}_n$. Substituting in $\psi_{\mathcal{X}} = \psi_{\mathcal{W}} \circ e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$ in Equation 5.30 we get:

$$p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n) = \prod_{r \in \sigma} \prod_{O \in \mathcal{X}_n^{\text{ar}}(r)} (\phi_r(\psi_{\mathcal{W}}(O_e)))^{\gamma_{\mathcal{W}_\sigma}^{r,O_e}} \cdot (1 - \phi_r(\psi_{\mathcal{W}}(O_e)))^{1 - \gamma_{\mathcal{W}_\sigma}^{r,O_e}}, \quad (5.31)$$

where for the sake of notational clarity $O_e \doteq e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}(O)$. Note that the new Equation 5.31 also takes care of the difference in interpretation labels, via switching from $\gamma_{\mathcal{X}_\sigma^n}^{r,O}$ to $\gamma_{\mathcal{W}_\sigma}^{r,O_e}$. This is because, by definition, $e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$ is an elementary embedding from $\mathcal{X}_\sigma^n$ to $\mathcal{W}_\sigma^n$ and $\mathcal{W}_\sigma^n$ is a substructure of $\mathcal{W}_\sigma$, which means that for all $r \in \sigma$ and $O \in \mathcal{X}_n^{\text{ar}}(r)$

$$\underbrace{O \in I_{\mathcal{X}_\sigma^n}(r) \leftrightarrow O_e \in I_{\mathcal{W}_\sigma^n}(r)}_{\text{via elementary embedding } e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}} \leftrightarrow O_e \in I_{\mathcal{W}_\sigma}(r). \quad (5.32)$$

via substructure relationship $\mathcal{W}_\sigma^n \subseteq \mathcal{W}_\sigma$

By transitivity of $\leftrightarrow$, we have that $O \in I_{\mathcal{X}_\sigma^n}(r)$ if and only if $O_e \in I_{\mathcal{W}_\sigma}(r)$. Hence $\gamma_{\mathcal{X}_\sigma^n}^{r,O}$ and $\gamma_{\mathcal{W}_\sigma}^{r,O_e}$ have the same values. The remaining task is to justify that

$$p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n) = p(\mathcal{W}_\sigma^n | \tilde{\mathcal{W}}_\sigma^n). \quad (5.33)$$

Considering Equation 5.31, proving Equation 5.33 is equivalent to proving the following Equation 5.34:

$$\begin{aligned} \prod_{r \in \sigma} \prod_{O \in \mathcal{X}_n^{\text{ar}}(r)} (\phi_r(\psi_{\mathcal{W}}(O_e)))^{\gamma_{\mathcal{W}_\sigma}^{r,O_e}} \cdot (1 - \phi_r(\psi_{\mathcal{W}}(O_e)))^{1 - \gamma_{\mathcal{W}_\sigma}^{r,O_e}} \\ = \prod_{r \in \sigma} \prod_{O \in \mathcal{W}_\sigma^{\text{ar}}(r)} (\phi_r(\psi_{\mathcal{W}}(O)))^{\gamma_{\mathcal{W}_\sigma^n}^{r,O}} \cdot (1 - \phi_r(\psi_{\mathcal{W}}(O)))^{1 - \gamma_{\mathcal{W}_\sigma^n}^{r,O}}, \end{aligned} \quad (5.34)$$

Firstly, as $\mathcal{W}_\sigma^n \subseteq \mathcal{W}_\sigma$, the interpretation labels for $\mathcal{W}_\sigma^n$ will match that of $\mathcal{W}_\sigma$, hence $\gamma_{\mathcal{W}_\sigma}^{r,O} = \gamma_{\mathcal{W}_\sigma^n}^{r,O}$,

giving

$$\begin{aligned} \prod_{r \in \sigma} \prod_{O \in \mathcal{X}_n^{\text{ar}(r)}} (\phi_r(\psi_{\mathcal{W}}(O_e)))^{\gamma_{\mathcal{W}_\sigma^n}^{r, O_e}} \cdot (1 - \phi_r(\psi_{\mathcal{W}}(O_e)))^{1 - \gamma_{\mathcal{W}_\sigma^n}^{r, O_e}} \\ = \prod_{r \in \sigma} \prod_{O \in \mathcal{W}_n^{\text{ar}(r)}} (\phi_r(\psi_{\mathcal{W}}(O)))^{\gamma_{\mathcal{W}_\sigma^n}^{r, O}} \cdot (1 - \phi_r(\psi_{\mathcal{W}}(O)))^{1 - \gamma_{\mathcal{W}_\sigma^n}^{r, O}}. \end{aligned} \quad (5.35)$$

Let us denote the sets of tuples

$$\mathcal{K}_\sigma^{\mathcal{X}_n} = \{ (r, O) \mid r \in \sigma, O \in \mathcal{X}_n^{\text{ar}(r)} \} \quad (5.36)$$

and

$$\mathcal{K}_\sigma^{\mathcal{W}_n} = \{ (r, O) \mid r \in \sigma, O \in \mathcal{W}_n^{\text{ar}(r)} \}. \quad (5.37)$$

To ensure equality of Equation 5.35 we must show that: there is a bijective mapping between $\mathcal{K}_\sigma^{\mathcal{X}_n}$ and $\mathcal{K}_\sigma^{\mathcal{W}_n}$, in which case the number of terms in each product will match; and that the corresponding interpretation labels are equal *i.e.* $\gamma_{\mathcal{W}_\sigma^n}^{r, O_e} = \gamma_{\mathcal{W}_\sigma^n}^{r, O}$, for all $(r, O) \in \mathcal{K}_\sigma^{\mathcal{X}_n}$. This requires us to show that for each $r \in \sigma$ and $O \in \mathcal{X}_n^{\text{ar}(r)}$, each $e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_n^{\text{ar}(r)}}(O)$ matches exactly one $O \in \mathcal{W}_n$ and vice versa. Now, $e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_n^{\text{ar}(r)}}$ is defined as a *surjective* elementary embedding, meaning that $\mathcal{X}_\sigma^n$ and $\mathcal{W}_\sigma^n$ are isomorphic⁸. As such, we have that $|\mathcal{X}_\sigma^n| = |\mathcal{W}_\sigma^n|$ and that the semantics of $\mathcal{X}_\sigma^n$ and $\mathcal{W}_\sigma^n$ are preserved under $e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$. Specifically, by condition (1) in Definition 5.6, the interpretations of $\mathcal{X}_\sigma^n$ and $\mathcal{W}_\sigma^n$ are equivalent under the elementary embedding, and so $\gamma_{\mathcal{W}_\sigma^n}^{r, O_e} = \gamma_{\mathcal{W}_\sigma^n}^{r, O}$, for all $(r, O) \in \mathcal{K}_\sigma^{\mathcal{X}_n}$. We can conclude that both products in Equation 5.35 match, and the overall structure supports are equal. Finally, since by Theorem 5.1 $p(\mathcal{W}_\sigma | \tilde{\mathcal{W}}_\sigma) = 1 - \epsilon \rightarrow p(\mathcal{W}_\sigma^n | \tilde{\mathcal{W}}_\sigma^n) \geq 1 - \epsilon$, then by Equation 5.33, $p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n) \geq 1 - \epsilon$ follows. This completes the proof.

Theorem 5.2 states that $p(\mathcal{W}_\sigma | \tilde{\mathcal{W}}_\sigma)$ acts as a lower bound on $p(\mathcal{X}_\sigma^n | \tilde{\mathcal{X}}_\sigma^n)$ when applying the substitution $\psi_{\mathcal{X}} = \psi_{\mathcal{W}} \circ e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$. Note that the isomorphism between $\mathcal{W}_\sigma^n$ and $\mathcal{X}_\sigma^n$ means that the semantics that apply to $\mathcal{X}_\sigma^n$ are also present in $\mathcal{W}_\sigma^n$, and vice versa. The key to realising the potential of Theorem 5.2 is then in the value of $\mathcal{W}_\sigma^n$ as an abstraction over $\mathcal{X}_\sigma^n$, *i.e.* in the respective representation of each domain element of $\mathcal{W}_\sigma^n$ compared to the corresponding domain

⁸Whether it is reasonable to make the bijective assumption is a matter for debate and is discussed in (Higgins et al., 2019). Although it is technically possible to change world-states whilst keeping observations the same (*e.g.* occlusion), this thesis adopts the bijective assumption, knowing that this limits observations to those that fully define the world-state. Another way to look at this assumption is that it states that anything the observation cannot tell us about the world-state is simply *unknown* and to know an unknown property of the world-state, this would necessarily change the observation.

element of $\mathcal{X}_\sigma^n$. Concretely, it may be easier to reason (interpolate/extrapolate) about $\mathcal{W}_\sigma$ from $\mathcal{W}_\sigma^n$, than it is to reason about $\mathcal{X}_\sigma$ from $\mathcal{X}_\sigma^n$. However, the agent cannot directly access $\mathcal{W}_\sigma^n$, only $\mathcal{X}_\sigma^n$, and so must instead infer the properties of $\mathcal{W}_\sigma^n$ from $\mathcal{X}_\sigma^n$. In this way, the encoding substitution $\psi_{\mathcal{X}} = \psi_{\mathcal{W}} \circ e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$ can be seen as the agent's attempt to make sense of its observations with respect to an abstract understanding that it builds about its world. Next, we shall see how Theorem 5.2 can be used to provide coherence guarantees between observational structures, using a similar corollary to Corollary 5.2.

Corollary 5.3 *Let $\mathcal{T}$ be a theory over a signature σ , assume we have structures $\mathcal{W}_\sigma$, $\mathcal{W}_\sigma^n$, $\mathcal{W}_\sigma^m$, where $\mathcal{W}_\sigma^n \subseteq \mathcal{W}_\sigma$ and $\mathcal{W}_\sigma^m \subseteq \mathcal{W}_\sigma$; and soft-substructures $\tilde{\mathcal{W}}_\sigma = (\psi_{\mathcal{W}}(\mathcal{W}), \phi^\sigma)$, $\tilde{\mathcal{W}}_\sigma^n = (\psi_{\mathcal{W}}(\mathcal{W}_n), \phi^\sigma)$, $\tilde{\mathcal{W}}_\sigma^m = (\psi_{\mathcal{W}}(\mathcal{W}_m), \phi^\sigma)$, $\tilde{\mathcal{X}}_\sigma^n = (\psi_{\mathcal{X}}^n(\mathcal{X}_n), \phi^\sigma)$, and $\tilde{\mathcal{X}}_\sigma^m = (\psi_{\mathcal{X}}^m(\mathcal{X}_m), \phi^\sigma)$, where $\psi_{\mathcal{X}}^n = \psi_{\mathcal{W}} \circ e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$ and $\psi_{\mathcal{X}}^m = \psi_{\mathcal{W}} \circ e_{\mathcal{W}_\sigma^m}^{\mathcal{X}_\sigma^m}$. Additionally, let $e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n}$ be a surjective elementary embedding from $\mathcal{X}_\sigma^n$ to structure $\mathcal{W}_\sigma^n$ and similarly let $e_{\mathcal{W}_\sigma^m}^{\mathcal{X}_\sigma^m}$ be a surjective elementary embedding from $\mathcal{X}_\sigma^m$ to structure $\mathcal{W}_\sigma^m$. Then if $\mathcal{X}_\sigma^n \in \mathcal{M}_{\mathcal{X}_n}^\mathcal{T}$ and $\mathcal{X}_\sigma^m \in \mathcal{M}_{\mathcal{X}_m}^\mathcal{T}$, and $p(\mathcal{W}_\sigma | \tilde{\mathcal{W}}_\sigma) = 1 - \epsilon$, then $\tilde{\mathcal{X}}_\sigma^n$ and $\tilde{\mathcal{X}}_\sigma^m$ will be at least ϵ -coherent.*

(Proof Sketch) The proof follows the same reasoning as that of Corollary 5.2, but with Theorem 5.2 in place of Theorem 5.1.

Corollary 5.3 is a key result, as it shows that it is possible to obtain coherence guarantees by leveraging elementary embeddings to a different domain, through appropriate encoder substitutions. But, in a practical setting a learner would only have access to observation sets (our $\mathcal{X}_n$ and $\mathcal{X}_m$). So is it reasonable to assume that the agent will learn to represent this data in a manner that aligns with $\psi_{\mathcal{W}}$, ultimately obtaining the required encoder substitutions for coherence? Moreover, can we leverage the assumption that same theory and signature apply to all structures and substructures, and the fact that ϕ^σ is shared between all soft-substructures? The next section will explore these questions.

5.4 Disentangled soft-structures

In a practical setting, a lifelong learning agent may only have access to sets of observations from $\mathcal{X}$. However, as discussed in see Section 2.4.1, by adopting disentanglement we can

think of these observations as being generated from a background system of world-states, via $f_{\mathcal{W} \rightarrow \mathcal{X}}$. In effect, $\mathcal{X}$ is evidence of a world-state space that carries certain regularities (or symmetries/invariances) and through learning disentangled representations an agent may be able to expose these regularities. Parallel to this idea, Corollary 5.3 demonstrates that the agent can learn coherent soft-substructures, providing it employs suitable encoders with respect to $\mathcal{X}$, $\mathcal{W}$, and $\mathcal{Z}$. Since the agent does not have direct access to domain $\mathcal{W}_n$ (or more broadly domain $\mathcal{W}$ where we had that $\mathcal{W}_n \subseteq \mathcal{W}$) it is not clear how these encoders can be learned. But the end result is quite clear - either the agent encodes directly from $\mathcal{X}$ to $\mathcal{Z}$, or (if it had access) it could go through a roundabout encoding from $\mathcal{X}$ to $\mathcal{W}$ and then from $\mathcal{W}$ to $\mathcal{Z}$. The key point is that the resulting encodings should match. This is exactly the goal of disentanglement, in which the internal representation $\mathcal{Z}$ should exhibit the same invariance properties as are present in $\mathcal{W}$, and such invariances should be extracted, through appropriate bias, from $\mathcal{X}$. Going back to disentanglement, as defined by Higgins et al. (2019), this modelling provides some light into the possible target encoding that Theorem 5.2 and the resulting corollaries rely upon. Specifically, the target encoder for a subset of observations $\mathcal{X}_n$ may be

$$\psi_{\mathcal{X}_n} = \psi_{\mathcal{W}} \circ e_{\mathcal{W}_\sigma^n}^{\mathcal{X}_\sigma^n} = f_{\mathcal{W} \rightarrow \mathcal{Z}} \circ f_{\mathcal{W} \rightarrow \mathcal{X}}^{-1}, \quad (5.38)$$

where $f_{\mathcal{W} \rightarrow \mathcal{X}}$ and $f_{\mathcal{W} \rightarrow \mathcal{Z}}$ are the $\mathcal{W} \rightarrow \mathcal{X}$ and $\mathcal{W} \rightarrow \mathcal{Z}$ functions, as defined in Equation 2.29 and Equation 2.32, respectively⁹. From this perspective, in working to obtain $\psi_{\mathcal{X}_n}$, an agent is in fact attempting to disentangle the EFs from a given set of data.

In Chapter 4 we saw how incorporating weak relational supervision using relation-decoders helped to disentangle the EFs at play in a given dataset, leading to clearer representations where possible ambiguities in $\mathcal{X}$ were resolved and stronger disentanglement was achieved for non-visual EFs. We now attempt to explain this within the formal framework of soft-substructures. Assuming that $\mathcal{W}_\sigma$ is governed by a theory $\mathcal{T}$ and that $\mathcal{X}$ is generated from $\mathcal{W}$, then it seems reasonable to assume that $\mathcal{X}$ is also governed by the same theory. Therefore, ensuring ϕ^σ 's beliefs are in agreement with a ground-truth structure $\mathcal{X}_\sigma^n$, can also mean ensuring that ϕ^σ 's beliefs are in agreement with the theory $\mathcal{T}$ that $\mathcal{X}_\sigma^n$ provides evidence for, and may well be a model of¹⁰. Now, if $\mathcal{T}$ governs the full $\mathcal{X}$, of which $\mathcal{X}_n \subseteq \mathcal{X}$, then learning ϕ^σ in a

⁹For $f_{\mathcal{W} \rightarrow \mathcal{X}}$ to have an inverse depends on whether it is invertible. In Section 5.3 this thesis assumed that a bijective mapping from $\mathcal{X}$ to $\mathcal{W}$ exists, in which case an inverse mapping is possible. Higgins et al. (2019) do not default to invertible mappings but refer to active sensing (Soatto, 2010) as a means to acquire invertibility.

¹⁰Sometimes additional observations may be necessary to complete the model, but the existence of the theory

consistent manner with respect to $\mathcal{T}$ may enable extrapolation and/or interpolation of beliefs beyond the present $\mathcal{X}_n$ observations, to regions of $\mathcal{Z}$ that, although not yet encoded to, could conceivably be encoded to from $\mathcal{X}$. As discussed in Section 5.3, this would have the desirable background effect of maximising $p(\mathcal{W}_\sigma|\tilde{\mathcal{W}}_\sigma)$, which can in turn encourage coherence between new observations through Equation 5.38. The following section will build on these ideas by more formally incorporating disentanglement into soft-structures/substructures.

5.4.1 A disentangled view of relations and theories

Previously we saw that Explanatory Factors (EFs) can be independently comparable (Equation 4.1), meaning that relationships on or between objects with respect to one EF should not be affected by the value of another EF. We can express this using the same terminology that is employed in Higgins et al. (2019)'s definition of disentanglement (see Section 2.4.1). Let $\mathcal{R}$ be a set of relations that is partitioned into subsets $\mathcal{R}_\nu$ for $\nu \in \{\nu_1, \dots, \nu_n\}$, where each ν_i denotes a particular EF. Further, let each $\mathcal{R}_\nu$ be partitioned into subsets with the same arity, $\mathcal{R}_\nu = \{\mathcal{R}_\nu^1, \mathcal{R}_\nu^2, \dots\}$ for sets of unary, binary, and so on, relations. We can then define a *disentangled relations* according to their invariance properties.

Definition 5.7 (Disentangled relation) $r \in \mathcal{R}_{\nu_i}^k$ is a disentangled relation with respect to EF ν_i if it is invariant under group actions $\cdot_\alpha^j, \forall j \neq i$. For instance, a binary relation ($k = 2$) is disentangled with respect to ν_i if it satisfies the following relation invariance condition:

$$(r(x_1, x_2) \leftrightarrow r(g \cdot_\alpha^j x_1, x_2) \leftrightarrow r(x_1, g \cdot_\alpha^j x_2) \leftrightarrow r(g \cdot_\alpha^j x_1, g \cdot_\alpha^j x_2)) \quad \forall j \neq i, \quad (5.39)$$

where $g \in \mathcal{G}_j$ is a symmetry transformation and $\cdot_\alpha^j$ is a subgroup action for ν_j .

A disentangled relation is in effect a relation that compares a specific representational property (a specific EF). Grouping disentangled relations together according to the respective EFs that they compare, we can define *disentangled family of relations*.

Definition 5.8 (Disentangled family of relations) Let σ be a signature of the theory $\mathcal{T}$. A subset of relation symbols from σ , denoted σ^ν , is called a disentangled family of relations for a particular EF, ν , if each relation symbol in σ^ν is disentangled with respect to ν .

can already be hypothesized.

Providing relations group into disentangled families, we can then define a *disentangled theory* as a theory whose sentences are strictly composed of disentangled families of relations.

Definition 5.9 (Disentangled theory) *Let $\mathcal{T}$ be a theory with signature σ . A subset of $\mathcal{T}$, denoted $\mathcal{T}^\nu$, is called a disentangled theory with respect to a particular EF, ν , if all sentences in $\mathcal{T}^\nu$ are composed using only relations in σ^ν .*

In essence, $\mathcal{T}^\nu$ is a theory that exclusively characterises the semantics of a particular EF. Decomposing theories and signatures into disentangled components is beneficial to soft-structure learning, as it provides a way to break up a given soft-structure $\tilde{\mathcal{S}}_\sigma$ into *disentangled soft-structures*.

Definition 5.10 (Disentangled soft-structure) *A disentangled soft-structure is a soft-structure $\tilde{\mathcal{S}}_{\sigma^\nu} = (\psi_{\mathcal{S}}(\mathcal{S}), \phi^{\sigma^\nu})$, where $\phi^{\sigma^\nu} = \{ \phi_r \mid r \in \sigma^\nu \}$, for a particular EF, ν .*

From a soft-structure $\tilde{\mathcal{S}}_\sigma$, whose signature comprises n disentangled families of relations $\sigma = \bigcup_{i=1}^n \sigma^{\nu_i}$, we can obtain a disentangled soft-structure for each ν_i . To merge disentangled soft-structures with disentanglement, as defined by Higgins et al. (2019), let $\mathcal{Z}$ be a vector space that decomposes into $\mathcal{Z} = \mathcal{Z}_1 \oplus \mathcal{Z}_2 \oplus \dots \oplus \mathcal{Z}_n$ (where $\oplus$ denotes the direct sum of vector spaces) such that each $\mathcal{Z}_i$ is invariant under the subgroup action $\cdot_\alpha^j$ for all $j \neq i$. Providing that each ϕ^{σ^ν} 's domain is a corresponding component $\mathcal{Z}_\nu$, then a soft-structure $\tilde{\mathcal{S}}_\sigma$ can be decomposed into a set of disentangled soft-structures of the form

$$\tilde{\mathcal{S}}_{\sigma^\nu} = (\psi_{\mathcal{S}}(\mathcal{S}), \phi^{\sigma^\nu}) \quad (5.40)$$

$$= (\mathcal{Z}'_\nu, \sigma^\nu) \quad (5.41)$$

where $\mathcal{Z}'_\nu \subseteq \mathcal{Z}_\nu$. It is then clear that, providing signatures can be broken up into disentangled families of relations, the notion of ϵ -consistency of a soft-structure with regards to a possibly broad theory, covering a wide set of relations, can similarly be broken up into sub-tasks of achieving disentangled soft-structure consistency with each disentangled theory $\mathcal{T}^\nu$. Let us now make an observation.

Assume we have two unary relations r_1 and r_2 such that

$$\forall A \ (r_1(A) \rightarrow \neg r_2(A)) \quad (5.42)$$

and take ϕ_{r_1} and ϕ_{r_2} to be associated relation-decoders. Furthermore, assume we are interested in a domain $\mathcal{S}$, and define $\lambda_{r_i} \in (0, 1)$ to be a threshold for relation-decoder ϕ_{r_i} , for which $\phi_{r_i}(\cdot) < \lambda_{r_i}$ signifies false and $\phi_{r_i}(\cdot) \geq \lambda_{r_i}$ signifies truth. Given the implication rule (Equation 5.42), we would expect a soft-structure to exhibit the following consistency property:

$$\forall s \in \mathcal{S} \left(\phi_{r_1}(\psi_{\mathcal{S}}(s)) \geq \lambda_{r_1} \rightarrow \phi_{r_2}(\psi_{\mathcal{S}}(s)) < \lambda_{r_2} \right). \quad (5.43)$$

When $\mathcal{Z}$ is a vector space then it decomposes into independent vector subspaces. In which case, it can be possible to violate Equation 5.43 under certain assumptions. To see this, let us define the following regions of interest with respect to a relation-decoder ϕ_{r_i} , following (Gutiérrez-Basulto and Schockaert, 2018):

$$\mathcal{D}_{\phi_{r_i}}^{\top} = \{ \mathbf{z}_j \mid \phi_{r_i}(\mathbf{z}_j) \geq \lambda_{r_i}, \mathbf{z}_j \in \mathcal{Z} \} \quad (5.44)$$

$$\mathcal{D}_{\phi_{r_i}}^{\perp} = \{ \mathbf{z}_j \mid \phi_{r_i}(\mathbf{z}_j) < \lambda_{r_i}, \mathbf{z}_j \in \mathcal{Z} \}, \quad (5.45)$$

where, under λ_{r_i} , $\mathcal{D}_{\phi_{r_i}}^{\top}$ gives the region of $\mathcal{Z}$ where ϕ_{r_i} is considered true and $\mathcal{D}_{\phi_{r_i}}^{\perp}$ where it is considered false. To satisfy Equation 5.43 we require that $\mathcal{D}_{\phi_{r_1}}^{\top} \cap \mathcal{D}_{\phi_{r_2}}^{\top} = \emptyset$, but given the independence of $\mathcal{Z}_{\nu_1}$ and $\mathcal{Z}_{\nu_2}$, this cannot be the case unless either (or both) $\mathcal{D}_{\phi_{r_1}}^{\top} = \emptyset$ and $\mathcal{D}_{\phi_{r_2}}^{\top} = \emptyset$. This is a strong restriction with perhaps unintended side-effects, for instance if $\mathcal{D}_{\phi_{r_1}}^{\top} = \emptyset$ then this implies that $\forall A (\neg r_1(A))$ (*i.e.* r_1 is never true). Therefore, assuming a disentangled $\mathcal{Z}$ vector space, relation-decoders that approximate relations in a common disentangled family of relations must share domains. This detail will prove useful in the design of a relation-decoder that can facilitate consistency and coherence between disentangled soft-substructures and will be explored further in Section 6.1.

To summarise this section, the view that has been presented extends disentanglement to the proposed soft-structure formalism, where an EF can be characterised by a corresponding disentangled theory, and a soft-structure defined over a disentangled latent space can be broken up into a set of disentangled soft-structures. The following chapter will bring the focus to practical learning of ϵ -consistent and ϵ -coherent disentangled soft-structures, whose signatures are composed of disentangled families of relations.

Chapter 6

Learning consistent and coherent disentangled soft-structures

So far we have built theoretical foundations for formally describing the learning of consistent and coherent soft-structures. More specifically, learning to disentangle data representations using disentangled relations can be modelled as a disentangled soft-structure learning task. One of the aims of this thesis is to take a step towards transferability of learned representations. Using our new terminology, core to this goal are the formal notions of ϵ -consistency and ϵ -coherence, in which FOL-like transferability of relation-decoders depends on the model-weight (Equation 5.9) of the soft-structures that they are part of. The focus of this chapter is to bring the discussion back to a practical level, by presenting an architecture that learns consistent and coherent disentangled soft-structures from images. Corollary 5.3 and Equation 5.38, presented in the previous chapter, together show that under assumptions of disentanglement it can be possible to produce coherence guarantees between suitably learned soft-substructures. Inspired by this, this chapter first presents a set of design-criteria that acts as the basis for the construction of a consistent and coherent relation-decoder implementation. Then, based upon this specification, a novel class of Dynamic Comparator (DC) relation-decoders is proposed, together with an empirical investigation of its behaviour in learning disentangled soft-structures. Next, bringing the focus back to that of transferable knowledge, a proposed Partial Relation Transfer (PRT) experiment will be introduced and conducted, in which the transferability of learned DC relation-decoders between a source and target task can be evaluated and compared against other candidate relation-decoder implementations. Within this investigation, an empirical

Table 6.1: Characteristic sentences that a binary relation r may adhere to (Trouillon et al., 2019). Uppercase letters denote FOL variables and lowercase letters denote FOL constants.

Characteristic	Sentence	Example
Reflexivity	$\forall A, (r(A, A)).$	$\leq$ is reflexive, because for any a , $a \leq a$.
Irreflexivity	$\forall A, \neg (r(A, A)).$	$<$ is irreflexive, because for any a , it is not the case that $a < a$.
Symmetry	$\forall A, B, (r(A, B) \Rightarrow r(B, A)).$	$=$ is symmetric, because if $a = b$, then $b = a$.
Antisymmetry	$\forall A, B, (r(A, B) \wedge r(B, A) \Rightarrow a = b).$	$\leq$ is antisymmetric, because if $a \leq b$ and $b \leq a$, then $a = b$.
Transitivity	$\forall A, B, C, (r(A, B) \wedge r(B, C) \Rightarrow r(A, C)).$	$\leq$ is transitive, because if $a \leq b$ and $b \leq c$, then $a \leq c$.
Anti-transitivity	$\forall A, B, C, (r(A, B) \wedge r(B, C) \Rightarrow \neg r(A, C)).$	The successor relation is transitive, because if a is the successor of b and b is the successor of c , then a is not the successor of c .

ϵ -consistency and ϵ -coherence loss will be presented, which will enable us to see the relationship between ϵ -consistency, ϵ -coherence, and FOL-like relation transferability.

6.1 Designing a consistent and coherent relation-decoder for disentangled soft-structures

The primary goal of this section is to produce a relation-decoder that can facilitate learning consistent and coherent disentangled soft-structures from data, where the signature decomposes into $\sigma = \bigcup_{i=1}^n \sigma^{\nu_i}$ for n EFs. This will be achieved by identifying a set of design-criteria underpinned by the framework presented in Chapter 5, that will form the basis of this construction. Then, based upon this specification, a novel class of DC relation-decoders are proposed, together with an empirical investigation of its behaviour in disentangled soft-structure learning.

6.1.1 Design principles

When training a structure support task with respect to a model of a theory $\mathcal{T}$, getting close to 100% test accuracy for relation prediction will usually mean that the relation-decoders have managed to achieve good ϵ -consistency with respect to $\mathcal{T}$ for the test-set. This is because the interpretation labels are from a model of a theory $\mathcal{T}$. For this to be possible, we require a relation-decoder that can capture the behaviour of the relation it approximates, and the ways

in which its relation and others interact, as described by $\mathcal{T}$, at least for the test-set. This gives us two types of sentences that a set of relation-decoders can be required to capture, referred to in this thesis as sentences pertaining to *individual* relations and those that describe semantics *across* relations. For the ‘individual’ case we have the following definition:

Definition 6.1 (Consistency-individual sentence) *A consistency-individual sentence is a sentence that describes the properties of a single relation, expressed as a FOL sentence about only that relation.*

For example a consistency-individual sentence for a relation r could be:

$$\forall A (r(A, A)), \quad (6.1)$$

which states that r is a symmetric relation. The main examples of consistency-individual sentences for binary relations, which this chapter primarily focuses on, are given in Table 6.1. Similarly, for the ‘across’ case we have:

Definition 6.2 (Consistency-across sentence) *A consistency-across sentence is a sentence that describes the properties over a set relations, expressed as a FOL sentence about multiple relations.*

For example a consistency-across sentence for a unary relation r_1 and binary relation r_2 could be:

$$\forall A, B (r_1(A) \rightarrow \neg r_2(A, B)), \quad (6.2)$$

which states for instance that any a for which r_1 is true, cannot be related to any b through the relation r_2 .

We will refer to **D1** as the design-criteria for which *a relation-decoder should be able to express all consistency-individual sentences*; and **D2** as the design-criteria for which *the relation-decoder should be able to model all consistency-across sentences*. To satisfy both of these criteria for any set of individual/across sentences is a strong ideal and perhaps intractable to prove or to achieve empirically. This chapter will focus on a specific case, where for consistency-individual sentences is defined in terms of the sentences provided in Table 6.1 for binary relations; and the

consistency-across sentences will be scoped to a theory of ordinality, specified in Appendix C, over the ordinal relations listed in Equation 5.1.

Satisfying **D1** and **D2** does not necessarily demonstrate that the relation-decoders will have coherent beliefs when applied to new domains, as part of other soft-substructures, because the new dataset encodings may not exactly match the encodings of a previous dataset. This leads us to the next design-criteria, denoted as **D3**, for which *the relation-decoder should generalise consistency (i.e. be coherent) across $\mathcal{Z}$* . At the extreme, in achieving ϵ -consistency for a soft-substructure, by Theorem 5.1, we would like our relation-decoders to generalise consistency to the parent soft-structure, whose domain is by definition all of $\mathcal{Z}$. So how can this generalisation be encouraged at the relation-decoder level? As the focus of this chapter is on disentangled soft-substructures/structures, whose signatures are composed of disentangled families of relations, this question will be scoped to ϵ -coherence between disentangled soft-substructures/structures. We know from Section 5.4.1 that the relation-decoders corresponding to relations in a given disentangled family of relations must all share the same orthogonal subspace of $\mathcal{Z}$, and that this should be different from the orthogonal subspaces of other disentangled families of relations. Therefore, the final design-criteria, denoted as **D4**, states that: *the relation-decoder should be able to select specific orthogonal subspaces of $\mathcal{Z}$ and ignore all others*. Note that these may have to be identified from data, purely through interpretation labelling, and so the relation-decoders will have to learn to identify the relevant latent subspaces and ensure agreement (independence) with other relation-decoders that approximate relations in the same (different) disentangled family of relations.

D4 presents an additional opportunity for practical ϵ -coherence. From Equation 5.38, we know that under assumptions of a generative process from a disentangled world-state space, which decomposes into specific EF components, we can define a class of encoders that essentially disentangle back out the EFs from observed data. However, take $\mathcal{Z} = \mathcal{Z}_1 \oplus \mathcal{Z}_2 \oplus \dots \oplus \mathcal{Z}_n$, where $\oplus$ denotes the direct sum, to be a disentangled space for n EFs. We can define permutations of the components and still have a valid disentanglement (Higgins et al., 2019). This can be problematic in practice because it means two separately trained encoders (*e.g.* from a VAE) can successfully disentangle two data domains generated from shared EFs, but encode the EFs into conflicting spaces within $\mathcal{Z}$. By incorporating relation-decoders that have ‘selected’ the appropriate subspace for encoding a specific EF, they can facilitate the coordination among a

set of encoders, guiding them towards a consensus in the encoding of the EFs with respect to $\mathcal{Z}$. Moreover, as each relation-decoder determines *how* one or more data-embeddings should be positioned in order to produce a given belief about it/them, the relation-decoders provide additional bias for the encoders.

6.1.2 Dynamic Comparator

This section introduces a novel class of Dynamic Comparator (DC) relation-decoders, designed to satisfy the criteria **D1-4** outlined in the previous section. These criteria aim to equip relation-decoders with the capability to accurately express and generalize individual and cross-relational semantics within specific orthogonal latent subspaces. In particular, DC adopts inductive biases that encourage disentanglement between families of DC relation-decoders, based on a suitably disentangled representation of the EFs, whilst being encouraged to extrapolate predictions to unobserved regions of $\mathcal{Z}$. Firstly, to achieve consistency, a relation-decoder should be able to model the various sentences that define the semantics of the relations, and sentences can come in different forms. There are in principle two types of sentences that a relation-decoder should be able to model: consistency-individual sentences (Definition 6.1), which describe the semantics of a single relation; and consistency-across sentences (Definition 6.2), which describe the semantics of a relation in terms of other relations. For binary relations, which this chapter will primarily focus on, examples of the different consistency-individual sentences are transitivity, reflexivity/irreflexivity, and symmetry/asymmetry (see Table 6.1 for their definitions). Consistency-across sentences can be broad, and so, staying in theme with Chapter 5, this thesis will focus on consistency-across sentences pertaining to a theory of ordinality, which includes sentences such as $\forall A, B \left(\text{isGreater}(A, B) \leftrightarrow \neg (\text{isLess}(A, B) \vee \text{isEqual}(A, B)) \right)$, describing mutual exclusivity between `isGreater` versus `isLess` and `isEqual`. To be able to model each type of sentence, DC must have a functional form that exhibits the required individual relation behaviour, but these functions must be able to ‘fit’ together in such a way that the sentences over relations are also satisfied.

Definition 6.3 (Dynamic Comparator) *A Dynamic Comparator relation-decoder is a func-*

tion $\phi_r^{DC} : \mathcal{Z} \times \mathcal{Z} \rightarrow (0, 1)$, where $\mathcal{Z} = \mathbb{R}^n$, that is composed of m modes:

$$\phi_r^{DC}(\mathbf{z}_i, \mathbf{z}_j; \mathbf{a}_r, \mathbf{u}_r, \boldsymbol{\eta}_r, \mathbf{b}_r) = \sum_{k=0}^{m-1} a_{r,k} \phi_r^k(\Delta \mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^k) \quad (6.3)$$

where each $\phi_r^k(\Delta \mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^k)$ has the form

$$\phi_r^k(\Delta \mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^k) = f^k(g^k(\Delta \mathbf{z}^b; \mathbf{u}_r); \boldsymbol{\eta}_r^k), \quad (6.4)$$

with the following elements:

- $\phi_r^k : \mathcal{Z} \rightarrow (0, 1)$, $k \in \{0, \dots, m-1\}$,
- attention weights $\mathbf{a}_r = \text{softmax}(\mathbf{a}'_r) \in (0, 1)^m$, with $\mathbf{a}'_r \in \mathbb{R}^m$,
- mask weights $\mathbf{u}_r = \text{softmax}(\mathbf{u}'_r) \in (0, 1)^n$, with $\mathbf{u}'_r \in \mathbb{R}^n$,
- rate parameters $\boldsymbol{\eta}_r = [\boldsymbol{\eta}_r^0; \boldsymbol{\eta}_r^1; \dots; \boldsymbol{\eta}_r^{m-1}] \in \mathbb{R}^l$, where each $\boldsymbol{\eta}_r^k$, $k \in \{0, \dots, m-1\}$, may have a different number of parameters¹,
- bias parameters $\mathbf{b}_r \in \mathbb{R}^n$,
- offset difference $\Delta \mathbf{z}^b = \mathbf{z}_i - \mathbf{z}_j + \mathbf{b}_r$ between the two vectors $\mathbf{z}_i$ and $\mathbf{z}_j$,
- functions $g^k(\Delta \mathbf{z}^b; \mathbf{u}_r)$ responsible for aggregating a directional (via $\mathbf{u}_r$) offset difference (via $\Delta \mathbf{z}^b$) between the two vectors $\mathbf{z}_i$ and $\mathbf{z}_j$, but allows for different functional forms for each mode k ,
- and functions $f^k(\cdot; \boldsymbol{\eta}_r^k)$ responsible for shaping the output of each mode k within $(0, 1)$, but again allowing for different functional forms for each mode k .

.

Note that although DC is defined for binary relations, it can also approximate unary relations by setting $\mathbf{z}_j = \mathbf{0}$ such that $\Delta \mathbf{z}^b = \mathbf{z}_i + \mathbf{b}_r$.

Important features of DC are the use of the relative, but asymmetrical, difference between $\mathbf{z}_i$ and $\mathbf{z}_j$, via $\mathbf{z}_i - \mathbf{z}_j$, and that it is composed of attention-weighted (Bahdanau et al., 2015) *modes*.

¹Note that $[\cdot; \cdot; \dots]$ denotes vector concatenation.

Using a relative difference biases DC towards extrapolation over $\mathcal{Z}$, as this is symmetrical to translations across $\mathcal{Z}$ *i.e.* $\mathbf{z}_i - \mathbf{z}_j = (\mathbf{z}_i + \mathbf{z}') - (\mathbf{z}_j + \mathbf{z}')$, for some displacement vector $\mathbf{z}'$. In contrast, if we were to use absolute values to express relations this could lead to region-specific behaviour. Using attention-weighted modes enables DC to adaptively *select* a mode that can exhibit the consistency-individual characteristics of the relation that it should approximate. The benefit of discretising functionality into modes, is that each can be carefully designed to ensure certain relations can be approximated in accordance with **D1** (*i.e.* satisfying consistency-individual sentences) and that they together can satisfy **D2** (*i.e.* satisfying consistency-across sentences). In the following we first look at options and characteristics of possible modes, and then discuss how these can be combined with the rest of DC's properties to satisfy **D1-4**.

In this thesis, DC is defined with two modes, denoted by $\phi_r^0 = \phi_r^\dagger$ and $\phi_r^1 = \phi_r^\ddagger$, with specific focus on satisfying **D1-2** for a signature of ordinal relations (Equation 5.1)². Here $k = 0, 1$ is replaced by $\dagger, \ddagger$ respectively, in order to indicate the specific choice being made in the following experiments. Hereafter, $\dagger$ and $\ddagger$ are also used in place of $k = 0$ and $k = 1$ for the underlying DC components, *e.g.* $g^0 = g^\dagger$, $g^1 = g^\ddagger$, and so on.

Definition 6.4 (Distance mode $\phi_r^\dagger$) *The distance mode, denoted $\phi_r^\dagger$, is a function of the form:*

$$\phi_r^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^\dagger) = f(g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r); \boldsymbol{\eta}_r^\dagger) \quad (6.5)$$

with the following properties:

- $\phi_r^\dagger : \mathcal{Z} \rightarrow (0, 1)$,
- $g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r)$ is the Euclidean L^2 norm of the masked ($\mathbf{u}_r$) offset difference ($\Delta\mathbf{z}^b$), defined as:

$$g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r) = \|\mathbf{u}_r \odot \Delta\mathbf{z}^b\|_2 \quad (6.6)$$

where $\odot$ is the Hadamard product, and so $g^\dagger : \mathcal{Z} \rightarrow \mathbb{R}^+$,

- $\boldsymbol{\eta}_r^\dagger \in (\mathbb{R}^+)^p$ is a vector of non-negative rate parameters.

²Possibilities for additional modes will be discussed as future work in Section 8.1.

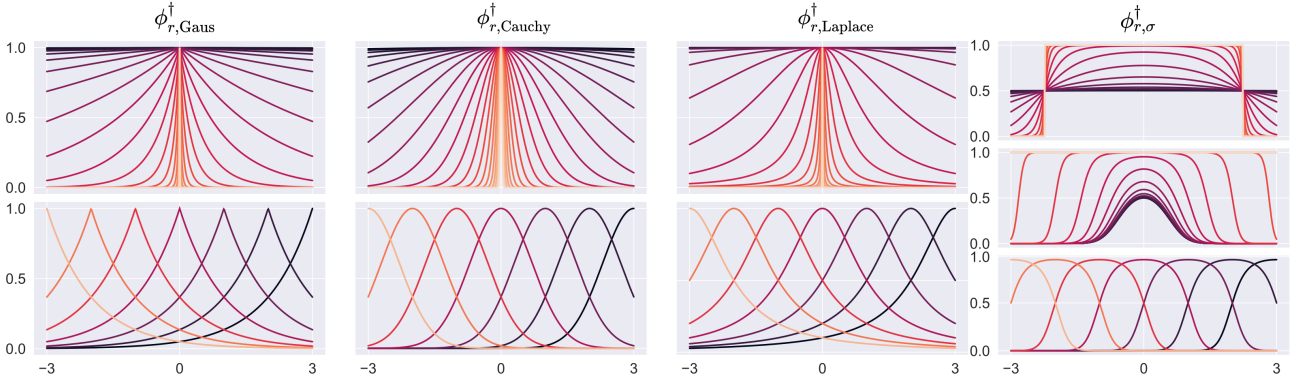


Figure 6.1: An illustration of possible choices for the $\phi^\dagger$ DC mode (given in Equations 6.7, 6.8, 6.9, and 6.10). The y -axis gives the belief for different x -axis $z_{i,k} - z_{j,k}$ values, where it is assumed that $\mathbf{u}_r$ is one-hot, such that the k th dimension is close to 1 and all others close to 0. This has the effect of making $g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)$ reduce to the L^2 norm over the k th dimension of $\Delta \mathbf{z}^b$. Each plot shows how the respective function varies as its learnable parameters are varied, where brighter colours indicate higher values and darker colours indicate lower values.

Example functions that satisfy these properties are:

$$\phi_{r,\text{Laplace}}^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^\dagger) = \exp \left(-\eta_{r,0}^\dagger \cdot |g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)| \right), \quad (6.7)$$

$$\text{where } |g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)| = g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)$$

$$\phi_{r,\text{Gaus}}^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^\dagger) = \exp \left(-\eta_{r,0}^\dagger \cdot g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)^2 \right) \quad (6.8)$$

$$\phi_{r,\text{Cauchy}}^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^\dagger) = \frac{1}{\eta_{r,0}^\dagger} \left(1 + \left(\frac{g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)}{\eta_{r,0}^\dagger} \right)^2 \right)^{-1} \quad (6.9)$$

$$\phi_{r,\sigma}^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^\dagger) = \sigma \left(\eta_{r,0}^\dagger \left(\eta_{r,1}^\dagger - g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)^2 \right) \right), \quad (6.10)$$

where σ is the logistic sigmoid function. The properties of these functions are best seen through illustration. Figure 6.1 plots each function, assuming a one-hot $\mathbf{u}_r$ mask, allowing the functions to be visualised over to a 1-dimensional offset difference between $\mathbf{z}_i$ and $\mathbf{z}_j$. With respect to ϕ_r^{DC} inputs $\mathbf{z}_i$ and $\mathbf{z}_j$, $\phi_r^\dagger$ is a distance-based hat-shaped function which computes the directional distance between two inputs relative to an offset $\mathbf{b}_r^\dagger$. This is achieved through the use of the masked offset difference $\Delta \mathbf{z}^b$ and the L^2 norm function $g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)$, which is masked by $\mathbf{u}_r \odot \Delta \mathbf{z}^b$ to ensure that only the relevant dimensions of $\Delta \mathbf{z}^b$ are considered. Moreover, L^2 is a symmetric function around $\mathbf{0}$, and so $\phi_r^\dagger$ is also symmetric with respect to $\mathbf{z}_i - \mathbf{z}_j$ around $\mathbf{b}_r$. For each function, the rate parameter $\eta_{r,0}^\dagger$ controls the rate of decay of $\phi_r^\dagger$ as $g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)$ increases. All functions that use an exponential produce an output close to 1 or 0 for some input, regardless of parameter setting. In contrast, $\phi_{r,\sigma}^\dagger$ is based on the sigmoid function, and so given certain parameter settings, is not guaranteed to output close to 1 or 0. Instead, for this $\phi_r^\dagger$ option,

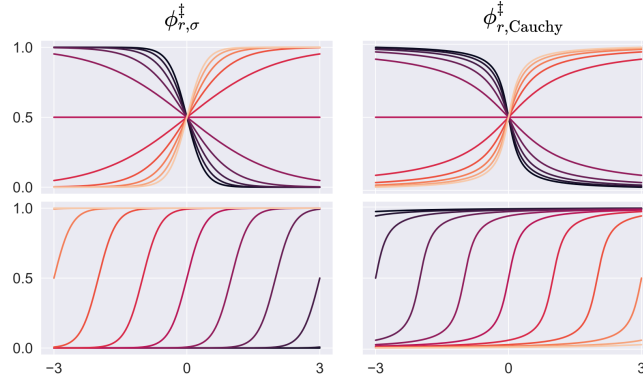


Figure 6.2: An illustration of possible choices for the $\phi^\dagger$ DC mode (given in Equations 6.13 and 6.14). The y -axis gives the belief for different x -axis $z_{i,k} - z_{j,k}$ values, where it is assumed that $\mathbf{u}_r$ is one-hot, such that the k th dimension is close to 1 and all others close to 0. This has the effect of making $g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)$ reduce to the L^2 norm over the k th dimension of $\Delta \mathbf{z}^b$. Each plot shows how the respective function varies as its learnable parameters are varied.

$\eta_{r,0}^\dagger$ and $\eta_{r,1}^\dagger$ together control the shape of $\phi_{r,\sigma}^\dagger$, with $\eta_{r,0}^\dagger$ controlling the rate of decay and $\eta_{r,1}^\dagger$ controlling the width of the hat. We can see that using a sigmoid instead of exponential (0,1) bounding function can produce a more square shape, whereas exponential can produce more triangular/bell-like shapes.

We now look at the second mode, $\phi_r^\dagger$.

Definition 6.5 (Polarity mode $\phi_r^\dagger$) The polarity mode, denoted $\phi_r^\dagger$, is a function of the form:

$$\phi_r^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^\dagger) = f(g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r); \boldsymbol{\eta}_r^\dagger) \quad (6.11)$$

with the following properties:

- $\phi_r^\dagger : \mathcal{Z} \rightarrow (0, 1)$,
- $g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r)$ is the masked ($\mathbf{u}_r$) offset difference ($\Delta \mathbf{z}^b$), defined as:

$$g^\dagger(\Delta \mathbf{z}^b; \mathbf{u}_r) = \mathbf{u}_r^T \Delta \mathbf{z}^b \quad (6.12)$$

and so $g^\dagger : \mathcal{Z} \rightarrow \mathbb{R}$,

- $\boldsymbol{\eta}_r^\dagger \in \mathbb{R}^p$ is a vector of real valued rate parameters.

Suitable $\phi_r^\dagger$ functions include:

$$\phi_{r,\sigma}^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}^\dagger) = \sigma\left(\eta_{r,0}^\dagger \cdot g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r)\right) \quad (6.13)$$

$$\phi_{r,\text{Cauchy}}^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}^\dagger) = \left(\frac{1}{\pi} \arctan\left(\eta_{r,0}^\dagger \cdot g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r)\right) + \frac{1}{2}\right), \quad (6.14)$$

and are illustrated in Figure 6.2. In contrast to $\phi_r^\dagger$, $\phi_r^\dagger$ is an asymmetric function, which computes the directional polarity between two inputs relative to an offset $\mathbf{b}_r^\dagger$. This is achieved through the definition of $g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r)$, which takes the dot product between $\mathbf{u}_r$ and $\Delta\mathbf{z}^b$. In doing so, $g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r)$ is positive when $\mathbf{z}_i - \mathbf{z}_j$ is in the same direction as $\mathbf{u}_r$, and negative when $\mathbf{z}_i - \mathbf{z}_j$ is in the opposite direction to $\mathbf{u}_r$. Moreover, changes in $\mathbf{z}_i - \mathbf{z}_j$ are only relevant when they are in the same direction as $\mathbf{u}_r$, allowing $\phi_r^\dagger$ to evaluate its arguments with respect to a specific direction in $\mathcal{Z}$. The polarity aspect is due to the fact that $g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r) = -g^\dagger(-\Delta\mathbf{z}^b; \mathbf{u}_r)$, and so $\phi_r^\dagger$ is asymmetric around $\mathbf{b}_r$ with respect to $\mathbf{z}_i - \mathbf{z}_j$. This means that the order of which vector is greater in the direction of $\mathbf{u}_r$ is important, and so $\phi_r^\dagger$ is sensitive to the polarity of $\mathbf{z}_i - \mathbf{z}_j$. For each function, the rate parameter $\eta_{r,0}^\dagger$ controls the rate of increase or decay (depending on whether $\eta_{r,0}^\dagger > 0$ or $\eta_{r,0}^\dagger < 0$) of $\phi_r^\dagger$ as $g^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r)$ increases.

Next, we shall see an example of how the above definitions of $\phi_r^\dagger$ and $\phi_r^\dagger$ can be used to satisfy **D1-4** for the signature of ordinal relations (Equation 5.1) with regards to a theory of ordinality (specified in Appendix C). The DC form used in the following analysis is:

$$\phi_r^{\text{DC}}(\mathbf{z}_i, \mathbf{z}_j; \mathbf{a}_r, \mathbf{u}_r, \boldsymbol{\eta}_r, \mathbf{b}_r) = a_{r,0} \cdot \phi_{r,\text{Cauchy}}^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^\dagger) + a_{r,1} \cdot \phi_{r,\sigma}^\dagger(\Delta\mathbf{z}^b; \mathbf{u}_r, \boldsymbol{\eta}_r^\dagger). \quad (6.15)$$

Figure 6.3 depicts how a set of DC relation-decoders are able to approximate the **isGreater** (G), **isLess** (L), **isEqual** (E), **isSuccessor** (S) and **isPredecessor** (P) family of binary ordinal relations, where it is assumed that each corresponding relation-decoder has learned a common one-hot mask on the k th dimension *i.e.* $\mathbf{u}_G = \mathbf{u}_E = \dots = \mathbf{u}_P$, such that $u_{r,k} \approx 1$ and all other mask dimensions are close to zero. This means $\phi_r^{\text{DC}}(\mathbf{z}_i, \mathbf{z}_j; \mathbf{a}_r, \mathbf{u}_r, \boldsymbol{\eta}_r, \mathbf{b}_r)$ will respond only to 1-dimensional differences $z_{i,k} - z_{j,k}$, plotted on the x -axis of Figure 6.3. An important capability of DC is its ability to dynamically *select* (via $\mathbf{a}_r$) an appropriate functional mode, either $\phi_r^\dagger$ or $\phi_r^\dagger$, depending on the type of relation it needs to approximate. As shown by Figure 6.3, with respect to $z_{i,k} - z_{j,k}$, this allows **isEqual** to exhibit its reflexive, symmetric and transitive characteristics, whilst **isGreater** and **isLess** both carry transitivity but are asymmetric and irreflexive, and

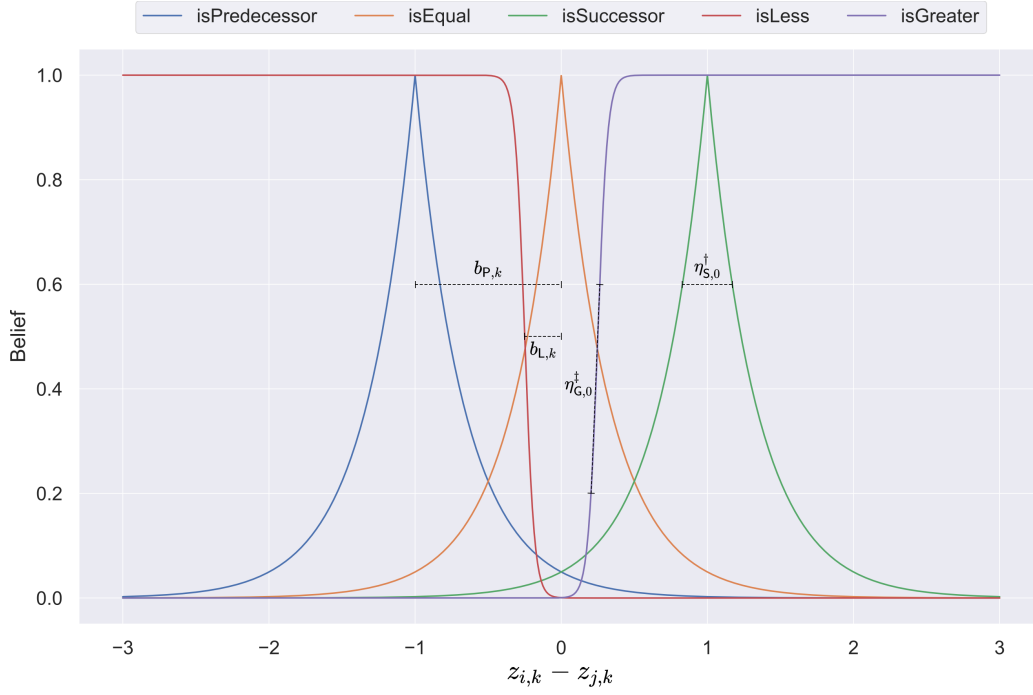


Figure 6.3: Depiction of a set of DC relation-decoders for binary relations *isGreater* (G), *isLess* (L), *isEqual* (E), *isSuccessor* (S) and *isPredecessor* (P). The relation-decoders for *isPredecessor*, *isEqual*, and *isSuccessor* use (via $a_{r,0} \approx 1$) the $\phi_{r,\text{Cauchy}}^\dagger$ mode, whereas *isGreater* and *isLess* use (via $a_{r,1} \approx 1$) the $\phi_{r,\sigma}^\dagger$ mode. As in Figures 6.1 and 6.2, each DC relation-decoder (for each relation) shown here has a one-hot mask, $\mathbf{u}_r$, that is in this example the same across relations, which ensures only the k th dimension of $\mathbf{z}_i$ and $\mathbf{z}_j$ are compared. This lets us assess beliefs over x -axis values of $z_{i,k} - z_{j,k}$. We can see that through suitable parameter selection (some of which annotated on the plots) each relation-decoder is able to learn the appropriate characteristics of its relation, whilst ensuring that they fit together in a manner that ensures consistent beliefs with respect to the mutual across-sentences in the ordinal theory (specified in Appendix C). For instance, we can see that the illustrated relation-decoders for *isGreater* and *isLess* are both asymmetric and irreflexive, but the relation-decoder for *isEqual* is both symmetric and reflexive. Additionally, when the belief of the relation-decoder for *isSuccessor* (*isPredecessor*) is high, the belief of the relation-decoder for *isGreater* (*isLess*) is also high, which is required by their associated implication rules.

isSuccessor and *isPredecessor* are both asymmetric and irreflexive³. We can see that for this signature of relations, DC is able to satisfy the consistency-individual sentences **D1**. We also observe adherence with respect to the consistency-across sentences (**D2**). For instance, our theory of ordinality (Appendix C) states that *isLess*, *isEqual*, and *isGreater* are mutually exclusive and we indeed see that when the belief for either corresponding relation-decoder is high, all others are low. Moreover, when the belief for *isSuccessor* (*isPredecessor*) is high, the belief for *isGreater* (*isLess*) is also high, which is required by their associated implication rules, and when the belief for either *isLess*, *isGreater*, *isEqual* is low, which is required by the implication rules between *isEqual* and the other relations, *e.g.* $\forall A, B \text{ (isSuccessor}(A, B) \rightarrow \text{isGreater}(A, B))$ and $\forall A, B \text{ (isPredecessor}(A, B) \rightarrow \text{isLess}(A, B))$. Additionally, as stated above, the fact that DC

³The individual characteristics of each relation are formally provided in Appendix C as defined by a theory of ordinality.

employs subtraction between input arguments means it can extrapolate its analysis through translations in $\mathcal{Z}$. This means the consistent beliefs illustrated in Figure 6.3 will be maintained across $\mathcal{Z}$, leading to generalised consistency **(D3)**. Lastly, the role of each $\mathbf{u}_r$ is particularly important. As stated, in the illustrated example it is assumed that each relation-decoder has learned a common one-hot mask on the k th dimension. This ensures that the collection of DC relation-decoders select a common 1-dimensional orthogonal subspace of $\mathcal{Z}$, in accordance with the disentangled family of relations that they aim to approximate **(D4)**. In the next section we will see that this agreement in terms of $\mathbf{u}_r$ can indeed be practically achieved through data-driven learning.

6.1.3 Learning to approximate disentangled families of relations

If a set of relation-decoders are able to approximate disentangled families of relations for a given EF, then the outputs of all corresponding relation-decoders should be invariant to changes in all other EFs. To evaluate whether this can be achieved with DC, we will look at a representation learning study performed on the **dSprites** dataset, using the same architecture from Section 4.2, but this time employing DC relation-decoders for the weakly-supervised relation-decoder auxiliary layer. As we previously saw in Table 3.1, the **dSprites** dataset was generated using six EFs: colour, shape, scale, orientation x-position, and y-position. For each of these we can define a disentangled family of relations and associated set of relation-decoders. However in this study colour and orientation families are omitted. For colour this is due to there being no variation - all shapes are white⁴. The reason for omitting orientation is more complex and is due to the difficulties with orientational symmetry; tackling periodic properties and associated relations is out of scope for this thesis, but is a natural direction to take towards expanding DC mode capabilities as discussed in Section 8.1. This leaves the study with four EFs, each with their own disentangled family of relations. The aim of this study is to show that, for each EF, we can learn associated families of DC relation-decoders that together share the same independent domain. Since DC includes an initial masking layer, we can compare domains of each learned DC relation-decoder by visualising each $\mathbf{u}_r$ mask. As in Chapter 4, a 10-dimensional latent space ($\mathcal{Z} = \mathbb{R}^{10}$) is used.

⁴The **Abstract dSprites** dataset variant could be used, in which new colours are introduced, but for the purposes of this study standard **dSprites** is sufficient.

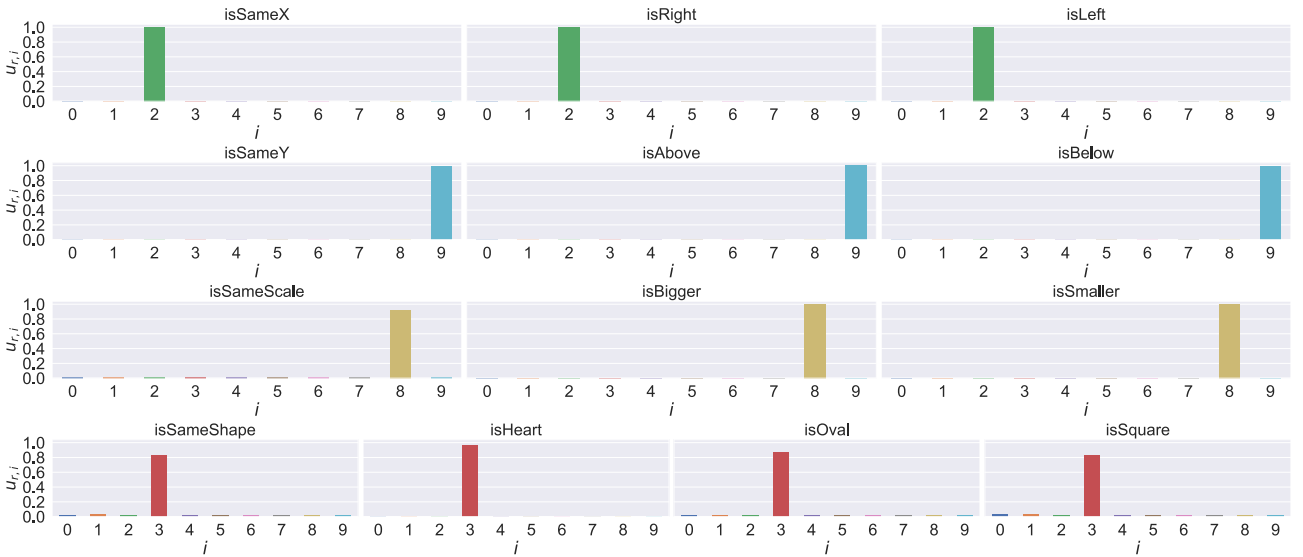


Figure 6.4: Masks for a set of DC relation-decoders trained on the **dSprites** dataset alongside a VAE, with $\mathcal{Z} = \mathbb{R}^{10}$. Each row provides a set of relation-decoders associated with a disentangled family of relations with respect to a particular EF. We can see that each set of relation-decoders learns a similar one-hot mask, indicating that they each take the same $\mathcal{Z}$ subspace as their domain. In general **isSameShape** is the least one-hot, likely due to the inherently visual nature of the shape EF.

Figure 6.4 shows that each DC relation-decoder is able to learn a one-hot mask, and that collections of relation-decoders either completely agree or disagree. Results show that for the shape EF, the mask is slightly less one-hot, which once again (*c.f.* Chapter 4) suggests that inherently visual concepts such as shape similarity can be confusing for relational representation learning. Next, to check whether the EFs that each relation-decoder is sensitive to (via its mask) correspond with the underlying property that their relation-decoding should depend on, Figure 6.5 shows how the activations of each $\mathcal{Z}$ dimension of a different set of **dSprites** trained DC relation-decoders, with respect to changes in the underlying EF values. The plots are produced by setting a specific EF at a given value, gathering all data in the dataset that match it, and then taking a random sample such that the other EFs vary across their available range. The plots clearly show that for each EF we can visually identify a unique dedicated $\mathcal{Z}$ dimension, suggesting that DC assigns independent subspaces to each disentangled family of relations, and that these correspond with the correct underlying EF.

The results of Figure 6.4 and Figure 6.5 together present a strong indication that DC is able to learn to approximate disentangled families of relations, in terms of learning to select orthogonal subspace domains associated with each **dSprites** EF. In the remainder of this chapter we will see a more thorough investigation into relative transferability and ϵ -consistency/ ϵ -coherence behaviour of DC, versus NTN⁺ and a number of other popular relation-decoder implementations.

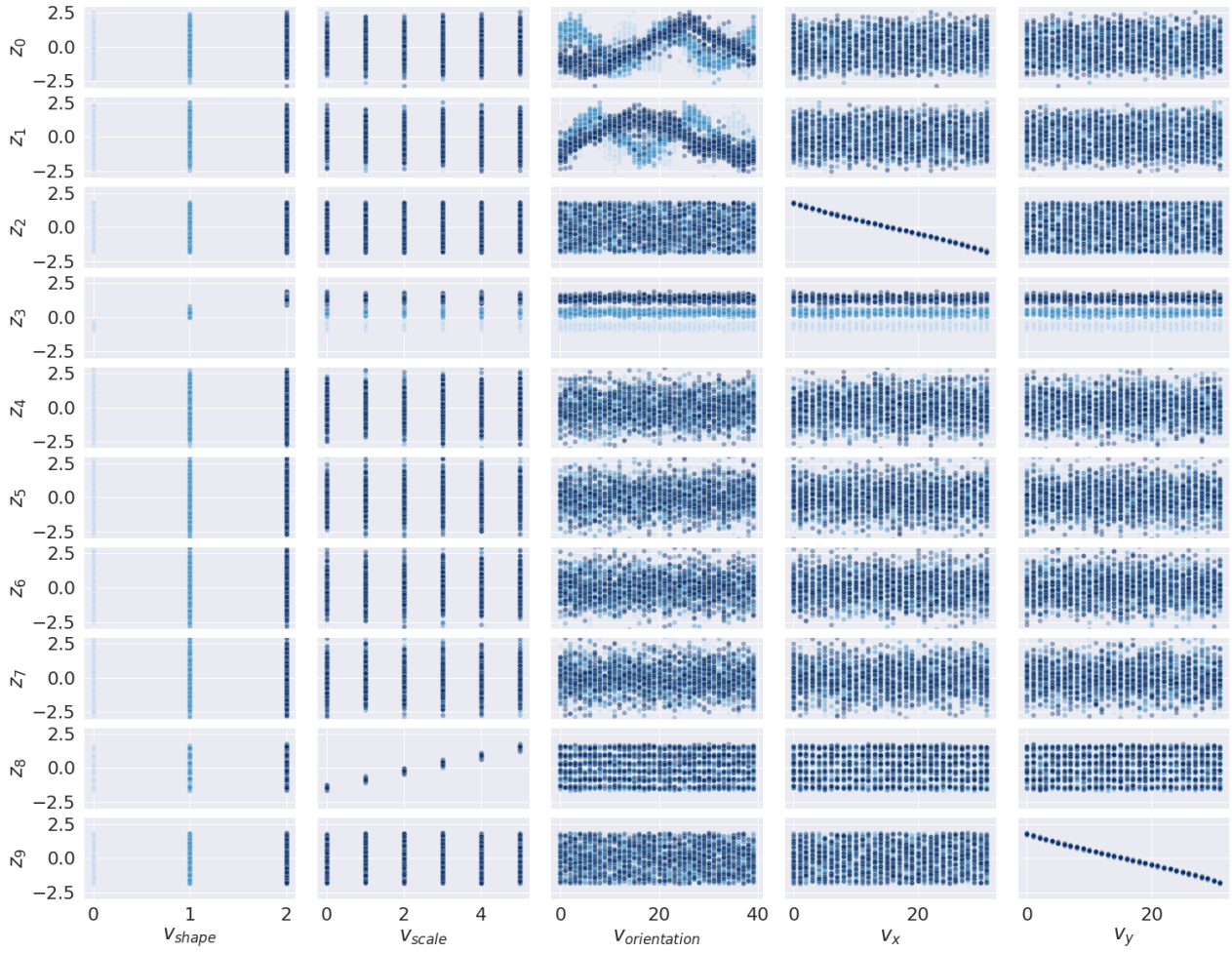


Figure 6.5: Plots showing the activations of each $Z = \mathbb{R}^{10}$ dimension, as the underlying ground-truth EF settings are changed, when training a VAE with DC relation-decoders in the relational auxiliary layer. Each row gives the activations for a specific Z dimension, and each row presents a specific EF. The samples are produced by gathering all data where the EF matches the chosen value and then randomly sampling from all other EF settings. For each EF we can visually identify a unique latent dimension that is sensitive to its changes, demonstrating disentanglement of the dSprites EFs in Z .

6.2 Partial Relation Transfer

Chapter 3 motivated FOL state abstractions as a selectively reusable knowledge that could be suitable for a lifelong learner. Coming back to this discussion, this section seeks to evaluate the relationship between consistency/coherence and FOL-like transfer. At its core, coherence is a transfer oriented property, as it attempts to provide a DL bridge from the general semantics of FOL relation symbols to relation-like DL relation-decoders. In FOL if we are given a signature σ then an associated theory $\mathcal{T}$ clearly determines which structures are or are not models. This does not depend on the domains of the structures per se, but rather whether the interpretations of σ over each given domain are consistent with $\mathcal{T}$. The semantics of σ as defined by $\mathcal{T}$

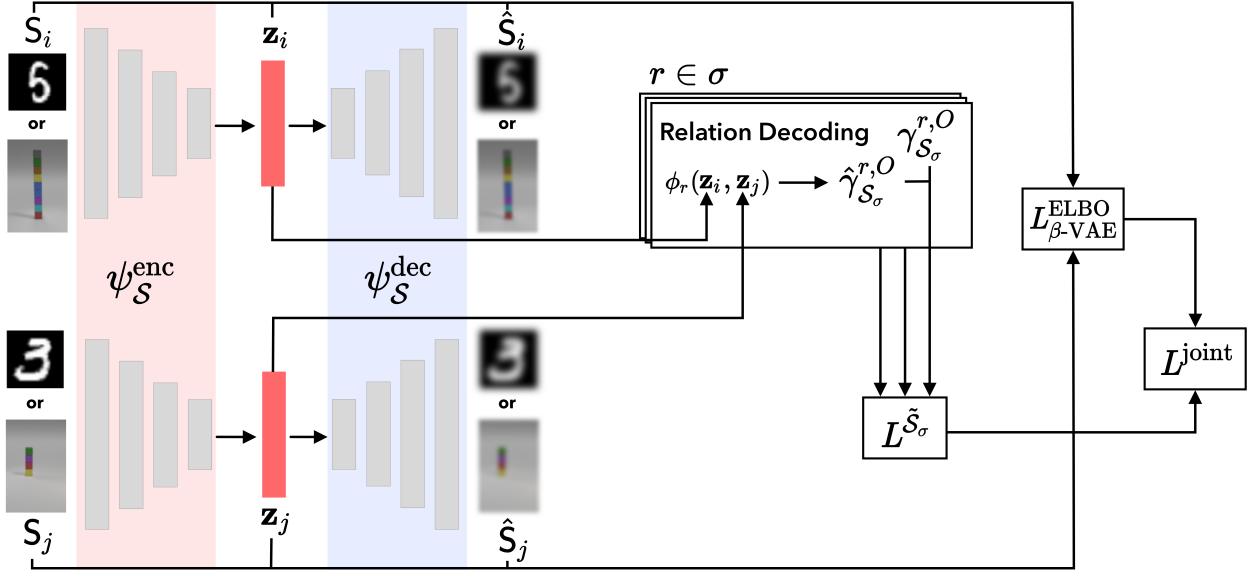


Figure 6.6: Depiction of the architecture used for PRT tasks. In the experiments $s_{i/j}$ are either MNIST (domain $\mathcal{X}_s$) or BlockStacks (domain $\mathcal{X}_t$) images. Relational learning is performed on the source ($\mathcal{S} = \mathcal{X}_s$) MNIST domain (to learn e.g. that digit 5 is greater than 3). Moving to the target ($\mathcal{S} = \mathcal{X}_t$) domain (stacks of blocks) involves training a new image autoencoder together with a subset of the relation-decoders from MNIST with fixed parameters. The remaining relations are held-out to evaluate zero-shot TL performance. $\gamma_{\mathcal{S}_\sigma}^{r,O}$ provides the ground truth for the given structure, which ϕ_r predicts as $\hat{\gamma}_{\mathcal{S}_\sigma}^{r,O}$. O is used to abbreviate $(\psi_{\mathcal{S}}^{\text{enc}}(\mathbf{S}_i), \psi_{\mathcal{S}}^{\text{enc}}(\mathbf{S}_j))$.

are in this sense *domain-independent*. In contrast, when learning a set of relation-decoders by maximising structure support with respect to a specific model of $\mathcal{T}$, even if the resulting ϵ -consistency tends towards $\epsilon = 0$, it is not clear if the semantics of $\mathcal{T}$ have been learned in a general sense or if they are limited to the specific structure (or, more specifically, domain) under observation. The degree to which the consistency with respect to $\mathcal{T}$ can be transferred over to other structures/domains is what ϵ -coherence measures. From a practical sense, it would be useful if we could identify when a trained set of ϕ^σ relation-decoders is likely to induce coherence between soft-substructures that together share ϕ^σ . Towards this end, this section outlines an experimental setup called Partial Relation Transfer (PRT), which is devised to investigate under which practical conditions maximising structure support for a source model of $\mathcal{T}$ leads to coherence between soft-substructures.

Experimental setup

Let $\mathcal{X}_\sigma^s$ and $\mathcal{X}_\sigma^t$ be two models of a theory $\mathcal{T}$ that share the same signature σ , but have disjoint domains $\mathcal{X}_s$ and $\mathcal{X}_t$, *i.e.* $\mathcal{X}_s \cap \mathcal{X}_t = \emptyset$. Additionally, let $\tilde{\mathcal{X}}_\sigma^s = (\psi_{\mathcal{X}_s}^{\text{enc}}(\mathcal{X}_s), \phi^\sigma)$ and $\tilde{\mathcal{X}}_\sigma^t = (\psi_{\mathcal{X}_t}^{\text{enc}}(\mathcal{X}_t), \phi^\sigma)$ be soft-structures, with $\phi^\sigma = \{ \phi_r \mid r \in \sigma \}$. Note that here we return back to specifying ψ^{enc} with its superscript, as the experimental architecture will once again employ

VAEs and we must specify the input encoder versus reconstruction decoder in the notation. PRT is a TL experiment in which the source objective is to maximise structure support $p(\mathcal{X}_\sigma^s | \tilde{\mathcal{X}}_\sigma^s)$ and the target objective is to maximise structure support $p(\mathcal{X}_\sigma^t | \tilde{\mathcal{X}}_\sigma^t)$. Given the shared signature and ϕ^σ relation-decoders, it is desirable that maximising the source objective leads to maximisation of the target objective without additional target training. However, in a practical setting it is unlikely that naïvely setting $\psi_{\mathcal{X}_t}^{\text{enc}} = \psi_{\mathcal{X}_s}^{\text{enc}}$ will be effective, due to $\mathcal{X}_s$ and $\mathcal{X}_t$ being disjoint and potentially having different data distributions. The experimental design involves first learning ϕ^σ on source domain $\mathcal{X}_s$ (source phase), together with its domain-specific autoencoder, $\psi_{\mathcal{X}_s}^{\text{enc}} / \psi_{\mathcal{X}_s}^{\text{dec}}$. Then, a new domain-specific autoencoder, $\psi_{\mathcal{X}_t}^{\text{enc}} / \psi_{\mathcal{X}_t}^{\text{dec}}$, is trained on the target domain $\mathcal{X}_t$ (target phase), alongside a selection of the now learned ϕ^σ relation-decoders but with fixed-parameters, denoted by ϕ^{guide} . The guide relation-decoders, ϕ^{guide} , are expected to help guide the training of $\psi_{\mathcal{X}_t}^{\text{enc}}$ such that its encodings lead to consistent beliefs under ϕ^σ . Held-out relation-decoders, $\phi^{\text{held-out}}$, are then evaluated with respect to $\mathcal{X}_t$, in which transfer will depend on the regularities that are shared between ϕ^{guide} and $\phi^{\text{held-out}}$. In FOL, these regularities are the sentences that constrain the relations; in this way, PRT is a TL experiment that seeks to evaluate the zero-shot transferability from ϕ^{guide} to $\phi^{\text{held-out}}$ relation-decoders. The core hypothesis is that good PRT performance (*i.e.* good zero-shot transfer) likely relies on coherence between soft-substructures, since coherence is a measure of the degree to which the semantics of $\mathcal{T}$ are learned in a general sense, and from a FOL standpoint it is exactly this generality that is necessary to facilitate transfer. The overall experimental setup is most similar to source-free domain adaptation (Yu et al., 2023), where an architecture is trained in a source domain and then adapted to a target domain using unlabelled data. However, these methods do not investigate the zero-shot inter-architecture behaviour, as is done here, and are instead primarily concerned with the achievable target domain prediction accuracy.

In the following experiments, both structures have image datasets as their domains: $\mathcal{X}_s$ is the MNIST handwritten digits dataset and $\mathcal{X}_t$ is a novel **BlockStacks** dataset, consisting of a single stack of multi-coloured cubes of differing heights, each containing a unique randomly-positioned red cube (see Appendix A examples and additional dataset details). As stated earlier, the architecture that will be employed is the same as that given in Section 4.2 and so a VAE is used in conjunction with the base soft-structure learning architecture; ψ^{enc} denotes the common encoder (shared between the relation-decoders and the VAE) and ψ^{dec} denotes the VAE decoder. An illustration of the overall architecture is given in Figure 6.6. The shared

signature includes ordinal relations $\sigma = \{\text{isGreater (G)}, \text{isEqual (E)}, \text{isLess (L)}, \text{isSuccessor (S)}, \text{isPredecessor (P)}\}$, and is applied to digit ordering in **MNIST** and to red cube position ordering in **BlockStacks**. These relations are together constrained by a theory of ordinality containing rules such as $\forall A, B (\text{isGreater}(A, B) \rightarrow \neg \text{isLess}(A, B))$ ⁵. Intuitively, for this signature and theory, establishing the ordering over a set using the **isSuccessor** relation-decoder should lead to consistent beliefs over the for the remaining ordinal relation-decoders, providing the full set of relation-decoders is consistent with the underlying ordinal theory. This fact is more clearly discussed in Appendix C, where the full ordinal theory is specified. It is therefore only necessary to employ the **isSuccessor** relation-decoder for the target domain ϕ^{guide} , to guide the learning of $\psi_{\mathcal{X}_t}^{\text{enc}}$. If coherence, as defined in Section 5.1.2, is carried across domains, we would expect that accuracy of the **isSuccessor** relation-decoder in the target domain to be carried over to the remaining relation-decoders.

Training

Staying in theme with Chapter 4, PRT uses the same architecture as described in Section 4.2, in which a backbone VAE is trained alongside an additional set of relation-decoder heads, and uses the same objective function (Equation 4.2) for training in both the source phase and target phase. Inline with Chapter 4’s experiments, weak relational supervision is provided through partial relation labelling of dataset tuples (see Section 4.3 for further details). However, equipped with the theoretical foundations laid down in Chapter 5, we can re-formulate Chapter 4’s experiments in terms of structure support maximisation. In effect, being given relational labels for a set of relations $r \in \sigma$ which determine which relations hold between instances of some dataset $\mathcal{S} = \{s_1, s_2, \dots, s_{|\mathcal{S}|}\}$ is equivalent to being provided the interpretation $I_{\mathcal{S}_\sigma}$ of a structure $\mathcal{S}_\sigma = (\mathcal{S}, I_{\mathcal{S}_\sigma})$. This is to say that the partial labelling can be viewed as a partial sampling from the interpretation labels $\left\{ \gamma_{\mathcal{S}_\sigma}^{r,O} \mid r \in \sigma, O \in \mathcal{S}^{\text{ar}(r)} \right\}$ of a ground-truth interpretation $I_{\mathcal{S}_\sigma}$ and the corresponding structure $\mathcal{S}_\sigma$ serves as a ground-truth structure that the learner is trained to support. Supervised training over the relation labels using a negative log-likelihood objective,

⁵In Section 5.4.1 we saw how specific EFs can be described by their own disentangled theories, for instance, but here we are describing two EFs using the same theory. This is possible because, in this case, the two EFs are both discrete ordinals and their semantics can be described using a single ordinal theory. We are only able to reuse the same relation-decoders here because we only focus on one ordinal in each domain.

as we did in Chapter 4, can then be re-formulated as a structure support objective,

$$\mathcal{L}^{\tilde{\mathcal{S}}_\sigma}(\mathcal{S}_\sigma; \psi_{\mathcal{S}}^{\text{enc}}, \phi^\sigma) = \arg \min_{\psi_{\mathcal{S}}^{\text{enc}}, \phi^\sigma} \left(-\log p(\mathcal{S}_\sigma | \tilde{\mathcal{S}}_\sigma) \right), \quad (6.16)$$

where $\phi^\sigma = \{ \phi_r \mid r \in \sigma \}$. Together with the VAE loss term, we arrive at the re-formulated final objective,

$$\mathcal{L}^{\text{joint}}(\mathcal{S}_\sigma; \psi_{\mathcal{S}}^{\text{enc}}, \psi_{\mathcal{S}}^{\text{dec}}, \phi^\sigma) = \mathcal{L}^{\text{ELBO}}(\mathcal{S}; \psi_{\mathcal{S}}^{\text{enc}}, \psi_{\mathcal{S}}^{\text{dec}}) + \lambda \mathcal{L}^{\tilde{\mathcal{S}}_\sigma}(\mathcal{S}_\sigma; \psi_{\mathcal{S}}^{\text{enc}}, \phi^\sigma). \quad (6.17)$$

Therefore, we can view both Chapter 4’s experiments and PRT as a structure support experiments with an additional inductive bias towards disentangled representations. Whereas Chapter 4 sought to understand how weak relational supervision impacts disentanglement and downstream transfer, PRT seeks to understand how disentanglement and an appropriate relation-decoder implementation can lead to coherent concept learning.

Neural model components and hyperparameters

Together with NTN⁺ and DC, PRT relation-decoder models evaluated here are: Translational embedding relation-decoder with relation-specific parameters (TransR) (Lin et al., 2015), Holographic Embeddings relation-decoder (HolE) (Nickel et al., 2016b) and a basic feedforward neural network denoted by Basic neural network relation-decoder (NN). A flexible fitting procedure for DC was used, in which training of the DC parameters for the ϕ_5 relation-decoder is enabled in the target domain, but with the additional loss term $\|\boldsymbol{\rho}^* - \boldsymbol{\rho}\|$ between pretrained $\boldsymbol{\rho}^*$ and untrained parameters $\boldsymbol{\rho}$, respectively. In all cases the final parameter values were evaluated in the target domain and found to be approximately equivalent to the $\boldsymbol{\rho}^*$. The reasoning behind this fitting procedure is discussed further in Appendix A.

For the VAE all experiments employ a β -VAE (Higgins et al., 2017a), as this produces a single ‘disentanglement pressure’ hyperparameter to tune⁶. In the source domain, β values explored are $\beta \in \{1, 4, 8, 12\}$ and λ is set to 10^3 . In the target domain losses are normalised and β is set to 10^{-4} and λ to 10^{-2} . These values were selected as they produced good image reconstructions whilst still being able to optimise structure support. In all experiments $\mathcal{Z}$ is set

⁶Additionally In (Locatello et al., 2019) it was found that with appropriate choice of hyperparameter, most VAE implementations can obtain similar disentanglement results.

to $\mathbb{R}^{10}$ (10-dimensional vector space). Further details for all models, including training regimen, parameterization and implementation are provided in Appendix B.

As stated at the beginning of this section, the goal behind PRT is to understand when good ϵ -coherence is likely to be achieved between soft-substructures in a practical setting. With this understanding, it may be possible to identify properties on soft-substructures that indicate their ϵ -coherence with respect to possible downstream soft-substructures. In which case, such properties can then be used to guide the design of soft-substructures (in terms of encoder and relation-decoder implementation) and act as a source diagnosis for soft-substructure quality in terms of coherence. One such property that is of course necessary to measure is the ϵ -consistency of the source and target soft-substructures, but direct computation of Equation 5.9 can quickly become intractable as the domain grows, due to exponential growth in the number of structure support product terms, and so we require a more practical alternative. The following section provides such a measure.

6.3 Measuring consistency and coherence in practice

Measuring consistency and coherence through direct calculation of Equation 5.9 can be too computationally expensive for larger domains and so an efficient approach to measuring the ϵ -consistency/coherence of soft-substructures is required. This section introduces a proxy measure for a soft-substructure's ϵ -consistency and ϵ -coherence with respect to a given theory when access to every logical model is not available or is computationally intractable. The intuition for this measure is that soft-substructure consistency can be estimated on a case-by-case basis, achieved by randomly sampling groundings of the theory's sentences and evaluating soft-substructure support for consistent versus inconsistent groundings. This section presents the proxy measure with its derivation outline. The full derivation can be found in Appendix D.

Suppose that there is a fixed domain $\mathcal{S}$ and a theory $\mathcal{T}$ defined using a signature σ . Let $k \in \{1, \dots, K_0\}$ denote the index associated with each unique ground instance of the sentences in $\mathcal{T}$. Lastly, take $B_{\mathcal{T}}$ to be a Boolean random variable where for the k -th grounding of $\mathcal{T}$, the probability of the theory being satisfied under soft-substructure $\tilde{\mathcal{S}}_{\sigma}$ is expressed as $p(B_{\mathcal{T}} = 1 | \tilde{\mathcal{S}}_{\sigma}, k)$, and the probability of non-satisfaction is given by $p(B_{\mathcal{T}} = 0 | \tilde{\mathcal{S}}_{\sigma}, k)$. Similarly $p(B_{\mathcal{T}} = 1 | \mathcal{S}_{\sigma}, k)$ and $p(B_{\mathcal{T}} = 0 | \mathcal{S}_{\sigma}, k)$ are the probabilities of satisfaction and non-satisfaction

for a structure $\mathcal{S}_\sigma$; since structures have deterministic interpretations, these probabilities will be deterministic (either 0 or 1). In particular, for a model $\mathcal{S}_\sigma \in \mathcal{M}_\mathcal{S}^\mathcal{T}$ of the theory $\mathcal{T}$ comprised of sentences, any grounding k of $\mathcal{T}$ is always satisfied (by definition), and thus $\forall k, p(B_\mathcal{T} = 1 | \mathcal{S}_\sigma, k) = 1$. When $\tilde{\mathcal{S}}_\sigma$ is consistent with $\mathcal{T}$, we should also find that $\forall k, p(B_\mathcal{T} = 1 | \tilde{\mathcal{S}}_\sigma, k) \approx 1$. Hence, we can define a consistency loss function as the expectation over a randomly-chosen grounding k of the binary cross-entropy between $p(B_\mathcal{T} | \mathcal{S}_\sigma, k)$ and $p(B_\mathcal{T} | \tilde{\mathcal{S}}_\sigma, k)$ for any $\mathcal{S}_\sigma \in \mathcal{M}_\mathcal{S}^\mathcal{T}$, where larger values of the loss indicate worse consistency. This in turn simplifies to the expected negative log-likelihood of satisfying a random grounding of $\mathcal{T}$, as follows:

$$\mathcal{L}^\epsilon(\mathcal{T}, \tilde{\mathcal{S}}_\sigma) = \mathbb{E}_{k \sim p(k)} [-\ln p(B_\mathcal{T} = 1 | \tilde{\mathcal{S}}_\sigma, k)]. \quad (6.18)$$

where $p(k) = \frac{1}{K_0}$ is the uniform distribution over the set of unique groundings. A measure based on this loss is required to enable a practical approximation of ϵ -consistency. More precisely, we can define

$$\bar{\Gamma}_{\mathcal{M}_\mathcal{S}^\mathcal{T}}^{\tilde{\mathcal{S}}_\sigma} = \exp \left(-\mathcal{L}^\epsilon(\mathcal{T}, \tilde{\mathcal{S}}_\sigma) \right) \quad (6.19)$$

as a proxy measure of $\Gamma_{\mathcal{T}}^{\tilde{\mathcal{S}}_\sigma}$, and say that soft-substructure $\tilde{\mathcal{S}}_\sigma$ is $\bar{\epsilon}$ -consistent with $\mathcal{T}$ if

$$\ln \frac{1}{1 - \bar{\epsilon}} \geq \mathcal{L}^\epsilon(\mathcal{T}, \tilde{\mathcal{S}}_\sigma) \quad (6.20)$$

where $\bar{\epsilon} \geq 1 - \bar{\Gamma}_{\mathcal{T}}^{\tilde{\mathcal{S}}_\sigma}$. Due to the relationship between $\bar{\epsilon}$ and $\mathcal{L}^\epsilon(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$, we can take the $\bar{\epsilon}$ -coherence of a set of soft-substructures to be the smallest value of $\mathcal{L}^\epsilon(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$ evaluated for each soft-substructure in the set. The full derivation of this proxy measure is given in Appendix D.

Although our treatment of consistency has thus far focused on a particular theory $\mathcal{T}$, notice that a subset of the sentences of $\mathcal{T}$ form a partial theory, which is itself a theory. This means that ϵ -consistency can be evaluated for a partial (even single sentence) theory, allowing us to examine consistency losses for any partial specifications of a theory that can be of interest for given domain. In other words, consistency can be interrogated at different levels of granularity, from the full theory to any of its partial theories. In the following PRT experiment, the proposed consistency loss will be evaluated against two partial specifications within the theory of ordinality. As introduced in Section 6.1.1, these are the Consistency-Across (Con-A) and Consistency-Individual (Con-I) partial theories⁷. Con-A includes the sentences that

⁷The full theory specification and associated truth-tables for each set of sentences are given in Appendix C.

determine inter-relation semantics. For ordinality this includes, for example, the sentence $\forall A, B \text{ (isGreater}(A, B) \rightarrow \neg \text{isEqual}(A, B) \wedge \neg \text{isLess}(A, B))$, stating that if A is greater than B then A must not be equal to or less than B . Con-I includes the sentences that are about a single relation, describing any property of an individual relation. Each relation may satisfy a number of properties, for instance $\forall A, B, C \text{ (isGreater}(A, B) \wedge \text{isGreater}(B, C) \rightarrow \text{isGreater}(A, C))$ represents transitivity of the **isGreater** relation. Transitivity is true for **isGreater**, but is false for other relations investigated in this paper, *e.g.* **isSuccessor**. Consistency losses will be evaluated for each of the characteristic Con-I sentences (Table 6.1) of each relation in the ordinality relations signature: transitivity (Con-I-T), asymmetry (Con-I-A) and reflexivity (Con-I-R). Furthermore, an important feature of Equation 6.18 is that it does not depend on a provided structure - its derivation relies instead on the fact for a logically consistent structure, any grounding of a theory will be satisfied, and the mirror of this is that a soft-structure should only support consistent groundings of the theory’s sentences. The lack of an explicit structure in Equation 6.18 means that the loss can be calculated for arbitrary relation-decoder inputs, providing an analysis that is detached from specific $\mathcal{S}$ encodings. This will provide useful in the following experimental analysis, as we will be able to explore consistency losses for multiple regions of $\mathcal{Z}$ with respect to a given $\mathcal{S}$ and its encoding $\psi_{\mathcal{S}}(\mathcal{S})$. In the following experiments, only universally quantified sentences are considered, where quantification is at the top level of the sentence. Generalisation of the method to formulae with nested quantifiers and existential quantifiers will be discussed in Section 8.1.

Equipped with a practical consistency loss and $\bar{\epsilon}$ -consistency/coherence proxy, we can estimate the consistency and coherence of soft-substructures with respect to a universally quantified theory of ordinality. In the following results, such a comparison will be presented for different soft-substructures, each implemented using a variety of relation-decoder models and trained under varied disentanglement inductive bias.

6.4 Results

As stated previously, the goal of PRT is to act as a test-bed experiment for evaluating the coherence properties of soft-structures, such that: we can understand under which source training conditions the learned relation-decoders may obtain FOL-like transferability between

Table 6.2: Spearman rank coefficients between consistency loss and PRT accuracy. Coefficients are calculated for each consistency loss reported in the main text, across all models, β settings and regions of latent space. Results show a strong inverse rank correlation between interpolation Con-A/Con-I-T/Con-I-A and PRT performance.

$\mathcal{Z}$ region	Spearman Rank Coefficient			
	Con-A	Con-I-T	Con-I-A	Con-I-R
Data-Embeddings	0.25	-0.45	-0.47	0.23
Interpolation	-0.77	-0.75	-0.71	-0.42
Extrapolation	-0.60	-0.66	-0.61	-0.49

domains; and we can diagnose a set of relation-decoders to determine their expected coherence quality. The results that are recruited for this purpose include: a correlation analysis between consistency loss, using the proposed practical loss outlined in Section 6.3, and zero-shot PRT performance in terms of $\phi^{\text{held-out}}$ target domain relation-decoder accuracy; source and target domain relation-decoder accuracies for each relation-decoder in ϕ^{guide} and $\phi^{\text{held-out}}$, including a breakdown of aggregated accuracy for each choice of source β setting; and a consistency loss breakdown for each relation-decoder with respect to both target and source domains.

In the previous section we saw that the independence of Equation 6.18 with respect to any specific structure means that consistency losses can be calculated for arbitrary regions of relation-decoder input space $\mathcal{Z}$, not just the domain encodings of a reference structure. This fact will be exploited in the following analysis, where consistency losses are evaluated for three regions of $\mathcal{Z}$: *data-embeddings* region, where all inputs are encodings of a domain’s test data; *interpolation* region, where samples are from a Gaussian distribution whose parameters are the empirical mean and variance of the domain’s data-embeddings; and *extrapolation* region, where samples are from regions strictly outside the smallest, axis-aligned, hyper-rectangle that encloses all domain data-embeddings.

Spearman rank correlation

To firstly investigate the broad trends that run across all β values and relation-decoder implementations, a Spearman rank correlation analysis was performed between each consistency loss (Con-A, Con-I-T, Con-I-A, Con-I-R) and zero-shot PRT performance. Results are tabulated in Table 6.2, where coefficients are reported for each combination of consistency loss and region of latent space (data-embedding, interpolation and extrapolation). Note that coefficients are not

separated between β and relation-decoder choice, but instead evaluate the general correlation between consistency loss and PRT performance. Overall, each coefficient aims to characterize the PRT performance change when a soft-substructure is more or less consistent with a given partial theory, over a particular region of latent space. Unlike the Pearson correlation, the Spearman rank correlation can describe monotonic *curvilinear* relationships between variables (Schober et al., 2018). A Spearman rank coefficient varies between $[-1, 1]$, where a coefficient ± 1 indicates a perfect rank correlation. If the coefficient is negative (positive) this means a reduction (increase) in one variable corresponds with an increase in the other. Separate coefficients are produced for each theory fragment’s corresponding consistency loss: Con-A and Con-I, further divided into Con-I-T (transitivity), Con-I-A (asymmetry) and Con-I-R (reflexivity); and regions of latent space: data-embedding, interpolation and extrapolation. Lower consistency losses are expected to produce higher PRT performance as indicated by a negative Spearman rank coefficient. The coefficients show that the consistency losses of data-embeddings in the source domain are weakly rank correlated with PRT. The consistency losses in the case of interpolation are, in most cases, strongly rank correlated with PRT. The consistency losses in the case of extrapolation lie in between and are generally moderately rank correlated with PRT. This supports the thesis that generalised consistency can facilitate reliable PRT. Furthermore, consistency of certain partial theories may matter more. Here, Con-A, Con-I-T and Con-I-A on interpolation are the most relevant partial theories for PRT performance. As we shall see, DC outperforms all other models on these losses and this result is indeed mirrored by its PRT performance.

PRT performance profiles

Figure 6.7-top provides relation-decoder prediction accuracies in both the source (MNIST, left) and target (BlockStacks, right) domains. Values shown are for the optimal source β setting, with respect to zero-shot PRT performance, for each relation-decoder. To obtain accuracy values, ϕ_r beliefs of 0.5 or above are taken to signify *true*, and otherwise *false*. In the source phase ϕ^{guide} includes all relation-decoders, whereas in the target phase ϕ^{guide} includes only the *isSuccessor* (S) relation-decoder and the rest are held-out in $\phi^{\text{held-out}}$. Key observations are that DC produces excellent PRT performance, whilst NN, NTN⁺ and HoIE all see some degradation from their source-domain accuracies for relations other than *isSuccessor*. TransR maintains target-domain

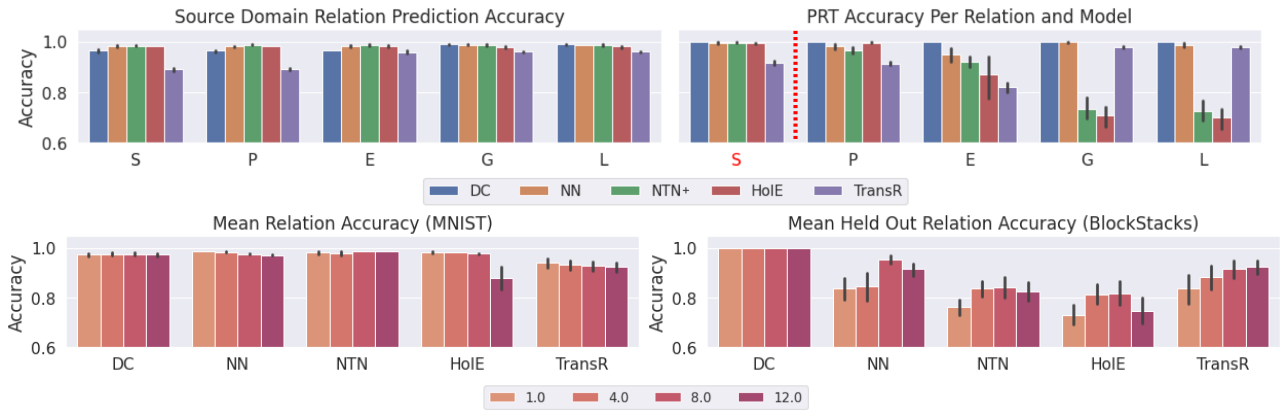


Figure 6.7: **[Top]** Relation-decoder prediction accuracy per model (DC, NN, NTN⁺, HoIE, TransR) and relation (abbreviated on the x -axis), in the source domain (MNIST, left) and target domain (BlockStacks, right). A red highlighted S and dotted line (top right) indicates that the relation-decoder for the *isSuccessor* relation is included in ϕ^{guide} and used to train the target domain autoencoder, but none of the other relation-decoders are. Both DC and NN retain a good performance while all other models show a decrease of accuracy in the target domain for one or more of the relations not included in training. **[Bottom]** Impact of different values of $\beta \in \{1, 4, 8, 12\}$ for each relation-decoder averaged across all relation-decoders in the source domain (left) and $\phi^{\text{held-out}}$ held-out relation-decoders for relations {P, E, G, L} in the target domain (right). It can be seen that DC is not impacted by changes in β and it maintains performance in the target domain. All other models show a decrease of accuracy for the held-out relation-decoders in the target domain.

accuracies similar to its performance in the source domain, but this is significantly below the performance of other models in the source domain. The impact of adjusting β (disentanglement pressure) can be seen in Figure 6.7-bottom. The left plot shows source ϕ^{guide} mean performance under each β setting, and the right plot shows target domain zero-shot PRT performance for the $\phi^{\text{held-out}}$ held-out relation-decoders. Barring DC which has little discernible change in either source or target domains, PRT performance is significantly impacted by β for all models in the target domain, but has little effect in the source domain. TransR shows a clear positive correlation between target domain accuracy and β values, whereas the remaining models produce their best PRT performance with medium β disentanglement pressure. Next we will look more closely at the consistency losses obtained using each relation-decoder and under different β settings.

Consistency and coherence loss breakdown

To gain a deeper insight as to which underlying characteristics can explain the observed PRT accuracies, Figure 6.8 and Figure 6.9 present Con-A and Con-I loss profiles, respectively, using the practical consistency loss presented in Section 6.3, for varied β . Results refer to both source (left) and target domain embeddings (right), but cover each region of latent space (for

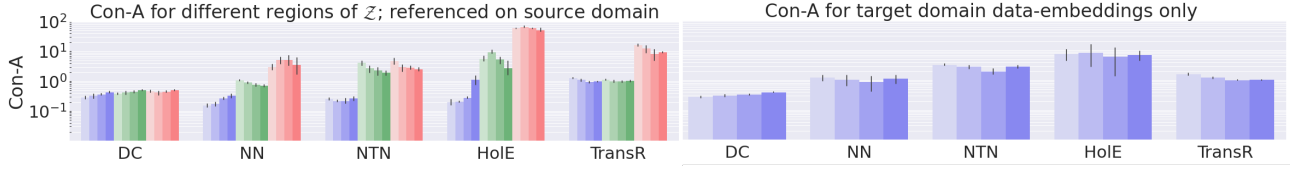


Figure 6.8: Consistency-Across (Con-A) losses (lower values are better) for the models (DC, NN, NTN⁺, HoIE, TransR) using the MNIST dataset (source domain $\mathcal{X}_s$) [left] and BlockStacks (target domain $\mathcal{X}_t$) [right]. The blue bars show the consistency loss of the data-embeddings, with darker shades corresponding to relation-decoders trained with higher β (disentanglement pressure). The green bars show the results for interpolation. The red bars show the results for extrapolation.

data-embeddings in blue, interpolation in green and extrapolation in red). Firstly, looking at Figure 6.8 we can see that DC retains excellent Con-A for all regions of latent space. TransR retains consistency from data-embeddings to the interpolated regions, but not to the extrapolated regions. The remaining relation-decoder models show degradation of consistency between data-embeddings and interpolation/extrapolation regions, with extrapolation often being worse than interpolation. However, NN achieves better interpolation Con-A than TransR, but not DC, and this is in agreement with their PRT accuracy profiles. Looking at β trends, aside from DC, increasing β appears to have a positive but limited effect on interpolation and extrapolation performance. Considering the Con-A performance of data-embeddings in the target domain, DC shows the best performance. The Con-A performance in the target domain is in agreement with the PRT accuracies, which is expected. But it is particularly interesting that, for all the relation-decoder models, Con-A performance in the target domain appears to match the interpolation or extrapolation Con-A performance in the source domain. This points to the possibility of anticipating PRT performance by evaluating the consistency of partial theories on regions of $\mathcal{Z}$ beyond the source data-embeddings. Such analysis is desirable as it can be performed in a downstream domain agnostic manner - only the source dataset and its data-embeddings are required to define the additional $\mathcal{Z}$ regions of interest, at which point the consistency loss can be calculated by sampling groundings directly from these $\mathcal{Z}$ regions.

Many of the same trends can be seen in the results for Con-I (Con-I-T, Con-I-A and Con-I-R) in Figure 6.9. As in Figure 6.8, results are presented with respect to source domain (left) and target domain (right), but here results are averaged over individual relations. We can firstly observe that DC and NN share the best overall Con-I performance profiles, with TransR following closely. DC and TransR again show comparable data-embedding versus interpolation/extrapolation CON-I, whereas NN, NTN⁺, and HoIE suffer from degradation. With regards to β 's impact, DC is not affected by β , while NN and NTN⁺ show a negative correlation between β and Con-I

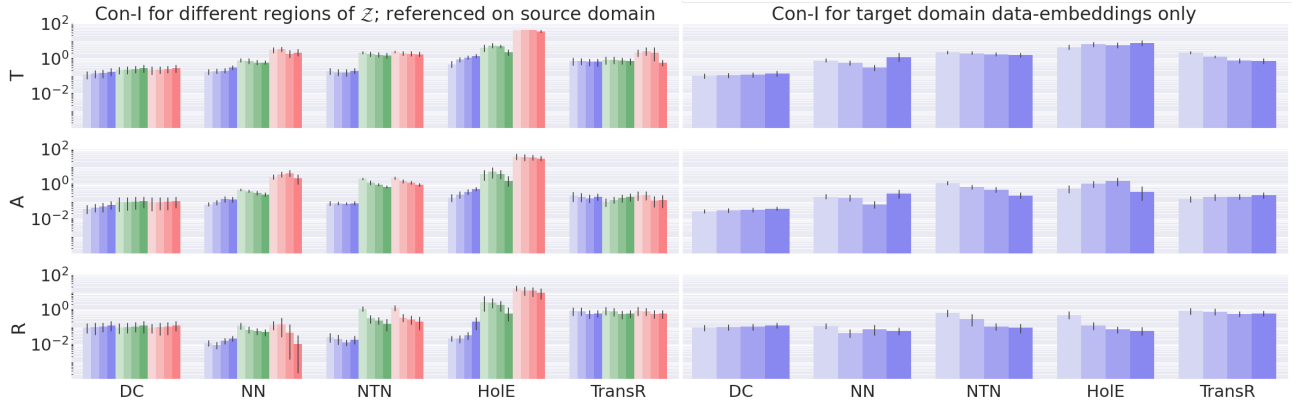


Figure 6.9: Consistency-Individual (Con-I) losses (lower values are better) for the models (DC, NN, NTN⁺, HolE, TransR) using the MNIST dataset (source domain $\mathcal{X}_s$) [left] and BlockStacks (target domain $\mathcal{X}_t$) [right]. From top to bottom (following the same colour schematic as Figure 6.8): Con-I-T (Transitivity), Con-I-A (Asymmetry) and Con-I-R (Reflexivity).

Table 6.3: Coherence proxy comparison with respect to source and target data-embeddings. Results include coherence loss $\mathcal{L}^{\bar{\epsilon}}$ and proxy $\bar{\epsilon}$. $\mathcal{L}^{\bar{\epsilon}}$ values are obtained by first taking the maximum data-embeddings consistency loss between soft-substructures for each source β setting. Then, the best (lowest) of these values is reported, together with their underlying $\beta = \beta^*$ (in parenthesis). The $\bar{\epsilon}$ values are calculated for each partial theory $\mathcal{L}^{\bar{\epsilon}}$ value via $\bar{\epsilon} \geq 1 - e^{-\mathcal{L}^{\bar{\epsilon}}}$ (c.f. Equation 6.20). The $\mathcal{T}^*$ abbreviations refer to: Con-A(cross) ($\mathcal{T}^A$), Con-I-T(ransitivity) ($\mathcal{T}^{I-T}$), Con-I-A(symmetry) ($\mathcal{T}^{I-A}$), Con-I-R(eflexivity) ($\mathcal{T}^{I-R}$) and Aggr(egate), which gives the best obtained summed consistency losses only.

ϕ^σ	Aggr.		$\mathcal{T}^A$			$\mathcal{T}^{I-T}$			$\mathcal{T}^{I-A}$			$\mathcal{T}^{I-R}$		
	$\mathcal{L}^{\bar{\epsilon}}$	(β^*)	$\mathcal{L}^{\bar{\epsilon}}$	$\bar{\epsilon}$	(β^*)	$\mathcal{L}^{\bar{\epsilon}}$	$\bar{\epsilon}$	(β^*)	$\mathcal{L}^{\bar{\epsilon}}$	$\bar{\epsilon}$	(β^*)	$\mathcal{L}^{\bar{\epsilon}}$	$\bar{\epsilon}$	(β^*)
HolE	12.12	(1)	6.61	1.00	(8)	4.30	0.99	(1)	0.52	0.41	(12)	0.08	0.08	(8)
NTN+	4.11	(8)	1.92	0.85	(8)	1.50	0.78	(12)	0.22	0.20	(12)	0.09	0.09	(12)
TransR	2.51	(8)	1.02	0.64	(12)	0.71	0.51	(12)	0.18	0.16	(4)	0.55	0.42	(8)
NN	1.71	(8)	0.82	0.56	(8)	0.44	0.36	(12)	0.18	0.16	(4)	0.05	0.05	(4)
DC	0.53	(1)	0.29	0.25	(1)	0.11	0.10	(1)	0.04	0.04	(1)	0.09	0.09	(1)

losses with comparable results for each underlying partial theory.

Finally, Table 6.3 provides a comparison between best coherence loss achieved for each relation-decoder model, as defined in Section 6.3. Results are partitioned according to each consistency type and also include the aggregate (summed) value. According to the $\bar{\epsilon}$ values, DC clearly outperforms all other relation-decoder models in terms of $\bar{\epsilon}$ -coherence on all partial theories, barring reflexivity, where NN performs slightly better. The aggregated loss values show that DC achieves markedly lower aggregate coherence loss than the other models. NN achieves strong aggregated loss, followed by TransR and NTN⁺, with HolE performing generally worse. Looking at β^* profiles, we see that most models achieve optimum aggregate coherence loss at $\beta = 8$, apart from DC and HolE, which perform better at $\beta = 1$. Overall, this is in broad agreement with the β profiles given by Figure 6.7-bottom (right). However, we can see that β^* profiles

for Con-A coherence are in more direct agreement, as TransR achieves its best at $\beta = 12$ and HolE at $\beta = 8$. This suggests that Con-A is more indicative of PRT performance, which could be expected since from a FOL standpoint, PRT can be achieved through deductive inferences across relations.

6.5 Discussion

The results presented in this chapter paint a picture in which source domain accuracy alone is not a strong enough indicator of PRT performance. Specifically, we see that even if it is possible to learn relation-decoders with close to 100% accuracy in the source domain, and establish similar accuracy for ϕ^{guide} in the target domain, this does not imply good zero-shot transfer of accuracy to $\phi^{\text{held-out}}$ in the target domain. But from a FOL standpoint, we know that the relation that ϕ^{guide} approximates, namely **isSuccessor** (S), should be sufficient to determine an ordering over a new domain providing positive examples exist for each domain element, which is indeed the case for the experiments at hand. Indeed, with close to 100% test accuracy for ϕ^{guide} in the target domain, it seems that the target domain is indeed ordered according to **isSuccessor**. We also know that through implication rules in the theory of ordinality that we are working with, from this ordering all other relations can be deduced from **isSuccessor**. All in all, these results demonstrate a gap between formal Logic and logic-like DL, which cannot be understood solely through the lens of accuracy (Goyal and Bengio, 2022).

Consistency as a deeper quality measure

Since reaching high levels of test accuracy is not enough to produce similar levels of PRT performance, we must look elsewhere to understand the results. Table 6.2 offers a possible explanation. A notable observation is a strong Spearman rank correlation between interpolated consistency loss and PRT performance, as well as a moderate rank correlation between extrapolated consistency loss and PRT performance. In contrast, there is only a weak rank correlation between source data-embedding consistency and PRT performance. Achieving high source domain accuracy necessitates low data-embedding consistency loss, since the training objective is a consistent structure (*i.e.* a model) of the ordinal theory. The fact that achieving consistency over data-embeddings alone does not correlate with PRT performance is in agreement with the

fact that relation prediction accuracy over the source domain is not a strong enough indicator of PRT performance. Instead, it seems that relation-decoders that are able to generalise consistency to other regions of latent space, in particular the interpolated region with respect to the data-embeddings, are more likely to achieve strong PRT performance, and even extrapolation consistency is more indicative than data-embeddings consistency. This could be expected since source and target domain data-embeddings are unlikely to coincide precisely. It is worth noting that the apparent greater importance of interpolated consistency versus extrapolated consistency may be circumstantial and merits further experimentation. One hypothesis is that this pattern can be explained by the fact that both datasets have 10 unique ordered elements (digits 0-9 for MNIST, and block heights 1-10 for BlockStacks). It may well be that relative importance of interpolation versus extrapolation would differ if the ordinal ranges of each dataset are different. For example generalising to orderings beyond the range of the source dataset may well place a greater importance on extrapolated consistency.

Empirical evidence for disentanglement as an inductive bias for coherent soft-structure learning

One result that is particularly interesting is the boost in PRT performance that disentanglement can offer to NN. In Section 5.4, we saw that disentanglement can be a useful inductive bias for coherent soft-substructure learning, since it may offer a natural ground-truth target for data representations. This can be achieved in two complimentary ways: through biasing an encoder towards learning invariances in the data, namely the so-called EFs, that correspond with a background representation $\mathcal{W}$; or through biasing relation-decoder towards learning disentangled families of relations that correspond with the underlying data EFs. These are complimentary because one approach to obtaining the same invariance properties in $\mathcal{Z}$ with respect to $\mathcal{W}$ is to encode each EF into independent subspaces in $\mathcal{Z}$ (Higgins et al., 2019), and this is what a VAE inductively biases. As shown in the design of DC (Section 6.1), ensuring that relation-decoders have domains that are either orthogonal (different family) or fully overlapped (same family) can satisfy the invariance conditions for disentangled families of relations, whilst being more robust to consistency with respect to arbitrary regions of $\mathcal{Z}$ and thus coherence between those regions. In representing the EFs in different vector subspaces, and with relation-decoders that learn disentangled families of relations, suitably defining the semantics of these subspaces, we

may obtain a system that is more able to learn coherent soft-substructures by guiding other encoders through a narrower hypothesis space⁸. A core design focus of DC was its ability to learn such families. However NN does not offer the same inductive bias, and so it may be the case that disentanglement is a necessary inductive bias to encourage NN to obtain the same disentanglement grouping of relation-decoders, leading to more robust consistency over $\mathcal{Z}$. Then, disentangling the ordinal feature into this same family-specific domain enables robust fitting with respect to the coherence guarantees of Chapter 5.

Summarising the above findings, it seems that 100% accuracy on the test-set is not enough (Goyal and Bengio, 2022). Instead, the empirical results presented in this chapter firstly suggest that consistency evaluation with respect to the source domain can be used to diagnose relation-decoder characteristics, offering a distinction between those that have learned a coherent and more logic-like notion of some concept (defined by a family of relations) and those that have not. Alongside this diagnosis, results also suggest that, in agreement with Chapter 5, disentanglement may be a key inductive bias for coherent soft-substructure learning.

⁸Perhaps resulting with an encoding that indeed satisfies Equation 5.38, but this is premature to say without further investigation.

Chapter 7

Related Work

This chapter overviews work related to the experiments and theoretical foundations presented in this thesis. The experimental work in this thesis can be viewed as a set of investigations into the learnability of explicit FOL-like relation modules, under different learning conditions: in Chapter 3 this was done by training a DRL agent in an RL setting, and in Chapters 4 and 6 through weakly supervised representation learning using VAEs. Additionally, the theoretical work in Chapter 5 provides a framework for evaluating the FOL-like behaviour of these relation-decoders. The related work is therefore broken up into these three areas: RRL and its variants, where the focus is primarily on learning relations within an RL setting; FOL-like relation learning from latent representations and associated theoretical frameworks; and finally, relevant approaches to weakly supervised representation learning using VAEs. Additionally, a short review of alternative disentanglement metrics to the MIG score is included.

Relational reinforcement learning

The incorporation of FOL into RL has been explored in a number of ways. Dzeroski et al. (2001) introduced RRL, in which the environment is represented using FOL relation symbols, towards improved generalisation and planning. For instance, through the use of variables, a block stacking objective can be represented by: $\text{goal}(\text{on}(A, B))$, which can be ground with different constants to produce flexible goal states such as: “stack block a onto block b ” becomes semantically similar to “stack block c onto block d ”. In contrast, a propositional representation would represent each goal as a separate proposition and their similarity would not be captured,

and the then state-of-the-art RL would need to be retrained to deal with such a change in goal. Compared to Chapter 3, both states and actions are initially expressed in symbolic form, and a policy is learned using Q-learning with relational regression trees which take as input the relational state, goal (target end state) and action. Other approaches in which DL and FOL are coupled in some form, have been developed since. Kimura et al. (2021) use a semantic parser to extract a FOL symbolic state representation from natural language states, which are then passed into a Logic-inspired ANN to compute a policy. Evans et al. (2022) integrate raw sensory input into a FOL state representation, where their model learns an explicit symbolic theory to explain the transitions between a sequence of visual states. Here the final representation is again a pure FOL representation, in which uncertainty is represented by $\vee$ between possible relation symbols. This work relies on data being represented as binary (on/off) pixel, appropriate for *e.g.* MNIST, but not for more complex environments, such as those present in ATARI games. Jiang and Luo (2019) developed their Neural Logic Reinforcement Learning implementation, where a FOL state representation is abstracted from an initial set of FOL symbols, using a proposed variant of δ ILP (Evans and Grefenstette, 2018). Results showed robust transfer as the training distribution shifts, together with interpretable policies. However, the work relies on a base set of relation symbols being predefined, and so does not learn any relations directly from raw visual input. Lastly, Konidaris et al. (2018) are able to learn propositional abstractions from grouping start and end states, associated with actions in a goal oriented RL task. Though not FOL, this work was able to associate real-world 3-dimensional voxel based states into propositions and use these to improve downstream planning.

Moving towards more difficult domains, Zambaldi et al. (2019) produced an entirely Deep Relational Reinforcement Learning implementation, wherein raw images are encoded as a set of vectors and an iterative message-passing procedure is applied to model interactions between entities. Their experiments included a **Box-World** grid-based game which required sequences of color-directed interactions between objects in order to unlock a specific box and obtain a gem. It was found that a standard deep Q-learning approach could not reliably solve the game, whereas their relational inductive-biases led to reliable solving. The work also demonstrated that relational inductive-biases far outperform competitive relation-less RL methods on a number of complex StarCraft II mini-games. Though relational at its core, the relations are not explicitly represented. Instead transformers (Vaswani et al., 2017) are used to encode entity-entity interactions into attention-weighted entity representations via message-passing. Many attention

heads can be employed, each able to learn a different relational-query under-which to compute relation interactions. The architecture demonstrated impressive zero-shot transfer performance over a standard CNN baseline, on tasks that involved longer sequences of decision making and novel combinations of entities. Investigating whether explicit relations are learned by transformers and how consistency and coherence evaluation may be applied to them is an interesting direction of future work.

Most similar to the approach taken in Chapter 3 are works by Garnelo et al. (2016) and Badreddine and Spranger (2019). Garnelo et al. (2016) proposed a Deep Symbolic Reinforcement Learning implementation, which represents state symbolically starting from low-level sensory input. The approach used CNN feature maps to produce entity representations, trained using a simple AE, and then employ hand-designed heuristics to produce a symbolic state representation. The policy is learned via tabular Q-learning, over a series of Q-functions, one defined for each object-type interaction. The action that maximizes the aggregate Q-value over all Q-functions is taken. Whilst Garnelo et al. (2016) re-emphasizes the value of symbolic representations in RL, and outlined some of the key computational processes required, it does not offer a method for end-to-end grounding of relations directly from observations, but instead relies on the availability of hand-designed heuristics. Nonetheless, this work demonstrates the value of a symbolically augmented DRL pipeline over more standard DRL Q-learning. Badreddine and Spranger (2019) were motivated by (Garnelo et al., 2016) and develop a more advanced pipeline, wherein the symbolic layer could be extracted from the images using relation-decoders. However, in contrast to the approach taken in Chapter 3, the relation-decoders are pretrained. The model receives both the image-feature maps and symbolic abstraction as input to a DRL model, together with previous timestep detections. The game is discretized into grid-square feature maps, as in (Garnelo et al., 2016), where each channel of the map is an object-type predicate output. Unary object predicates are $\{\text{agent}(X), \text{circle}(X), \text{square}(X)\}$ and additional unary predicates $\{\text{avoid}(X), \text{goto}(X)\}$ are also included. The paper then uses a Logic Tensor Network (LTN) (Serafini and d’Avila Garcez, 2016; Badreddine et al., 2022) to express the axiomatic background knowledge that determines which objects are rewarding and which should be avoided, for instance $\forall A (\text{circle}(A) \leftrightarrow \text{goto}(A))$ and $\forall A (\text{cross}(A) \leftrightarrow \text{avoid}(A))$. The work demonstrates that passing invariant object-type groundings (given as background knowledge) means that the model is more resilient to changes in game colour settings. Some re-learning is still required as the raw visual feed is passed into the action-selection module, but the

performance drop diminishes over time, whereas a baseline DRL model is shown to repeatedly hit catastrophic forgetting breakdowns. In second experimental setting, the association of object-type to reward/penalty is repeatedly altered. It is shown that including the `goto(A)` and `avoid(A)` relations increases training speed and greatly diminishes performance drop-off. This presents an interesting direction for future work for this thesis, where incorporating background knowledge in the form of sentences implemented using LTNs may improve the robustness of the unsupervised relation-decoding layer in Chapter 3.

First-order logic abstraction from deep learning representations

A second relevant field of work to this thesis is Knowledge Graph Embedding (KGE) using Latent Factor/Feature Models (LFMs) (Trouillon et al., 2019; Wang et al., 2021; Nickel et al., 2016a). Similarly to the theory behind disentanglement, LFMs take an entity oriented approach to learning representations of relations, where it is assumed that each dimension/collection of dimensions of the entity representations is semantically informative. LFMs are commonly used to perform link prediction over knowledge graphs, where the goal is to predict the missing relation between two entities. This is achieved by jointly learning representations of both relation-decoders and entities, typically by treating entity vectors as trainable parameters, and so not incorporating an encoder from some other fixed representation. However, within this branch of literature, Graph Neural Network (GNN) methods can be used to perform KGE by using a GNN implemented autoencoder architecture (an early version presented in (Schlichtkrull et al., 2018)). Here an encoder is used to learn the entity embeddings, through relation-specific message-passing, and relation-decoders are then used to predict the missing relation between two entities (see (Wang et al., 2021) for a survey of such methods). Incorporating relation-based encoding using GNNs is a very interesting direction for future work (Battaglia et al., 2018), as this offers a way to incorporate non-visual semantics into the latent representation, allowing non-visual EFs to also be represented. Overall, a LFM learns an inherently relational representation and inspired the use of relation-decoders in this thesis. The term ‘relation-decoder’ is not used in LFM literature, but since the LFM refers to the overall model, the term ‘relation-decoder’ is used to refer to the decoder component of the LFM.

To predict a link between a set of entities means learning whether a particular relation holds between them. This naturally lends itself to FOL, where certain consistency patterns may be

present in the knowledge graph. LFM literature evolved to explore this, where the expressive capacity of a relation-decoder is a key part of the model (Kazemi and Poole, 2018). Typically the concern is with their ability to model different consistency-individual characteristics (Trouillon et al., 2019). This was explored in more depth by Gutiérrez-Basulto and Schockaert (2018) (and built upon in (Schockaert and Gutiérrez-Basulto, 2021; Bouraoui et al., 2022)), who perform a geometric analysis of relation-decoder expressive capacity, by introducing the notion of *geometric interpretations* and *geometric models*. Using these concepts, they show that many state-of-the-art relation-decoders cannot be used to model certain FOL formulae without inducing unwanted implication side-effects. This work inspired the development of DC and the idea that ϵ -coherence could rely on ϵ -consistency being interpolated/extrapolated across broader regions of latent space beyond specific data-embeddings.

Building on LFMs, in particular the FOL-like expressiveness of relation-decoders, the aforementioned LTN can be viewed as wrapping a mathematical syntax around relation-decoder outputs, in order to produce a more expressive loss function (Serafini and d’Avila Garcez, 2016; Badreddine et al., 2022). This is achieved using their Real Logic, in which a grounding operator is used in place of an interpretation function. The role of the grounding operator is to associate tensors of real numbers with terms, and $[0,1]$ scalars to formulae and sentences, in a FOL language. For instance, grounding a domain leads to the assignment of tensors to domain elements, so if the domain refers to a set of images, the grounding operator is responsible for assigning an image its tensor representation. When applied to function/relation symbols, the grounding theoretically assigns an $[0,1]$ value to the pairing of each tuple of a given set of domain elements, with the function/relation symbol. In practice, this assignment can be *computed* by a corresponding relation-decoder, and so the grounding operator needed store these values and may instead be considered to be a parametrised function. This leads to groundings of formulae, where connectives are computed using fuzzy operations on the $[0,1]$ scalars assigned to the formulae/sub-formulae (which eventually leads to groundings of function and relation symbols). T-norms and t-conorms are used to compute conjunction and disjunction respectively, and fuzzy negation and implication are used to compute negation and implication, respectively¹. Grounding of sentences is achieved through quantifier operators, which implement aggregation over ground formulae. Importantly, the grounding operator can be differentiable, and so losses can be computed between the grounding of sentences and the target truth values.

¹See (van Krieken et al., 2022) for a critical analysis over possible choices of fuzzy operators.

In various works, *e.g.* (Serafini and d’Avila Garcez, 2016; Donadello et al., 2017), it was shown that including such losses can lead to significant improvements in FOL-like classifier accuracy. Differentiable Logic based losses and differentiable methods for grounding formula have also been explored in a number of other works, such as (Fischer et al., 2019; Xu et al., 2018; Riegel et al., 2020), and, together with the consistency loss presented in Section 6.3, these can be viewed as employing specific grounding operator instantiations². This presents very interesting directions for future work: firstly, in using Real Logic as an alternative approach to producing a consistency loss than the approach taken in Section 6.3; and secondly, in evaluating whether the inclusion of such losses can lead to improved ϵ -consistency/coherence of soft-substructures.

Lastly, (Shanahan et al., 2020b) presents a framework for end-to-end learning of propositional representations, through grounding of FOL-like relations. Here, relations are implemented using transformers (Vaswani et al., 2017), and the overall architecture is able to extract entities from multi-entity images using the transfer’s self-attention mechanism. This is a desirable property; whereas MNIST and dSprites dataset consist of a single object in each image, which can be taken as the entity of interest, this is an ideal scenario that is unlikely in real-world scenes consisting of many objects. However, each transformer implements many relations, and so it is less clear what each relation is, compared to relation-decoders used in LFMs.

Though other works implement a relational inductive bias in their architectures, such as the RN (Santoro et al., 2017) and associated WReN (Barrett et al., 2018) from Chapter 4, these are not considered immediate related work since they do not learn explicit FOL-like relations.

Weakly supervised representation learning with VAEs

There are a number of works that incorporate weak supervision over VAEs³. This became increasingly popular after Locatello et al. (2019) formally proved that it is impossible to perform unsupervised disentangled representation learning, unless inductive biases are applied to both the data and choice of generative model. Adding some degree of supervision over the generative model is one way to do this (Narayanaswamy et al., 2017; Locatello et al., 2020a; Hosoya, 2019; Bouchacourt et al., 2018; Chen and Batmanghelich, 2020, 2019; Brehmer et al., 2022; Dittadi

²Dash et al. (2022) provide a good recent review over these works.

³Following the taxonomy of Hernández-González et al. (2016), here weak supervision is being used as a catch-all for supervision in between fully supervised and fully unsupervised learning.

et al., 2021; Khemakhem et al., 2020; Karaletsos et al., 2016), and Locatello et al. (2020b) show that disentanglement can be achieved with only a very small amount of supervision, of the order of 100 or 1000 labels for datasets with the order of 100,000 instances. However, weak supervision with VAEs had been explored very close to their inception in (Kingma et al., 2014)

Particularly relevant to this thesis are the methods that incorporate mutual label constraints. Following Hernández-González et al. (2016)’s taxonomy of weak supervision, this setting groups examples together based on having some relationship between their class, which can be extended to a relationship between their EF values. Whilst some methods influence the grouping directly at the representation level, by modifying the ELBO under the assumption that parts of the representation are shared (content variable) and the rest not (style/transformations) (Bouchacourt et al., 2018; Hosoya, 2019), others employ additional functions to query the latent representation. For instance, (Locatello et al., 2020b) use a function to check if each latent dimension is ordered according to the true normalised EF variable, but only for a very small subset of the data. This enforces an ordinality over the latent dimensions, which was found to be important to disentanglement success when working with such a small fraction of data. Closer to the implementation of this thesis are (Chen and Batmanghelich, 2020, 2019; Karaletsos et al., 2016). Firstly, (Karaletsos et al., 2016) use crowd sourced annotations to produce qualitative triplet rankings of data, according to questions such as “Who has the most similar identity?”. They do this for a range of questions and jointly train a VAE together with the available annotations by including an additional triplet loss term. The truth of a triplet if drawn from $\mathbf{t}_{i,j,k} \sim p(\mathbf{t}|\mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_l)$ which states that the data encoded to $\mathbf{z}_i$ is more similar to that of $\mathbf{z}_j$ than $\mathbf{z}_k$. Distinct distributions are learned for each question, and the authors also incorporate a masking over $\mathcal{Z}$ to allow for specific subsets of latent dimensions to be used for each question. This inspired the use of a masking process in DC(Section 6.1.2). However, Karaletsos et al. (2016) use a dimension-wise sigmoid mask, whereas DC uses a Softmax over the entire latent space. Similarly to Karaletsos et al. (2016), but now looking specifically at pairwise relations, Chen and Batmanghelich (2020) and Chen and Batmanghelich (2019) use a relation-decoder where the dimensions of $\mathcal{Z}$ that are input to the relation-decoders is fixed. The former uses a relation-decoder that is suitable for a DC distance mode and inspired the definition of $\phi_{r,\sigma}^\dagger$ (Equation 6.10) but does not include a masking process, nor a bias term and so is less flexible and only able to look for distances centred around $\mathbf{0}$. However, the work only considered a single `isSimilar` relation applied to rotations and digit classes, and so the relation-decoder was

not required to be more flexible. The earlier work, Chen and Batmanghelich (2019), uses a relation-decoder that works on ordinal EFs, effectively implementing a $>$ relation via a relation-decoder of the form $\sigma(c \cdot (z_i - z_j))$, where c is a non-negative scaling factor. This inspired the DC polarity mode option $\phi_{r,\sigma}^\dagger$. However, in contrast to DC, Chen and Batmanghelich (2019) again uses a fixed association to the dimensions of $\mathcal{Z}$ and fixes the comparison around $\mathbf{0}$, and also enforce non-negativity on the scaling factor, overall making the function less flexible.

Disentanglement metrics

Although Chapter 4 uses the MIG score as a quantitative measure of disentanglement, it should be noted that others have been proposed. Some of the more popular include: the BetaVAE score (Higgins et al., 2017a), which uses a linear classifier to predict what EF value is fixed given a set of images that vary across all other EFs; the FactorVAE score (Kim and Mnih, 2018), which compares the empirical variance of the latent representation as a specific EF is fixed and all others varied, then the lowest highest variance dimension is chosen and a classifier is trained to predict the ground-truth EF values solely from this dimension; DCI Disentanglement (Eastwood and Williams, 2018), a measure that assigns a relative importance of a dimension to an EF, given a trained random forest regressor used to predict EF values; and the Separated Attribute Predictability score Kumar et al. (2018), which uses linear regression R^2 scores as a predictive importance score for each latent dimension against each EF, to then compare the average difference of the two top-most predictive dimensions for each EF. Generally, all quantitative metrics were found to be substantially correlated with one another (Locatello et al., 2019). The MIG score was found to be appropriate as it not only evaluates the degree to which an EF is represented by a single latent dimension, but, by looking at dimension-wise MI, can also produce a general spectrum of the representational importance of each dimension with respect to a given EF, as in Figure 4.9.

Chapter 8

Conclusion

This thesis has motivated the importance of consistent and coherent relational representations towards bridging the gap between current DL and human-like learning capacity. This has been achieved by evaluating the impact of relational inductive biases in a range of DL architectures and settings, comparing their performance against non-relational, or less relational, counterparts.

In Chapter 3, representing an RL environment in an entity-wise manner was shown to unlock a DL learner’s ability to continuously learn to represent a widening distribution of observations towards solving a simplified foraging problem. Furthermore, Chapter 3 showed that passing entity-representations through a set of relation-decoders led to good initial performance, with relation-decoders exhibiting a degree of FOL-like behaviour. Classification over mutually exclusive entity types demonstrated a tendency towards corresponding discrete mutually exclusive prediction with respect to one-another. This result suggests that FOL-like relation-decoders can be learned in an RL setting from reward alone. However, as the data distribution changes, even if freezing the abstract FOL-like layers, significant instability and performance degradation was observed, suggesting that the lower-level representation learner could not re-align its representations to the already formed FOL-like abstractions. This is contrary to conventional intuition, which tells us is that an abstract understanding of a concept (type classification) would be useful in making sense of future observations. Additionally, relation-decoders were shown to ‘flip’ their beliefs, in which a relation-decoder that resembled an `isShapeTypeA` classifier in one phase of learning, may later resemble a `isShapeTypeB` classifier. This suggests a freedom to switch the semantics of the learned relation, which is intuitively undesirable. In essence, the coherence of

he learned relation is questionable. Instead, we might expect a true FOL-like relation-decoder to latch onto specific semantics and not give way so freely.

In Chapter 4, the focus shifted to the FOL-like interaction between the lower-level latent representation and relation-decoders. By weakly supervising a set of relation-decoders with a ‘greedy’ representation learner (in this work a VAE) it was firstly possible to improve disentanglement of a range of Explanatory Factors (EFs), demonstrating a synergy between multi-relational decoding from greedy representations. Here, greedy refers to the fact that the VAE is trained to reconstruct the input, and so the latent representation is encouraged to retain as much observational information as possible. This is in contrast to the RL setting, where the latent entity representations serve to maximise reward, and so the encoder is encouraged to retain only the reward-relevant information. It was then shown that the VAE-supervised relation-decoders were able to learn to decode relations that were not explicitly present in the representation learning phase, but were implied by the weakly supervised relation-decoders. This is an element of FOL-like behaviour, where the representations should be amenable to the relation-decoding of logically associated relations. However, illustrating the mismatch between DL and symbolic representations, it was shown that downstream unary decoding of shape types could be performed to a far higher degree of accuracy than the relation-decoding of an `isSameShape` binary relation which was trained during the representation learning phase. This is again counter-intuitive, as identification of shape type should be dependent on the identification of the `isSameShape` relation. Instead, results suggested that the ability to reliably compare scales may be more important to shape type classification, than the ability to compare shape types, and merits further investigation.

To better understand how to evaluate the quality of a learned relational representation, Chapter 5 establishes theoretical foundations that build on prior work by Bengio et al. (2013) and Higgins et al. (2019). Specifically, the formal notion of a soft-structure was introduced as a soft analog of a FOL structure, which acts as theoretical relational representation construct and effectively *is* the relational representation. It was then formally defined what it means for a soft-structure to be consistent (Definition 5.3) and coherent (Definition 5.4), and how these properties may be obtained under the assumption that observations can be explained by sets of common EF that are generated from a background world-state. Consistency refers to the adherence of the soft-structure to logical theories, where a soft-structure should support (Definition 5.2) models

(in a model theoretic sense) of the logical theories that govern observations and decoded relations. Coherence extends this to the notion that a soft-structure should maintain this consistency over new observations, *i.e.* the logical understanding of a concept, described by sets of relations, should not be domain-dependent. Together these formally characterise the FOL-like quality of a soft-structure and, thus, a relational representation. This is a key property of FOL-like behaviour, where if a theory provides domain-independent rules for a concept, then a valid understanding of the concept should be equivalently be domain-independent. A prominent outcome of Chapter 5 is that, by Corollary 5.3, under an appropriate substitution of domain-dependent encoders, coherence guarantees are possible. This substitution has then been associated back to the theoretical foundation of disentanglement provided by Higgins et al. (2019), illustrating that under assumptions of a disentangled generative process this substitution may have a natural interpretation, further motivating the adoption of disentanglement in representation learning theory. The opportunity presented here is that in biasing the learner to disentangle, families of encoders may be more likely to arrive at the appropriate substitution and lead to coherent relational representations.

Finally, Chapter 6 builds on the theoretical foundations of Chapter 5 and develops a class of Dynamic Comparator (DC) relation-decoders designed to encourage practical learning of consistent and coherent soft-structures. The DC class of relation-decoders incorporate distinct functional modes that each have different FOL expressive characteristics (in terms of the FOL formulae they can support) which can be selected through learning to decode an intended relation. Chapter 6 then introduces an experiment PRT setup, in which success relies on FOL-like zero-shot transfer between relation-decoders and the underlying latent representation. More concretely, in FOL if two relations are associated through a sentence, for instance $\forall A, B \text{ (isSuccessor}(A, B) \rightarrow \text{isGreater}(A, B))$, and this sentence is satisfied in a given domain, then learning to adapt a new domain $\text{isSuccessor}(A, B)$ in which a successor ordering can be induced, should also lead to the ability to decode $\text{isGreater}(A, B)$, without any explicit re-training of relation-decoder parameters. The results showed that the consistency of the source domain soft-structure over broad regions of latent space, beyond the specific data encodings, was highly Spearman correlated with the downstream zero-shot PRT performance. This supports the theoretical foundations of Chapter 5, where it was formally shown that the ability to induce consistency over a broader domain could permit coherence between soft-structures (Corollary 5.2).

8.1 Future work

As is the nature of research, the work in this thesis has a wealth of natural continuations. First, the direct extensions will be considered, and then a more speculative discussion will follow.

Firstly, the experimentation performed in Chapter 3 ends with a surprising failure mode of end-to-end relational representations learned purely from reward. The breakdown of FOL-like behaviour in later phases of learning suggests that without additional bias, the relation-decoders are not learning in a coherent way - whilst their beliefs appear consistent with classification of mutually exclusive entity types, they become incoherent when the data distribution changes. To begin this work with some ideas as to where this additional inductive bias may come from, it is possible that background knowledge can be imposed on the relation-decoders through LTN expressions. For instance imposing through additional loss that the relation-decoders should learn to be mutually exclusive in their predictions. This may further assert their tendency to form FOL-like representations, and so may lead to robustness in terms of coherent behaviour, where the learned lower-level representations may attempt to maintain the FOL-like abstractions. Another idea is to incorporate the learnings from Chapter 4 into the LRL architecture. Specifically, it would be interesting to see if biasing the architecture to greedily retain information regarding the entities, for instance by using a VAE to reconstruct the entities, would lead to more coherent behaviour, in which the encoder may represent new EFs in a compositional manner. Here the work by Achille et al. (2018) is of interest, where it was shown that VAEs can perform lifelong representation learning using a mechanism to identify which EFs are relevant to data distributions at each point in time.

Extensions to the work in Chapter 4 are also possible. The most obvious is to extend the experimental investigation to explore other relations, datasets, and disentanglement metrics. A study of similar scale to *e.g.* (Locatello et al., 2019) would be a natural next step. Additionally, the abstract visual reasoning experiments in Chapter 4 showed the importance of background knowledge, in the form of relation labels pertaining to abstract non-visual semantics (in this case the numeric properties of digit symbols). One can take this further and consider visually unidentifiable properties, that may be relevant to a lifelong learner. For instance, if we were to look at pictures of two visually identical boxes, we would have no way of knowing which is heavier - perhaps one is filled with lead and the other with air. Only through interaction can

we obtain this knowledge, but without some mechanism for encoding this knowledge back into our representation of the objects, we would not be able to make use of it. This leads to some interesting architectural extensions. In particular, we should be able to encode background knowledge as another source of information. An interesting approach here would be to use a GNN as an encoder (Battaglia et al., 2018), following work by Schlichtkrull et al. (2018). In building such an architecture, it may be possible to store a long term memory knowledge graph, which captures the relations that hold over entities that have been interacted with. This knowledge graph can then be queried, by associating the entities in a state with the entities in the graph, then being able to pull in the background information pertaining to those entities, leading to a much more informative state abstraction. Lastly for Chapter 4, it is worth noting that the architecture employed in this thesis, in which a relatively standard VAE is applied to an image to produce an entity representation, relies on the observations containing a single entity, and this limits the scalability of the experimental work to more complex, real-world scenes. Ideas for architectural improvements include incorporating encoders that can identify and separate entities within a scene, for instance using transformers as was done in (Zambaldi et al., 2019), or encoders that use slot-attention (Burgess et al., 2019).

The theoretical foundations established in Chapter 5 stand on the shoulders of model theory, effectively bridging the gap between DL relational representations, described in terms of soft-structures, and formal structures in model theory. Model theory is a well studied field (Hodges, 1993), and so there are many opportunities to explore the implications of soft-structures in this context. In particular, Section 5.1.1 showed that achieving a high model-weight can reliably be obtained by supporting a single model of a FOL theory. Theories that permit a single model, up to isomorphism, for a given cardinality are said to be categorical in that cardinality. When a theory is categorical, it reveals some deep and significant properties about the theory itself. It means the theory provides enough information to uniquely determine the model, and means that the models with the corresponding cardinality all have the same properties. Moreover, if a theory is categorical in some infinite cardinality, then it is *complete*, meaning that for every sentence in the language of the theory, either the sentence or its negation is provable from the theory. This is because isomorphic models satisfy the same sentences, so there can be no sentence which is true in some models of the theory and false in others. Completeness, however, does not imply categoricity. Within this theme, Morley’s Categoricity Theorem may be of interest (Morley, 1965), which states that if a theory is categorical in some uncountable cardinality, then it is

categorical in all uncountable cardinalities. Looking at theories that describe the characteristics of EFs, in effect describing physical behaviour, it would be interesting to explore whether these theories are categorical in some cardinality. On the other hand, theories may permit many models and so it may be beneficial to explore how this can be exploited in the context of DL relational representations. A significant limitation in this regard is the independence assumption built into the definition and implementation of structure support (see Definition 5.2). By relaxing this, and developing relation-decoders that are aware (conditioned) on the beliefs of others, it may then be possible to support multiple models by orchestrating beliefs such that each belief leads the collection of relation-decoders towards supporting a particular model. By modifying the belief of one relation-decoder over particular observations, which may on its own be logically valid, the others may then be able to adapt their beliefs to support a different model when applicable. Lastly for Chapter 5, since supporting single models is the most reliable way to achieve high model-weights, Section 5.2 opted to support different models by fixing relation-decodings and instead modifying the encoder, leading to domain-specific encoders. An interesting direction for future work would be to repeat the coherence exploration under the reverse assumption (fixing the encoder and changing the relation-decoders). This is because the choice to fix relation-decoders means only a certain fixed set of models may be supported (each with different domains) and this may be too restrictive when considering non-ordinal theories, as has been the main focus of this thesis. This also points to future work that generally explores the opportunities and limitations of the proposed theory for broader theories. A possible avenue in this investigation could be found in adopting conditional beliefs.

Similarly, exciting extensions to Chapter 6 involve exploring PRT for other theories, and the experimental investigation could moreover be expanded, as proposed for Chapter 4, to include additional relation-decoder variants and datasets. Within this, investigating other mode definitions for DC could prove necessary. For instance, using a periodic function can allow for the encoding of periodic relations, such as rotational symmetries or the `isEven/isOdd` relations, which can be seen as a periodic relation over the natural numbers. Initial attempts at this were performed during this thesis, but training proved too difficult. Additionally, future work could explore how adding LTN based losses to the relation-decoder outputs may affect the results. In particular, it would be interesting to see if such losses may influence the overall coherence of the learned relation-decoders, perhaps through biasing them towards interpolating and/or extrapolating their consistency across broader regions of $\mathcal{Z}$. Exploring other forms of

bias towards this end would be similarly very interesting. Lastly for Chapter 6, the consistency loss presented in Section 6.3 merits further exploration, as the proposed implementation is not well suited to $\exists$ quantification, and does not consider situations where quantifiers may be nested. For the former, since the loss performs an aggregation over groundings, it does not appropriately penalise scenarios where a lack of a support of an existential should ‘break’ the ϵ -consistency. For instance, if a theory has the sentence $\exists A (\mathbf{p}(A))$, but a soft-structure only supports structures where $\forall A (\neg \mathbf{p}(A))$, then $\bar{\epsilon}$ may still be low. This is difficult to deal with, as existential quantification relies on exhaustive exploration over groundings. For the latter, the loss would need to deal with nested universal quantification and can face similar obstacles. For instance, as $\exists A (\mathbf{p}(A))$ is logically equivalent to $\neg \forall A (\neg \mathbf{p}(A))$, the same issue arises as with the outer existential quantifier.

Speculative future work

During the course of this thesis, many ideas for more speculative future work have been noted down. Within these, one stands out. DC demonstrated that data driven coupling between relation-decoder parameters can induce consistency between the learned relation-decoders, which is illustrated by Figure 6.3. Essentially, the FOL sentences can be embedded into the relation-decoder parameters, an idea that is implicit to the work by Gutiérrez-Basulto and Schockaert (2018), which investigates the geometric properties of relation-decoders and how these can support different FOL sentences. The parameters, however, start uncoupled, and the coupling emerges through training, a particularly valuable behaviour of DC. It may be possible to leverage couplings, in order to improve training efficiency. For instance, if the logical association of an `isGreater` and `isLess` relation is established through coupling, then parameter updates for one may be used to update the other. In a similar vein, consider a situation where two relation-decoders approximating `isHeavier` and `isLighter` have been learned and their domain is a 2-dimensional latent space, and we want to later learn a new set of relation-decoders for relations `isLouder` and `isQuieter`, which rely on encoding a different EF (sound intensity). Due to catastrophic forgetting, it is very possible that in learning the new relation-decoders, changes to the encoding may result previously learned relation-decoders losing their ability to decode the `isHeavier` and `isLighter` relations. One way to mitigate this is through joint training, for instance using memory replay techniques (Rebuffi et al., 2017). However, if the invariance of

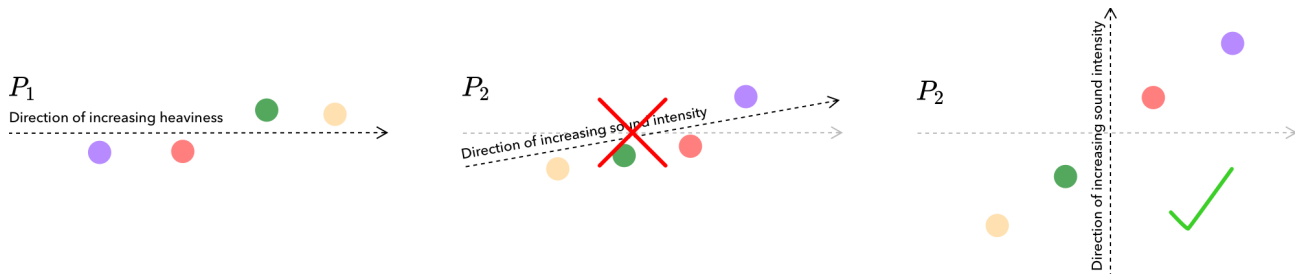


Figure 8.1: A illustration of how incremental learning of relation-decoders, over relations that should otherwise be invariant to one another, can lead to catastrophic forgetting. Instead, it is desirable that the new relation-decoders ‘pick’ an orthogonal direction to perform their comparison over, such that the directional ordering required by the previous relation-decoders is left unaffected. In the plot, P_i refers to a learning phase, as in Chapter 3.

relation-decoders can be identified, then it may be possible to induce direct parameter changes through appropriate couplings. In effect, if the past relation-decoders could inform the new ones of the invariance, then the new relation-decoder can preserve the invariance by learning a new relation-decoder that performs comparison using an orthogonal latent space direction to the previous ones. This is best understood visually in Figure 8.1, and furthermore illustrates the benefit of orthogonality of EF representations in $\mathcal{Z}$ common to disentanglement.

Finally, to finish this thesis let us very briefly look forward to where this work fits within the current remarkable progress in AI and specifically DL. Recently, the progress of Large Language Models (LLMs) has been phenomenal, demonstrating signs of Artificial General Intelligence (AGI) that only one or two years ago would have been considered out of reach for some time. However, there are significant signs that their success is not due to their ability to learn in a human-like manner, but rather due to their ability to learn from large amounts of data (Wu et al., 2023; Nye et al., 2021; Yang et al., 2023; Berglund et al., 2023), once again relying on pattern matching (albeit to a much more sophisticated degree) (Dziri et al., 2023). Possible avenues for improvement include the incorporation of more human-like inductive biases, through, for instance, *process supervision* (Lightman et al., 2023) and *chain-of-thought* prompting via the somewhat canonical “Let’s think step by step” prompt addon (Kojima et al., 2022). Nonetheless, the failure for LLMs to extract the logical regularities present in the very large data that they train on, suggests that further progress may rely on similar consistency and coherence based inquisitions to those presented in this thesis.

Bibliography

- Achille, A., Eccles, T., Matthey, L., Burgess, C. P., Watters, N., Lerchner, A., and Higgins, I. (2018). Life-long disentangled representation learning with cross-domain latent homologies. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9895–9905.
- Achille, A. and Soatto, S. (2018). Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19:1–34.
- Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Asai, M. (2018). Photo-realistic blocksworld dataset. *CoRR*, abs/1812.01818.
- Badreddine, S., d’Avila Garcez, A. S., Serafini, L., and Spranger, M. (2022). Logic tensor networks. *Artif. Intell.*, 303:103649.
- Badreddine, S. and Spranger, M. (2019). Injecting prior knowledge for transfer learning into reinforcement learning algorithms using logic tensor networks. In Doran, D., d’Avila Garcez, A. S., and Lécué, F., editors, *Proceedings of the 2019 International Workshop on Neural-Symbolic Learning and Reasoning (NeSy 2019), Annual workshop of the Neural-Symbolic Learning and Reasoning Association, Macao, China, August 12, 2019*.
- Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference*

on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Barnett, S. and Ceci, S. (2002). When and where do we apply what we learn? a taxonomy for far transfer. *Psychological Bulletin*, 128:612–637.

Barrett, D., Hill, F., Santoro, A., Morcos, A., and Lillicrap, T. (2018). Measuring abstract reasoning in neural networks. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 511–520. PMLR.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V. F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gülçehre, Ç., Song, H. F., Ballard, A. J., Gilmer, J., Dahl, G. E., Vaswani, A., Allen, K. R., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M. M., Vinyals, O., Li, Y., and Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261.

Baxter, J. (1995). Learning internal representations. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory, COLT '95*, page 311–320, New York, NY, USA. Association for Computing Machinery.

Bengio, Y., Courville, A. C., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.

Bengio, Y. and LeCun, Y. (2007). Scaling learning algorithms towards ai. In Bottou, L., Chapelle, O., DeCoste, D., and Weston, J., editors, *Large-Scale Kernel Machines*, pages 321–359. MIT Press, Cambridge, MA, USA.

Berglund, L., Tong, M., Kaufmann, M., Balesni, M., Stickland, A. C., Korbak, T., and Evans, O. (2023). The reversal curse: Llms trained on "a is b" fail to learn "b is a". *CoRR*, abs/2309.12288.

Bostrom, N. (2014). *Superintelligence: Paths, dangers, strategies*. Oxford University Press.

Bouchacourt, D., Tomioka, R., and Nowozin, S. (2018). Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In McIlraith, S. A. and Weinberger, K. Q., editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial*

- Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2095–2102. AAAI Press.
- Bouraoui, Z., Gutiérrez-Basulto, V., and Schockaert, S. (2022). Integrating ontologies and vector space embeddings using conceptual spaces (invited paper). In Bourgaux, C., Ozaki, A., and Peñaloza, R., editors, *International Research School in Artificial Intelligence in Bergen, AIB 2022, June 7-11, 2022, University of Bergen, Norway*, volume 99 of *OASICs*, pages 3:1–3:30. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- Brehmer, J., Haan, P. D., Lippe, P., and Cohen, T. (2022). Weakly supervised causal representation learning. In *ICLR 2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S. M., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.
- Buckner, C. J. (2023). *From Deep Learning to Rational Machines: What the History of Philosophy Can Teach Us About the Future of Artificial Intelligence*. Oxford University Press, New York, NY.
- Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2017). Understanding disentangling in β -vae. In *NIPS Workshop on Learning Disentangled Features*.

- Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M. M., and Lerchner, A. (2019). Monet: Unsupervised scene decomposition and representation. *CoRR*, abs/1901.11390.
- Caruana, R. (1997). Multitask learning. *Mach. Learn.*, 28(1):41–75.
- Chandar, S., Pascanu, R., and Precup, D., editors (2022). *Conference on Lifelong Learning Agents, CoLLAs 2022, 22-24 August 2022, McGill University, Montréal, Québec, Canada*, volume 199 of *Proceedings of Machine Learning Research*. PMLR.
- Chen, J. and Batmanghelich, K. (2019). Robust ordinal VAE: employing noisy pairwise comparisons for disentanglement. *CoRR*, abs/1910.05898.
- Chen, J. and Batmanghelich, K. (2020). Weakly supervised disentanglement by pairwise similarities. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3495–3502. AAAI Press.
- Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. (2018). Isolating sources of disentanglement in variational autoencoders. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2615–2625.
- Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2172–2180.
- Chen, Z. and Liu, B. (2018). *Lifelong Machine Learning, Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.
- DARPA (2017). Toward Machines that Improve with Experience.
- Dash, T., Chitlangia, S., Ahuja, A., and Srinivasan, A. (2022). A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, 12(1):1040.

- Dempster, A. (2008). The dempster–shafer calculus for statisticians. *International Journal of Approximate Reasoning*, 48(2):365–377. In Memory of Philippe Smets (1938–2005).
- Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., Yi, J., Zhao, W., Wang, X., Liu, Z., Zheng, H.-T., Chen, J., Liu, Y., Tang, J., Li, J., and Sun, M. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.
- Dittadi, A., Träuble, F., Locatello, F., Wuthrich, M., Agrawal, V., Winther, O., Bauer, S., and Schölkopf, B. (2021). On the transfer of disentangled representations in realistic settings. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Domingos, P. (2015). *The master algorithm: how the quest for the ultimate learning machine will remake our world*. Allen Lane, London.
- Donadello, I., Serafini, L., and d’Avila Garcez, A. (2017). Logic Tensor Networks for Semantic Image Interpretation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 1596—1602.
- Dzeroski, S., Raedt, L. D., and Driessens, K. (2001). Relational reinforcement learning. *Mach. Learn.*, 43(1/2):7–52.
- Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., West, P., Bhagavatula, C., Bras, R. L., Hwang, J. D., Sanyal, S., Welleck, S., Ren, X., Ettinger, A., Harchaoui, Z., and Choi, Y. (2023). Faith and fate: Limits of transformers on compositionality. *CoRR*, abs/2305.18654.
- Eastwood, C. and Williams, C. K. I. (2018). A framework for the quantitative evaluation of disentangled representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Eaton, E., Baveja, S., Grauman, K., Konidaris, G., Learned-Miller, E., Littman, M., Mataric, M., Sha, F., and Stone, P. (2022). Lifelong learning of perception and action in autonomous systems. Technical report, Trustees of the University of Pennsylvania.

- Eddahmani, I., Pham, C.-H., Napoléon, T., Badoc, I., Fouefack, J.-R., and El-Bouz, M. (2023). Unsupervised learning of disentangled representation via auto-encoding: A survey. *Sensors*, 23(4).
- Evans, R., Bosnjak, M., Buesing, L., Ellis, K., Pfau, D., Kohli, P., and Sergot, M. J. (2022). Making sense of raw input (extended abstract). In Raedt, L. D., editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5727–5731. ijcai.org.
- Evans, R. and Grefenstette, E. (2018). Learning explanatory rules from noisy data. *J. Artif. Intell. Res.*, 61:1–64.
- Feldman, J. (2013). The neural binding problem(s). *Cognitive neurodynamics*, 7(1):1–11.
- Fischer, M., Balunovic, M., Drachsler-Cohen, D., Gehr, T., Zhang, C., and Vechev, M. T. (2019). DL2: training and querying neural networks with logic. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1931–1941. PMLR.
- French, R. M. (1993). Catastrophic interference in connectionist networks: Can it be predicted, can it be prevented? In Cowan, J. D., Tesauero, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 1176–1177. Morgan Kaufmann.
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.
- Fukushima, K. and Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):455–469.
- Garnelo, M., Arulkumaran, K., and Shanahan, M. (2016). Towards deep symbolic reinforcement learning. *CoRR*, abs/1609.05518.
- Garnelo, M. and Shanahan, M. (2019). Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. *Current Opinion in Behavioral Sciences*, 29:17–23. Artificial Intelligence.

- Gensler, H. J. (2017). *Introduction to Logic*. Routledge, 3 edition.
- Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In Gordon, G. J., Dunson, D. B., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 315–323. JMLR.org.
- Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT Press.
- Goodfellow, I. J. (2017). NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.
- Goyal, A. and Bengio, Y. (2022). Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478.
- Greff, K., van Steenkiste, S., and Schmidhuber, J. (2020). On the binding problem in artificial neural networks. *CoRR*, abs/2012.05208.
- Gutiérrez-Basulto, V. and Schockaert, S. (2018). From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules. In Thielscher, M., Toni, F., and Wolter, F., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018*, pages 379–388. AAAI Press.
- Halford, G. S., Wilson, W. H., and Phillips, S. (2010). Relational knowledge: the foundation of higher cognition. *Trends in Cognitive Sciences*, 14(11):497–505.
- Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., Han, W., Huang, M., Jin, Q., Lan, Y., Liu, Y., Liu, Z., Lu, Z., Qiu, X., Song, R., Tang, J., Wen, J., Yuan, J., Zhao, W. X., and Zhu, J. (2021). Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

- Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034. IEEE Computer Society.
- Hernández-González, J., Inza, I., and Lozano, J. A. (2016). Weak supervision and other non-standard classification problems: A taxonomy. *Pattern Recognit. Lett.*, 69:49–55.
- Higgins, I., Amos, D., Pfau, D., Racanière, S., Matthey, L., Rezende, D. J., and Lerchner, A. (2019). Towards a definition of disentangled representations. In *Theoretical Physics for Deep Learning Workshop, ICML, Long Beach, CA*. arXiv preprint: 1812.02230.
- Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. (2017a). beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Higgins, I., Pal, A., Rusu, A. A., Matthey, L., Burgess, C. P., Pritzel, A., Botvinick, M. M., Blundell, C., and Lerchner, A. (2017b). DARLA: improving zero-shot transfer in reinforcement learning. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1480–1490. PMLR.
- Higgins, I., Racanière, S., and Rezende, D. J. (2022). Symmetry-based representations for artificial and biological general intelligence. *Frontiers Comput. Neurosci.*, 16:836498.
- Hodges, W. (1993). *Model Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press.
- Hodges, W. (2001). *Logic - An Introduction to Elementary Logic, 2nd Edition*. Penguin Books.
- Hornik, K., Stinchcombe, M. B., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

- Hosoya, H. (2019). Group-based learning of disentangled representations with generalizability for novel contents. In Kraus, S., editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2506–2513. ijcai.org.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org.
- Jeon, I., Lee, W., Pyeon, M., and Kim, G. (2021). IB-GAN: disentangled representation learning with information bottleneck generative adversarial networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7926–7934. AAAI Press.
- Jiang, Z. and Luo, S. (2019). Neural logic reinforcement learning. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3110–3119. PMLR.
- Karaletsos, T., Belongie, S. J., and Rätsch, G. (2016). When crowds hold privileges: Bayesian unsupervised representation learning with oracle constraints. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Kazemi, S. M. and Poole, D. (2018). Simple embedding for link prediction in knowledge graphs. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4289–4300.
- Khemakhem, I., Kingma, D. P., Monti, R. P., and Hyvärinen, A. (2020). Variational autoencoders and nonlinear ICA: A unifying framework. In Chiappa, S. and Calandra, R., editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28*

- August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2207–2217. PMLR.
- Kim, H. and Mnih, A. (2018). Disentangling by factorising. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2654–2663. PMLR.
- Kimura, D., Ono, M., Chaudhury, S., Kohita, R., Wachi, A., Agravante, D. J., Tatsubori, M., Munawar, A., and Gray, A. (2021). Neuro-symbolic reinforcement learning with first-order logic. In Moens, M., Huang, X., Specia, L., and Yih, S. W., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3505–3511. Association for Computational Linguistics.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3581–3589.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. In *NeurIPS*.
- Konidaris, G. (2019). On the necessity of abstraction. *Current Opinion in Behavioral Sciences*, 29:1–7. Artificial Intelligence.
- Konidaris, G. D., Kaelbling, L. P., and Lozano-Pérez, T. (2018). From skills to symbols:

- Learning symbolic representations for abstract high-level planning. *J. Artif. Intell. Res.*, 61:215–289.
- Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., and Steinbrecher, M. (2022). *Computational Intelligence - A Methodological Introduction, Third Edition*. Texts in Computer Science. Springer.
- Kumar, A., Sattigeri, P., and Balakrishnan, A. (2018). Variational inference of disentangled latent concepts from unlabeled observations. In *6th International Conference on Learning Representations, {ICLR}*, Vancouver, BC, Canada.
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building Machines That Learn and Think Like People. *Behavioral and Brain Sciences*, 40.
- LeCun, Y. (1989). Generalization and network design strategies. Technical Report CRG-TR-89-4, University of Toronto.
- LeCun, Y. (1998). The mnist database of handwritten digits.
- LeCun, Y., Bengio, Y., and Hinton, G. E. (2015). Deep learning. *Nat.*, 521(7553):436–444.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989a). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.
- LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1989b). Handwritten digit recognition with a back-propagation network. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 396–404. Morgan Kaufmann.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324.
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. (2023). Let’s verify step by step. *CoRR*, abs/2305.20050.
- Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on*

- Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2181–2187. AAAI Press.
- Liu, B. (2017). Lifelong machine learning: a paradigm for continuous learning. *Frontiers Comput. Sci.*, 11(3):359–361.
- Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4114–4124. PMLR.
- Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. (2020a). Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6348–6359. PMLR.
- Locatello, F., Tschannen, M., Bauer, S., Rätsch, G., Schölkopf, B., and Bachem, O. (2020b). Disentangling factors of variations using few labels. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Lu, L., Shin, Y., Su, Y., and Karniadakis, G. E. (2019). Dying relu and initialization: Theory and numerical examples. *CoRR*, abs/1903.06733.
- Marcus, G. and Davis, E. (2020). Insights for ai from the human mind. *Commun. ACM*, 64(1):38–41.
- Marcus, G. F. (2001). *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. The MIT Press.
- Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. (2017). dsprites: Disentanglement testing sprites dataset. <https://github.com/deepmind/dsprites-dataset/>.
- McCarthy, J. (1959). Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91. Her Majesty’s Stationary Office.

- McCarthy, J., Minsky, M., Rochester, N., and Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Mag.*, 27(4):12–14.
- McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- Mendez, J. A. and Eaton, E. (2021). Lifelong learning of compositional structures. In *International Conference on Learning Representations*.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- Morley, M. (1965). Categoricity in power. *Transactions of the American Mathematical Society*, 114(2):514–538.
- Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Fürnkranz, J. and Joachims, T., editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814. Omnipress.
- Narayanaswamy, S., Paige, B., van de Meent, J., Desmaison, A., Goodman, N. D., Kohli, P., Wood, F. D., and Torr, P. H. S. (2017). Learning disentangled representations with semi-supervised deep generative models. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5925–5935.
- Newcombe, N. S. (2013). Cognitive development: changing views of cognitive change. *WIREs Cognitive Science*, 4(5):479–491.
- Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016a). A review of relational machine learning for knowledge graphs. *Proc. IEEE*, 104(1):11–33.

- Nickel, M., Rosasco, L., and Poggio, T. A. (2016b). Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1955–1961. AAAI Press.
- Nie, W. and Patel, A. (2019). Towards a better understanding and regularization of GAN training dynamics. In Globerson, A. and Silva, R., editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 281–291. AUAI Press.
- Noether, E. (1915). The finiteness theorem for invariants of finite groups. *Mathematische Annalen*, 77:89–92.
- Nye, M. I., Tessler, M. H., Tenenbaum, J. B., and Lake, B. M. (2021). Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 25192–25204.
- OpenAI (2023). GPT-4 technical report. *CoRR*, abs/2303.08774.
- Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359.
- Paris, J. and Vencovská, A. (2015). *Pure Inductive Logic. Perspectives in Logic*. Cambridge University Press.
- Paris, J. B. (1994). *The Uncertain Reasoner’s Companion: A Mathematical Perspective*. Cambridge University Press.
- Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

- Piaget, J. (2005). *The Psychology of Intelligence*. Routledge and Kegan Paul.
- Platts, M. (1979). Ways of meaning. *Linguistics and Philosophy*, 4(1):141–156.
- Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Rebuffi, S., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5533–5542. IEEE Computer Society.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org.
- Riegel, R., Gray, A. G., Luus, F. P. S., Khan, N., Makondo, N., Akhalwaya, I. Y., Qian, H., Fagin, R., Barahona, F., Sharma, U., Ikbali, S., Karanam, H., Neelam, S., Likhyani, A., and Srivastava, S. K. (2020). Logical neural networks. *CoRR*, abs/2006.13155.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- Roskies, A. L. (1999). The binding problem. *Neuron*, 24(1):7–9, 111–25.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.
- Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P. W., and Lillicrap, T. (2017). A simple neural network module for relational reasoning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett,

- R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4967–4976.
- Schlichtkrull, M. S., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In Gangemi, A., Navigli, R., Vidal, M., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., and Alam, M., editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.
- Schober, P., Boer, C., and Schwarte, L. A. (2018). Correlation coefficients: Appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5).
- Schockaert, S. and Gutiérrez-Basulto, V. (2021). Modelling symbolic knowledge using neural representations. In Simkus, M. and Varzinczak, I., editors, *Reasoning Web. Declarative Artificial Intelligence - 17th International Summer School 2021, Leuven, Belgium, September 8-15, 2021, Tutorial Lectures*, volume 13100 of *Lecture Notes in Computer Science*, pages 59–75. Springer.
- Serafini, L. and d’Avila Garcez, A. S. (2016). Logic tensor networks: Deep learning and logical reasoning from data and knowledge. In Besold, T. R., Lamb, L. C., Serafini, L., and Tabor, W., editors, *Proceedings of the 11th International Workshop on Neural-Symbolic Learning and Reasoning (NeSy’16) co-located with the Joint Multi-Conference on Human-Level Artificial Intelligence (HLAI 2016), New York City, NY, USA, July 16-17, 2016*, volume 1768 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.
- Shakya, A. K., Pillai, G., and Chakrabarty, S. (2023). Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, 231:120495.
- Shanahan, M., Crosby, M., Beyret, B., and Cheke, L. (2020a). Artificial intelligence and the common sense of animals. *Trends in Cognitive Sciences*, 24(11):862–872.
- Shanahan, M., Nikiforou, K., Creswell, A., Kaplanis, C., Barrett, D. G. T., and Garnelo, M. (2020b). An explicitly relational neural network architecture. In *Proceedings of the 37th*

- International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8593–8603. PMLR.
- Shao, H., Yao, S., Sun, D., Zhang, A., Liu, S., Liu, D., Wang, J., and Abdelzaher, T. F. (2020). Controlvae: Controllable variational autoencoder. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8655–8664. PMLR.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.
- Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.
- Soatto, S. (2010). Steps toward a theory of visual information. Technical Report CSD100028, UCLA.
- Socher, R., Chen, D., Manning, C. D., and Ng, A. Y. (2013). Reasoning with neural tensor networks for knowledge base completion. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 926–934.
- Steenbrugge, X., Leroux, S., Verbelen, T., and Dhoedt, B. (2018). Improving generalization for abstract reasoning tasks using disentangled feature representations. *CoRR*, abs/1811.04784.
- Strömfelt, H., Dickens, L., d’Avila Garcez, A., and Russo, A. (2022). Formalizing consistency and coherence of representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 6873–6885. Curran Associates, Inc.
- Strömfelt, H., Dickens, L., d’Avila Garcez, A. S., and Russo, A. (2020). On the transferability of vae embeddings using relational knowledge with semi-supervision. *CoRR*, abs/2011.07137.
- Strömfelt, H., Dickens, L., d’Avila Garcez, A. S., and Russo, A. (2021). Coherent and consistent relational transfer learning with auto-encoders. In *NeSy*, pages 176–192.

- Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Thrun, S. (1995). Is learning the n-th thing any easier than learning the first? In Touretzky, D., Mozer, M., and Hasselmo, M., editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press.
- Thrun, S. and Mitchell, T. M. (1995). Lifelong robot learning. *Robotics Auton. Syst.*, 15(1-2):25–46.
- Träuble, F., Creager, E., Kilbertus, N., Locatello, F., Dittadi, A., Goyal, A., Schölkopf, B., and Bauer, S. (2021). On disentangled representations learned from correlated data. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10401–10412. PMLR.
- Treisman, A. (1996). The binding problem. *Current opinion in neurobiology*, 6(2):171–178.
- Trouillon, T., Gaussier, É., Dance, C. R., and Bouchard, G. (2019). On inductive abilities of latent factor models for relational learning. *J. Artif. Intell. Res.*, 64:21–53.
- Tschannen, M., Bachem, O., and Lucic, M. (2018). Recent advances in autoencoder-based representation learning. *CoRR*, abs/1812.05069.
- van Krieken, E., Acar, E., and van Harmelen, F. (2022). Analyzing differentiable fuzzy logic operators. *Artif. Intell.*, 302:103602.
- van Steenkiste, S., Locatello, F., Schmidhuber, J., and Bachem, O. (2019). Are disentangled representations helpful for abstract visual reasoning? In Wallach, H. M., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14222–14235.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

- Wang, M., Qiu, L., and Wang, X. (2021). A survey on knowledge graph embeddings for link prediction. *Symmetry*, 13(3):485.
- Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., and Deng, S.-H. (2019). Hyperparameter optimization for machine learning models based on bayesian optimizationb. *Journal of Electronic Science and Technology*, 17(1):26–40.
- Wu, Z., Qiu, L., Ross, A., Akyürek, E., Chen, B., Wang, B., Kim, N., Andreas, J., and Kim, Y. (2023). Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *CoRR*, abs/2307.02477.
- Xu, J., Zhang, Z., Friedman, T., Liang, Y., and den Broeck, G. V. (2018). A semantic loss function for deep learning with symbolic knowledge. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5498–5507. PMLR.
- Yang, Z., Ding, M., Lv, Q., Jiang, Z., He, Z., Guo, Y., Bai, J., and Tang, J. (2023). GPT can solve mathematical problems without a calculator. *CoRR*, abs/2309.03241.
- Yu, Z., Li, J., Du, Z., Zhu, L., and Shen, H. T. (2023). A comprehensive survey on source-free domain adaptation. *CoRR*, abs/2302.11803.
- Zambaldi, V. F., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D. P., Lillicrap, T. P., Lockhart, E., Shanahan, M., Langston, V., Pascanu, R., Botvinick, M. M., Vinyals, O., and Battaglia, P. W. (2019). Deep reinforcement learning with relational inductive biases. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In Fleet, D. J., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer.
- Zeiler, M. D., Taylor, G. W., and Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. In Metaxas, D. N., Quan, L., Sanfeliu, A., and Gool, L. V.,

editors, *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2018–2025. IEEE Computer Society.

Appendix

A Additional experimentation details

This section provides additional experimental details for the experiments conducted in this thesis. All architecture configurations are provided in Appendix B. All experiments were coded using the PyTorch library (Paszke et al., 2019) and used the ADAM optimiser (Kingma and Ba, 2015) with default parameters apart from the learning rate, which was varied between experiments¹.

Additional experimental details for Chapter 3

These experiments were designed to not require significant hyperparameter tuning, as the focus was on investigating the impact of each relational inductive biases. Where required, hyperparameters were tuned using Bayesian optimisation (Snoek et al., 2012) with the Optuna library (Akiba et al., 2019). For CNN (baseline) and EW-CNN, only encoder and policy learning rates were explored. The hypothesis was that the policy may benefit from slower learning than the encoder, as the encoder is required to learn a more complex representation and needs to adapt to distribution changes. For EW-CNN- ϕ architectures, the learning rate of the relation-decoders was also explored, in conjunction to the encoder and policy. Additionally, NTN⁺ (Equation 1) has a single k hyperparameter that was also tuned, within the range 1-4. Optimal hyperparameters for each architecture are provided in Table A.1.

¹Using the same notation (and not to be confused with the notation of this thesis), the default parameters used are $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$ and weight decay = 0.

Table A.1: Optimal Hyperparameters for Chapter 3 architectures. LR: Learning Rate

Model	ENCODER LR	POLICY LR	ϕ LR	k
CNN (baseline)	2.077×10^{-5}	3.883×10^{-5}	-	-
EW-CNN	9.955×10^{-5}	3.436×10^{-5}	-	-
EW-CNN- ϕ (P123)	2.687×10^{-5}	7.203×10^{-5}	9.473×10^{-5}	4
EW-CNN- ϕ (P1)	9.151×10^{-6}	7.145×10^{-5}	3.946×10^{-5}	3

Additional experimental details for Chapter 4

All experiments in Chapter 4 used an NTN⁺ relation-decoder with k set to 1.

Relation-decoding transfer and disentangled representations

Experiments here use only the **dSprites** dataset (2:1 train-test split²) and are divided into two settings: (a) where weak relational supervision is not included in the representation learning phase ($\lambda = 0$) and (b) where it is ($\lambda > 0$). Relation labels used to train $\mathcal{L}^\sigma$ are strictly sampled from the train split. However, since the dataset is considered to only be partially labelled, all facts are sampled from only 50% of the train-set. This means that the β -TCVAE does not get any relational supervision for 50% of the train data. For testing, samples are from the full test set, in order to obtain a better estimate of the test score. In case (b) σ^+ is trained part of the representation learning phase, using a 40% batch size in comparison to the β -TCVAE batch input. To fill the batch, relation-labels are sampled from the labelled data subset, where at least one of the data instances is within the β -TCVAE batch input either as first or second argument in the case of a binary relation-decoder. When instances are outside of the batch, they are encoded but without including these operations in the backpropagation graph. Note that the batch is shared between all relation-decoders and again split between positive and negative examples. Following (Chen et al., 2018) a β -TCVAE batch input size of 2048 was used.

During the source task, both β -TCVAE and the relation-decoders are trained with a learning rate of 1×10^{-3} , and are trained until an early stopping criteria of 12 consecutive evaluation-steps is reached. Lastly, all experiments use the same β -TCVAE set of hyperparameters, which were found to reliably obtain high MIG scores when presented with the full dataset in preliminary

²Following the reasoning in (Locatello et al., 2019), as a representation learning experiment, in which the representational qualities of the resulting representation are of more interest than generalisation, these experiments could have benefited from only using a train set. However the introduction of relation-decoders makes this more complex to justify and so a train-test split was used.

investigations. It was found that dropping all terms other than the Total Correlation in the β -TCVAE ELBO, as given by Equation 4.6, could consistently produce good results.

During the target phase, when relation-decoders are trained over a frozen β -TCVAE, the learning rate is increased to 1×10^{-2} and training is performed over 3000 data epochs, using the same early stopping criteria.

Abstract visual reasoning over non-visual semantics

Most of the experimental details for the abstract visual reasoning experiment are provided in Section 4.3.2. Additionally, learning rates for both the β -VAE³ and relation-decoders were set to 1×10^{-4} , and training was performed for up to 300,000 epochs but with a 5 evaluation-step early stopping rate (which was always triggered before the maximum number of epochs was reached), following best practices in (Locatello et al., 2019). In addition to these, the sampling process matches that of the relation-decoding transfer experiment, but relation-labels may be created from any data instance in the train-set. Furthermore, the β -VAE batch size is reduced to 64, and 10 relation-labels are produced for each trained relation-decoder, split between positive and negative relation-labels.

In order to compute the downstream reasoning task, a WReN was used. This is a purpose-built architecture designed for RPM tasks (van Steenkiste et al., 2019; Barrett et al., 2018). A WReN leverages a previous relational architecture Santoro et al. (2017) in order to compute pairwise interaction embeddings for each context-to-context tile pairing and context-to-answer. Using the original notation (not to be confused with the notation used in this thesis) the architecture uses two shared neural networks, g_θ and f_ϕ , one for interaction learning and another for overall scoring, which ensures that the same reasoning method is performed for each possible answer. Additionally, a feature projection h_γ is used to append panel positions onto encodings. In the experiments in this thesis, all encodings are produced by a pretrained β -VAE, as described in Section 4.3.2, upon which positional values are appended via h_γ .

³Which was used over the β -TCVAE, as described in Section 4.3.2.

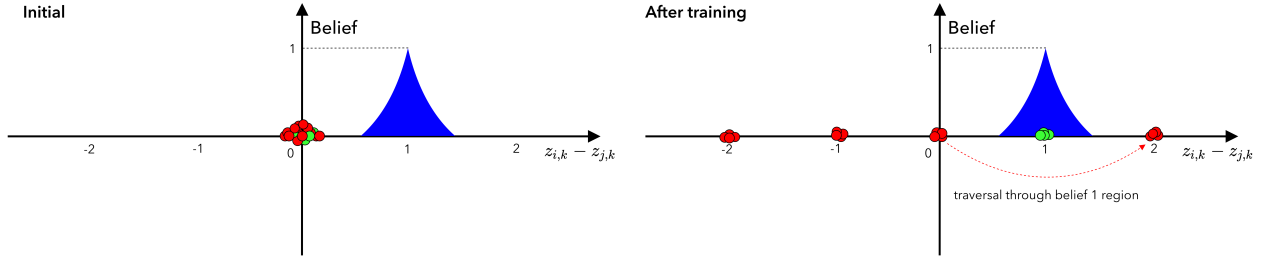


Figure A.1: Depiction of why DC requires flexible fitting for `isSuccessor` in target domain. In this figure it is assumed that ν_{ordinal} is encoded on the k th axis. The plot's x -axis thus shows the difference between the i th and j th encoding's k th dimension. The blue region represents $\phi_{\text{S,Cauchy}}^\dagger$'s corresponding belief over this difference space, in which $\phi_{\text{S,Cauchy}}^\dagger$ believes successors have a difference close to 1. This figure is for representational purposes and it is not required that $\phi_{\text{S,Cauchy}}^\dagger$ learns these specific successor intervals, but its design means that in the one-hot mask case, it will learn belief regions similar to the depiction. At initialisation, most VAE embeddings for $\mathcal{X}_t$ will be similar, meaning their differences are close to the origin. In which case $\phi_{\text{S,Cauchy}}^\dagger$ believes that no $\mathcal{X}_t$ encodings are successors. However, for the encodings to reach their appropriate successor intervals, in an incremental update procedure it is likely that encodings that are not successors, would need to pass through $\phi_{\text{S,Cauchy}}^\dagger$'s high belief region and be wrongly decoded. This leads to a local loss optima for $\hat{\mathcal{X}}_\sigma^t$ in which is it better to believe all $\mathcal{X}_t$ instances are not successors.

Additional experimental details for Chapter 6

Learning rates for this experiment were set to 1×10^{-4} for both the β -VAE and relation-decoders. The relation-label sampling strategy used in this experiment involves first selecting a ratio $R = \frac{|\mathcal{B}|}{|\mathcal{S}|}$ where $\mathcal{B} \subset \mathcal{S}_2$ is a set of O tuples. Then, relation-decoder specific relation-label subsets $\mathcal{B}_r$ where $|\mathcal{B}_r| = \frac{|\mathcal{B}|}{|\sigma|}$, to ensure a balanced distribution of tuples between relation-decoders. Furthermore, it was ensured that each $\mathcal{B}_r$ contains a balanced ratio of $\gamma_{\mathcal{S}_\sigma}^{r,O} = 1$ versus $\gamma_{\mathcal{S}_\sigma}^{r,O} = 0$ instances. It was found that each $|\mathcal{B}_r|$ set can be small without jeopardising the final relation-decoder performance level, allowing for the use of $R = 1$ for MNIST experiments and $R = 3$ for BlockStacks experiments. The β -VAE and relation-decoders are trained for up to 300,000 steps and an early stopping procedure was also used, where if the evaluation score does not increase over 30 and 120 training epochs for MNIST and Blockstacks experiments, respectively, the training is ended early.

Flexible fitting with Dynamic Comparator

As mentioned in Section 6.2, a flexible fitting procedure was used in the target PRT domain. This was done to ensure DC maximised accuracy for `isSuccessor` relation-decodings in the target domain. This procedure was necessary because a pretrained `isSuccessor` relation-decoder causes training to begin in a local loss optima. When using a pretrained `isSuccessor` relation-decoder,

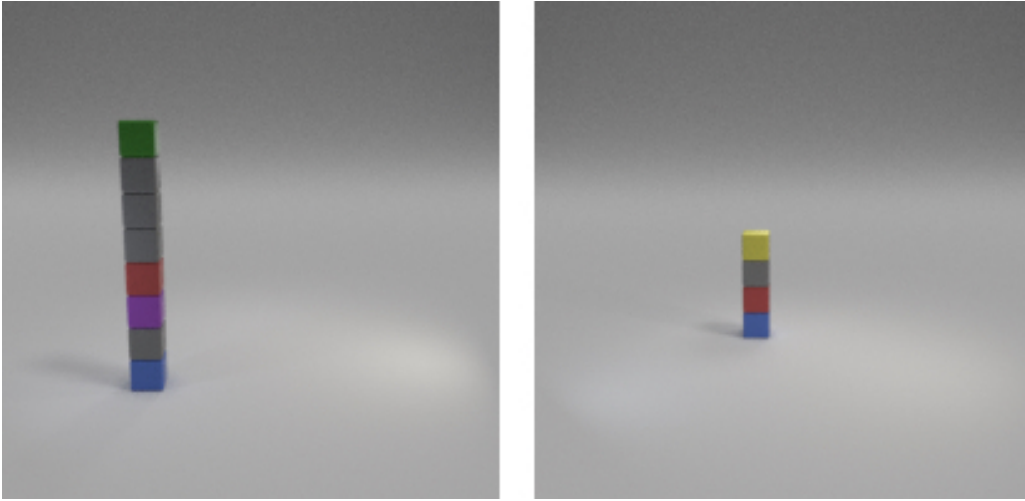


Figure A.2: Example of two **BlockStacks** dataset images. Each instance consists of a single red block varying in position within the block stack. On the left the red block is at height 3 (using a zero index) and on the right it is at height 1.

all beliefs are close to 0 at the beginning of a new VAE training phase, because the initialised VAE embeddings output close to zero vectors for all inputs. For the **isSuccessor** relation-decoder, this is actually not a bad set of beliefs in terms of accuracy, since most inputs are not successors of one-another. As such, the relation-decoder is already faced with a data imbalance that it's immediate beliefs are well suited to. To start to correctly decode true **isSuccessor** instances, it is necessary for the VAE's embeddings to move away from their $\mathbf{0}$ vector initial positions, and may lead to incorrect beliefs along the way towards producing correct beliefs. Overall this meant that the relation-decoder could fail to learn the correct belief regions for **isSuccessor** in the target domain from the very beginning of target domain $\psi_{\mathcal{X}_t}^{\text{enc}}$ training. This method was not applied to the other models as they were all able to achieve a similar **isSuccessor** relation-decoder performance level in the target domain as in the source domain.

BlockStacks dataset description

The **BlockStacks** data set consists of 12,000 RGB images ($200 \times 200 \times 3$ (Height $\times$ Width $\times$ Colour-Channels) pixels but resized in code to $128 \times 128 \times 3$) of individual block stacks, of varying height (between 1-10 blocks), block colours (uniformly sampled from options: {gray, blue, green, brown, purple, cyan, yellow}) and position (uniformly sampled from x, y range $(-3, -3)$ to $(3, 3)$), but with the requirement that each instance consists of a single red block at a random height (see Figure A.2 for example images). These were rendered using the **CLEVR** rendering agent with the help of code from (Asai, 2018). The dataset is divided into 9000:1500:1500 train,

validation and test splits.

B Architecture descriptions

This section provides architecture details for all experiments conducted in this thesis.

Relation-decoder implementations

TransR (Lin et al., 2015):

$$\phi_r^{\text{TransR}}(\mathbf{z}_i, \mathbf{z}_j) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2$$

with,

$$\mathbf{h}_r = \mathbf{M}_r \mathbf{z}_i \quad \text{and} \quad \mathbf{t}_r = \mathbf{M}_r \mathbf{z}_j.$$

where for $\mathbf{z}_i, \mathbf{z}_j \in \mathbb{R}^{d_z}$ vectors, $\mathbf{M}_r \in \mathbb{R}^{d_z \times d_z}$ and $\mathbf{r} \in \mathbb{R}^{d_z}$. As we want to obtain a $(0, 1)$ output, we modify TransR through $\phi_r^{\text{TransR}^+} = \sigma(c - \phi_r^{\text{TransR}})$, where σ is a sigmoid function and c is a scalar that ensures that at $\phi_r^{\text{TransR}}(\mathbf{z}_i, \mathbf{z}_j) = 0$, then $\phi_r^{\text{TransR}^+}(\mathbf{z}_i, \mathbf{z}_j) \approx 1$. In all experiments we set $c = 10$.

NTN⁺ (modified version of (Socher et al., 2013) from (Donadello et al., 2017; Serafini and d’Avila Garcez, 2016)):

$$\phi_r(\mathbf{z}_1, \dots, \mathbf{z}_n) = \sigma(\mathbf{u}_r^\top [\tanh(\mathbf{z}^{c\top} \mathbf{M}_r \mathbf{z}^c + \mathbf{V}_r \mathbf{z}^c + \mathbf{b}_r)]) \quad (1)$$

where $\mathbf{u}_r \in \mathbb{R}^k$, $\mathbf{M}_r \in \mathbb{R}^{n \cdot d_z \times n \cdot d_z \times k}$, $\mathbf{V}_r \in \mathbb{R}^{k \times n \cdot d_z}$ and $\mathbf{b}_r \in \mathbb{R}^k$. The only hyperparameter to consider is k , which controls the NTN⁺’s capacity - in all experiments, we set this to 1. If $k > 1$, $\mathbf{z}^{c\top} \mathbf{M}_r \mathbf{z}^c$ produces a k -dimension vector by applying the bilinear operation to each of the k $\mathbf{M}_r$ slices. Here $\mathbf{z}^c \in \mathbb{R}^{n \cdot d_z}$ is a concatenation of the inputs $\mathbf{z}_1, \dots, \mathbf{z}_n$, which was introduced in (Donadello et al., 2017; Serafini and d’Avila Garcez, 2016). In contrast, the original Neural Tensor Network (see (Socher et al., 2013)) is only applicable to binary relations and does not include the outer sigmoid.

HolE (Nickel et al., 2016b):

$$\phi_r^{\text{HolE}}(\mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{r}^\top(\mathbf{z}_i \star \mathbf{z}_j))$$

where $\mathbf{r} \in \mathbb{R}^{d_z}$ and $\star : \mathbb{R}^{d_z} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_z}$ denotes the circular correlation operator and is given by,

$$[\mathbf{z}_i \star \mathbf{z}_j]_k = \sum_{m=0}^{d-1} z_{i,m} \cdot z_{j,(k+m) \bmod d}$$

NN: a simple four-layer neural-network with layer sizes $l_{\text{in}} = 2d_z$, $l_1 = 2d_z$ and $l_2 = d_z$, with ReLU activations (Glorot et al., 2011; Nair and Hinton, 2010). The final output layer, l_{out} , is a single value passed through a sigmoid function, to bound the output within $(0, 1)$.

Reinforcement learning architectures

The encoder architectures for each RL experiment are provided in Table B.1, and the policy used for CNN (baseline) and EW-CNN is provided by⁴ Table B.2.

Table B.1: Specification for each RL encoder architecture parameters for **dSprites** 64×64×1 size input data. I: Input channels, O: Output channels, K: Kernel size, S: Stride, P: Padding, A: Activation, NEPB: Num Entities Per Batch

CNN (baseline) encoder Input: $64 \times 64 \times 1$	EW-CNN/EW-CNN-ϕ encoder Input: $64 \times 64 \times 1$
Layer_ID ; I ; O ; K ; S ; P ; A	Layer_ID ; I ; O ; K ; S ; P ; A
Conv2d_1 ; 1 ; 64 ; 4×4 ; 2 ; 1 ; ReLU	Conv2d_1 ; 1 ; 64 ; 4×4 ; 2 ; 1 ; ReLU
Conv2d_2 ; 64 ; 64 ; 4×4 ; 2 ; 1 ; ReLU	Conv2d_2 ; 64 ; 64 ; 4×4 ; 2 ; 1 ; ReLU
Conv2d_3 ; 64 ; 32 ; 4×4 ; 2 ; 1 ; ReLU	Conv2d_3 ; 64 ; 32 ; 4×4 ; 2 ; 1 ; ReLU
Conv2d_4 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; ReLU	Conv2d_4 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; ReLU
Layer_ID ; Num Nodes: In - Out ; A	Layer_ID ; Num Nodes: In - Out ; A
FC_z_1 ; $512 \times \text{NEPB} - 128 \times \text{NEPB}$; ReLU	FC_z_1 ; $512 - 128$; ReLU
FC_z_2 ; $128 \times \text{NEPB} - \text{Output Dim}$; -	FC_z_2 ; $128 - \text{Output Dim}$; -

VAE architectures

The VAE architecture configurations for both the β -TCVAE and β -VAE variants used in this thesis are provided in Tables B.3, B.4, and B.5.

⁴The EW-CNN- ϕ is illustrated in the main text by Figure 3.2

Table B.2: Specification of the CNN (baseline) and EW-CNN policy architecture parameters. Num Nodes: Number of Nodes, A: Activation, NEPB: Num Entities Per Batch

CNN (baseline)/EW-CNN policy	
Input: $10 \times \text{NEPB}$	
Layer_ID ; Num Nodes: In - Out ; A	
Processing_Layer_1 ; Input Dim - Input Dim ; ReLU	
Processing_Layer_2 ; Input Dim - Input Dim ; ReLU	
FC_1 ; Input Dim - Input Dim ; ReLU	
FC_2 ; Input Dim - Input Dim / 2 ; ReLU	
FC_Out ; Input Dim / 2 - 2 ; -	

Table B.3: Specification of the β -TCVAE encoder and decoder architecture parameters for **dsprites** $64 \times 64 \times 1$ size input data. I: Input channels, O: Output channels, K: Kernel size, S: Stride, P: Padding, A: Activation, BN: Batch Normalization (Ioffe and Szegedy, 2015)

Encoder	Decoder
Input: $64 \times 64 \times 1$	Input: $\mathbb{R}^{10}$
Layer_ID ; I ; O ; K ; S ; P ; BN ; A	Layer_ID ; I ; O ; K ; S ; P ; BN ; A
Conv1 ; 1 ; 32 ; 4×4 ; 2 ; 1 ; Yes ; ReLU	Conv1 ; 10 ; 512 ; 1×1 ; 1 ; 0 ; Yes ; ReLU
Conv2 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; Yes ; ReLU	Conv2 ; 512 ; 64 ; 4×4 ; 1 ; 0 ; Yes ; ReLU
Conv3 ; 32 ; 64 ; 4×4 ; 2 ; 1 ; Yes ; ReLU	Conv3 ; 64 ; 64 ; 4×4 ; 2 ; 1 ; Yes ; ReLU
Conv4 ; 64 ; 64 ; 4×4 ; 2 ; 1 ; Yes ; ReLU	Conv4 ; 64 ; 32 ; 4×4 ; 2 ; 1 ; Yes ; ReLU
Conv5 ; 64 ; 512 ; 4×4 ; - ; - ; Yes ; ReLU	Conv5 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; Yes ; ReLU
Layer_ID ; Num Nodes: In - Out ; A	Layer_ID ; Num Nodes: In - Out ; A
FC_z ; 512 - 10 ; -	FC_final ; 32 - 1 ; -

WReN architecture

The WReN architecture used in Chapter 4 is provided in Table B.6. The architecture used in this thesis is a factor 4 reduction in the parameter size of the WReN used in (van Steenkiste et al., 2019). Additional details and intuition regarding the WReN can be found in (Barrett et al., 2018).

Table B.4: Specification of the β -VAE encoder and decoder architecture parameters, for MNIST $28 \times 28 \times 1$ size input data. I: Input channels, O: Output channels, K: Kernel size, S: Stride, P: Padding, A: Activation

Encoder Input: $28 \times 28 \times 1$	Decoder Input: $\mathbb{R}^{10}$
Layer_ID ; I ; O ; K ; S ; P ; A Conv2d_1 ; 1 ; 32 ; 4×4 ; 2 ; 1 ; ReLU Conv2d_2 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; ReLU Conv2d_3 ; 32 ; 64 ; 3×3 ; 2 ; 1 ; ReLU Conv2d_4 ; 64 ; 64 ; 2×2 ; 2 ; 1 ; ReLU	Layer_ID ; Num Nodes : In - Out ; A FC_z ; 10 - 144 ; ReLU FC_z_mu ; 144 - 576 ; ReLU
Layer_ID ; Num Nodes : In - Out ; A FC_z ; 576 - 144 ; ReLU FC_z_mu ; 144 - 10 ; None FC_z_logvar ; 144 - 10 ; None	Layer_ID ; I ; O ; K ; S ; P ; A UpConv2d_1 ; 64 ; 64 ; 2×2 ; 2 ; 1 ; ReLU UpConv2d_2 ; 64 ; 32 ; 3×3 ; 2 ; 1 ; ReLU UpConv2d_3 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; ReLU UpConv2d_4 ; 32 ; 1 ; 4×4 ; 2 ; 1 ; Sigmoid

Table B.5: Specification of the β -VAE encoder and decoder architecture parameters, for BlockStacks $128 \times 128 \times 3$ size input data. I: Input channels, O: Output channels, K: Kernel size, S: Stride, P: Padding, A: Activation

Encoder Input: $128 \times 128 \times 3$	Decoder Input: $\mathbb{R}^{10}$
Layer_ID ; I ; O ; K ; S ; P ; A Conv2d_1 ; 3 ; 32 ; 4×4 ; 2 ; 1 ; ReLU Conv2d_2 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; ReLU Conv2d_3 ; 32 ; 64 ; 4×4 ; 2 ; 1 ; ReLU Conv2d_4 ; 32 ; 64 ; 4×4 ; 2 ; 1 ; ReLU Conv2d_5 ; 64 ; 64 ; 4×4 ; 2 ; 1 ; ReLU	Layer_ID ; Num Nodes : In - Out ; A FC_z ; 10 - 256 ; ReLU FC_z_mu ; 256 - 1024 ; ReLU
Layer_ID ; Num Nodes : In - Out ; A FC_z ; 1024 - 256 ; ReLU FC_z_mu ; 256 - 10 ; None FC_z_logvar ; 256 - 10 ; None	Layer_ID ; I ; O ; K ; S ; P ; A UpConv2d_1 ; 64 ; 64 ; 4×4 ; 2 ; 1 ; ReLU UpConv2d_2 ; 64 ; 32 ; 4×4 ; 2 ; 1 ; ReLU UpConv2d_3 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; ReLU UpConv2d_4 ; 32 ; 32 ; 4×4 ; 2 ; 1 ; ReLU UpConv2d_5 ; 32 ; 3 ; 4×4 ; 2 ; 1 ; Sigmoid

C Specification for theory of ordinality

To support the claim that only the `isSuccessor` relation is required in ϕ^{guide} in Chapter 6 due to its logical relationship with the remaining relations, the logical sentences are include here:

$$\begin{aligned}
& \forall A, B, C \text{ (isSuccessor}(A, B) \wedge \text{isSuccessor}(C, B) \rightarrow \text{isEqual}(A, C)) \\
& \quad \forall A, B \text{ (isSuccessor}(A, B) \rightarrow \text{isGreater}(A, B)) \\
& \quad \forall A, B, C \text{ (isSuccessor}(A, B) \wedge \text{isGreater}(B, C) \rightarrow \text{isGreater}(A, C)) \\
& \quad \forall A, B \text{ (isSuccessor}(A, B) \leftrightarrow \text{isPredecessor}(B, A)) \\
& \quad \forall A, B \text{ (isPredecessor}(A, B) \rightarrow \text{isLess}(A, B)) \\
& \quad \forall A, B, C \text{ (isPredecessor}(A, B) \wedge \text{isLess}(B, C) \rightarrow \text{isLess}(A, C)).
\end{aligned}$$

Table B.6: Specification of our the WReN model parameters. A: Activation

Scoring function f_ϕ Input: $28 \times 28 \times N_C$	Relation net g_θ Input: concatenation $[\mathbb{R}^{64}; \mathbb{R}^{64}]$
Layer_ID ; Num Nodes : In - Out ; A FC_1 ; 64 ; 64 ; ReLU FC_2 ; 64 ; 64 ; ReLU – Drop-out layer $p = 0.5$ – FC_4 ; 64 ; 1 ; None	Layer_ID ; Num Nodes : In ; Out ; A FC_1 ; 2*64 ; 128 ; ReLU FC_2 ; 128 ; 128 ; ReLU FC_3 ; 128 ; 128 ; ReLU FC_4 ; 128 ; 64 ; ReLU

Tile-position feature projection h_γ Input: $\mathbb{R}^{10} + \mathbb{R}^9$ position one-hot vector
Layer_ID ; Num Nodes : In ; Out ; A FC_1 ; 10+9 ; 64 ; ReLU

Therefore, by knowing all of the successor relations between data instances, assuming all are connected via at least one true successor case, it should be possible to infer the remaining relationships that they share.

For completeness, the truth-tables for each of the sub-theories that the consistency loss evaluate against are also provided. Only configurations that are valid under the constraints, indicated by $\subset \mathcal{T} = T$ are included, where this notation highlights the fact that each incomplete set of constraints form a subset of the overall theory $\mathcal{T}$.

Firstly, the truth-table that describes consistency-across constraints shared between relation truth-values is given by the following⁵, $\forall A, B$:

$G(A, B)$	$E(A, B)$	$L(A, B)$	$S(A, B)$	$P(A, B)$	$\subset \mathcal{T}$
T	F	F	F	F	T
T	F	F	T	F	T
F	T	F	F	F	T
F	F	T	F	F	T
F	F	T	F	T	T

where the same relation abbreviations are used, as in the main text.

Next, the three consistency-individual (Con-I) truth-tables are provided. These are referred to as being “individual” due to the fact that they describe constraints applied to the truth-state of

⁵Note that each of these truth-tables relies on the $\forall$ to denote that, although variables are shown in the headers, the constraints hold for all possible *assignments* to those variables.

Table C.1: Characteristic properties of the ordinal relations.

Relation	asymmetric	transitive	reflexive
G	Y	Y	N
E	N	Y	Y
L	Y	Y	N
S	Y	N	N
P	Y	N	N

a single relation. For transitivity, given by the rule *e.g.* $G(A, B) \wedge G(B, C) \rightarrow G(A, C)$, we have that $\forall A, B, C$:

$G(A, B)$	$G(B, C)$	$G(A, C)$	$\subset \mathcal{T}$
F	F	F	T
F	F	T	T
T	F	F	T
T	F	T	T
F	T	F	T
F	T	T	T
T	T	T	T

(2)

For asymmetry, where $S(A, B) \rightarrow \neg S(B, A)$, we have $\forall A, B$:

$S(A, B)$	$S(B, A)$	$\subset \mathcal{T}$
F	F	T
T	F	T
F	T	T

(3)

.

Finally, for reflexivity, given by $E(A, A)$ (in this case describing that an object is always equal to itself) we have $\forall A$:

$E(A, A)$	$\subset \mathcal{T}$
T	T

(4)

The truth-table matrices for each of the above truth-tables, required for the consistency loss, can be obtained by replacing T with 1 and F with 0.

D Expanded consistency loss derivation

This section provides the expanded derivation of the $\bar{\epsilon}$ proxy for ϵ -consistency/coherence, as defined in Section 6.3. For notational clarity, in the following $\psi_{\mathcal{S}}$ is omitted, such that $\phi_r(\psi_{\mathcal{S}}(O))$ is abbreviated to $\phi_r(O)$.

In the following, no assumptions about the sizes of domain $\mathcal{S}$, signature σ and arities of each $r \in \sigma$ are made. Further, take $\mathcal{T}$ to be an arbitrary theory over σ consisting of universally quantified formulae, and the validity of each ground instances of atomic formula with respect to $\mathcal{T}$, can be expressed by a single ground truth-table matrix, $\mathbf{T} \in \{0, 1\}^{K_0 \times K_1 \times K_2}$, wherein each slice, $\mathbf{T}_{k::}$ gives a unique grounding of domain objects to the variables, v , required by $\mathcal{T}$. For each grounding of the $K_0 = |\mathcal{S}|^{|v|}$ possible groundings, there are $K_1 = 2^l$ unique truth-assignments to the l atomic formulae that constitute $\mathcal{T}$, giving $K_2 = l + 1$ assignments per $\mathbf{T}_{k,t,:}$ row; one per atomic formulae and an additional value that denotes whether the particular row satisfies $\mathcal{T}$. $\mathbf{T}$ can be obtained by taking a truth-table (such as those provided in Appendix C) and switching true (T) for 1 and false (F) for 0, and producing K_0 copies for each assignment of domain elements to the variables. Given this truth-table matrix, notice that a structure $\mathcal{S}_{\sigma}$ can be composed by selecting a single row of $\mathbf{T}$ for each grounding (k th slice), giving a vector $\mathbf{c}_{kt} = \mathbf{T}_{kt(1:l)}$. If the structure is a model of $\mathcal{T}$, *i.e.* $\mathcal{S}_{\sigma} \in \mathcal{M}_{\mathcal{S}}^{\mathcal{T}}$, then only rows with $\mathbf{T}_{ktK_2} = 1$ are allowed. Taking t^+ to be the set of rows such that $\mathbf{T}_{ktK_2} = 1$ (which is identical for each k) *i.e.* $t^+ = \{ t \mid \mathbf{T}_{ktK_2} = 1, t \in \{1, \dots, K_1\} \}$, we can then rewrite $\Gamma_{\mathcal{M}_{\mathcal{S}}^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}}$ in terms of samples from $\mathbf{T}$:

$$\begin{aligned} \Gamma_{\mathcal{M}_{\mathcal{S}}^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}} &= \sum_{\mathcal{S}_{\sigma} \in \mathcal{M}_{\mathcal{S}}^{\mathcal{T}}} \prod_{r \in \sigma} \prod_{O \in \mathcal{S}^{\text{ar}(r)}} \phi_r(O)^{\gamma_{\mathcal{S}_{\sigma}}^{r,O}} (1 - \phi_r(O))^{1 - \gamma_{\mathcal{S}_{\sigma}}^{r,O}} \quad (\text{Equation 5.4}) \\ &= \sum_{\mathcal{S}_{\sigma} \in \mathcal{M}_{\mathcal{S}}^{\mathcal{T}}} \prod_{k=1}^{K_0} \sum_{t \in t^+} \mathbf{1}_{t_k^{\mathcal{S}_{\sigma}}}(t) \prod_{m=1}^l f(\phi_{r_m}, O_{km}, c_{ktm})^{N(\phi_{r_m}, O_{km}, c_{ktm}, \mathcal{S}_{\sigma}) - 1} \end{aligned} \quad (5)$$

with

$$f(\phi_{r_m}, O_{km}, c_{ktm}) = \phi_{r_m}(O_{km})^{c_{ktm}} (1 - \phi_{r_m}(O_{km}))^{1 - c_{ktm}}. \quad (6)$$

In the above, $\mathbf{1}_{t_k^{\mathcal{S}_{\sigma}}}(t)$ is an indicator function which equals 1 if $t = t_k^{\mathcal{S}_{\sigma}}$ and 0 otherwise, for active row $t_k^{\mathcal{S}_{\sigma}}$ under structure $\mathcal{S}_{\sigma}$ and grounding k . $\mathbf{1}_{t_k^{\mathcal{S}_{\sigma}}}(t)$ has the role of only including the *single* summand where t corresponds with $t_k^{\mathcal{S}_{\sigma}}$. $N(\phi_{r_m}, O_{km}, c_{ktm}, \mathcal{S}_{\sigma})$ is a function that counts

the number of repeat products of term $f(\phi_{r_m}, O_{km}, c_{ktm})$, such that the appropriate root can be applied. r_m is used to denote the relation for atomic formula at column m and O_{km} its corresponding arguments under grounding k ; and c_{ktm} to denote the truth-assignment of the atomic formula for column m , as designated by row t .

At this point, we are left with an expression for $\Gamma_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ in terms of truth-table matrix $\mathbf{T}$ entries, which is more reminiscent of $\mathcal{L}^{\bar{e}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$ as defined in Section 6.3. However, we must go further to expose the relationship between $\Gamma_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ and $\mathcal{L}^{\bar{e}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$ for arbitrary $\mathcal{T}$ expressed by $\mathbf{T}$. We will now show that the consistency loss $\mathcal{L}^{\bar{e}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$ gives the negative log-likelihood of satisfying $\mathcal{T}$ given a grounding $k \in \{1, \dots, K_0\}$, which can be further seen as a relaxation of $\Gamma_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ to sum over all rows $t \in t^+$ and without normalisation via the $N(\phi_{r_m}, O_{km}, c_{ktm}, \mathcal{S}_\sigma)^{-1}$ exponent. With Boolean random variable $B_{\mathcal{T}}$ denoting whether $\mathcal{T}$ is ($B_{\mathcal{T}} = 1$) or is not ($B_{\mathcal{T}} = 0$) satisfied, the consistency loss for a soft-structure $\tilde{\mathcal{S}}_\sigma$ against theory $\mathcal{T}$ is given by,

$$\mathcal{L}^{\bar{e}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma) = \mathbb{E}_{k \sim U[\{1, \dots, K_0\}]} [H(p(B_{\mathcal{T}}|\mathcal{S}_\sigma, k), p(B_{\mathcal{T}}|\tilde{\mathcal{S}}_\sigma, k))] \quad \text{Equation 6.18 base}$$

where $H(\cdot, \cdot)$ gives the *cross-entropy*. This can be expanded to

$$\begin{aligned} \mathcal{L}^{\bar{e}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma) = & -\frac{1}{K_0} \sum_{k=1}^{K_0} p(B_{\mathcal{T}} = 1|\mathcal{S}_\sigma, k) \ln p(B_{\mathcal{T}} = 1|\tilde{\mathcal{S}}_\sigma, k) \\ & + (1 - p(B_{\mathcal{T}} = 1|\mathcal{S}_\sigma, k)) \ln(1 - p(B_{\mathcal{T}} = 1|\tilde{\mathcal{S}}_\sigma, k)). \end{aligned} \quad (7)$$

where $\mathcal{S}_\sigma \in \mathcal{M}_S^T$. Given $\mathcal{S}_\sigma \in \mathcal{M}_S^T$, then $p(B_{\mathcal{T}} = 1|\mathcal{S}_\sigma, k) = 1$ always holds. This means the negative case in Equation 7 can be ignored, yielding the following simplified form:

$$\begin{aligned} \mathcal{L}^{\bar{e}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma) = & -\sum_{k=1}^{K_0} \frac{1}{K_0} \ln p(B_{\mathcal{T}} = 1|\tilde{\mathcal{S}}_\sigma, k) \\ = & -\mathbb{E}_{k \sim U[1, \dots, K_0]} [\ln p(B_{\mathcal{T}} = 1|\tilde{\mathcal{S}}_\sigma, k)]. \end{aligned} \quad \text{Equation 6.18}$$

and so $\mathcal{L}^{\bar{e}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$ is simply the negative log-likelihood of sampling a satisfied theory ($B_{\mathcal{T}} = 1$) from soft-structure $\tilde{\mathcal{S}}_\sigma$, for randomly sampled grounding k . Next, we show the similarities between $\mathcal{L}^{\bar{e}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$ and $\Gamma_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ by looking at the likelihood $p(B_{\mathcal{T}} = 1|\tilde{\mathcal{S}}_\sigma, k)$. First, we define $\bar{\Gamma}_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$

by isolating the likelihood:

$$\begin{aligned} \exp(-\mathcal{L}^{\tilde{\epsilon}}(\mathcal{T}, \tilde{\mathcal{S}}_{\sigma})) &= \prod_{k=1}^{K_0} p(B_{\mathcal{T}} = 1 | \tilde{\mathcal{S}}_{\sigma}, k)^{\frac{1}{K_0}} \\ &\doteq \bar{\Gamma}_{\mathcal{M}_S^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}} \end{aligned} \quad (8)$$

We then expand $p(B_{\mathcal{T}} = 1 | \tilde{\mathcal{S}}_{\sigma}, k)$ to

$$\begin{aligned} p(B_{\mathcal{T}} = 1 | \tilde{\mathcal{S}}_{\sigma}, k) &= \sum_{t=1}^{K_1} p(B_{\mathcal{T}} = 1 | \mathbf{c}_{kt}) p(\mathbf{c}_{kt} | \tilde{\mathcal{S}}_{\sigma}, k) \\ &= \sum_{t \in t^+} p(\mathbf{c}_{kt} | \tilde{\mathcal{S}}_{\sigma}, k) \end{aligned} \quad (9)$$

where t^+ is defined as before. For all other $t \notin t^+$, $p(B_{\mathcal{T}} = 1 | \mathbf{c}_{kt}) = 0$ and so this acts as a filter, yielding:

$$\bar{\Gamma}_{\mathcal{M}_S^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}} = \prod_{k=1}^{K_0} \sum_{t \in t^+} p(\mathbf{c}_{kt} | \tilde{\mathcal{S}}_{\sigma}, k)^{\frac{1}{K_0}}. \quad (10)$$

$p(\mathbf{c}_{kt} | \tilde{\mathcal{S}}_{\sigma}, k)$ is calculated by evaluating the belief of each relation-decoder against the expected truth-assignment as defined by truth-table row $\mathbf{c}_{kt}$:

$$\begin{aligned} p(\mathbf{c}_{kt} | \tilde{\mathcal{S}}_{\sigma}, k) &= \prod_{m=1}^l \phi_{r_m}(O_{km})^{c_{ktm}} (1 - \phi_{r_m}(O_{km}))^{1-c_{ktm}} \\ &= \prod_{m=1}^l f(\phi_{r_m}, O_{km}, c_{ktm}) \end{aligned} \quad (11)$$

where r_m is the relation for atomic formula associated with column m (which is the same for each k slice and t row) and O_{km} is the grounding of this entry for slice k (which is the same across rows). Putting it all back together, we finally have that:

$$\bar{\Gamma}_{\mathcal{M}_S^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}} = \prod_{k=1}^{K_0} \left(\sum_{t \in t^+} \prod_{m=1}^l f(\phi_{r_m}, O_{km}, c_{ktm}) \right)^{\frac{1}{K_0}}, \quad (12)$$

which makes the similarities between $\Gamma_{\mathcal{M}_S^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}}$ and $\bar{\Gamma}_{\mathcal{M}_S^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}}$ clear and exposes their relationship. In particular, for the special case where $|\mathcal{M}_S^{\mathcal{T}}| = 1$, the outer sum for $\Gamma_{\mathcal{M}_S^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}}$ can be removed, and the remaining differences between $\Gamma_{\mathcal{M}_S^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}}$ and $\bar{\Gamma}_{\mathcal{M}_S^{\mathcal{T}}}^{\tilde{\mathcal{S}}_{\sigma}}$ are the sum over t^+ rows and difference in

exponent over $f(\phi_{rm}, O_{km}, c_{ktm})$. For $\Gamma_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ to be maximised, through $p(\mathcal{S}_\sigma|\tilde{\mathcal{S}}_\sigma) \approx 1$, we would find that $\tilde{\mathcal{S}}_\sigma$ maximally supports only the rows associated with $\mathcal{S}_\sigma$ for each k grounding. Notice that $\bar{\Gamma}_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ is again bound to $(0, 1)$ and achieves $\bar{\Gamma}_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma} \approx 1$ when $\Gamma_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma} \approx 1$. We can use the correspondence between $\Gamma_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ and $\bar{\Gamma}_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ to define a practical ϵ -proxy consistency measure as follows. We firstly re-express ϵ -consistency/coherence but for $\bar{\Gamma}_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}$ and a different $\bar{\epsilon}$. We then trace this back to $\mathcal{L}^{\bar{\epsilon}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$ so a bound in terms of the consistency loss can be reported as the overall ϵ -proxy. Together this yields the following:

$$\begin{aligned} \bar{\epsilon} &\geq 1 - \bar{\Gamma}_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma} \\ \ln \frac{1}{1 - \bar{\epsilon}} &\geq -\ln(\bar{\Gamma}_{\mathcal{M}_S^T}^{\tilde{\mathcal{S}}_\sigma}) \\ &\geq \mathcal{L}^{\bar{\epsilon}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma) \end{aligned} \tag{13}$$

and, via the relationship between $\bar{\epsilon}$ and $\mathcal{L}^{\bar{\epsilon}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$, we can use the consistency loss $\mathcal{L}^{\bar{\epsilon}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)$ as a proxy measure for ϵ -consistency/coherence. Concretely, the proxy $\bar{\epsilon}$ can be computed as

$$\bar{\epsilon} \geq 1 - \frac{1}{\exp \mathcal{L}^{\bar{\epsilon}}(\mathcal{T}, \tilde{\mathcal{S}}_\sigma)} \tag{14}$$

and this completes the derivation.