Large Language Models as In-context AI Generators for Quality-Diversity

Bryan Lim[1], Manon Flageat[1], and Antoine Cully[1]

[1]Imperial College London, United Kingdom
bryan.lim16@imperial.ac.uk

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

Abstract

Quality-Diversity (QD) approaches are a promising direction to develop open-ended processes as they can discover archives of high-quality solutions across diverse niches. While already successful in many applications, QD approaches usually rely on combining only one or two solutions to generate new candidate solutions. As observed in open-ended processes such as technological evolution, wisely combining large diversity of these solutions could lead to more innovative solutions and potentially boost the productivity of QD search. In this work, we propose to exploit the pattern-matching capabilities of generative models to enable such efficient solution combinations. We introduce In-context QD, a framework of techniques that aim to elicit the in-context capabilities of pre-trained Large Language Models (LLMs) to generate interesting solutions using the QD archive as context. Applied to a series of common QD domains, In-context QD displays promising results compared to both QD baselines and similar strategies developed for single-objective optimization. Additionally, this result holds across multiple values of parameter sizes and archive population sizes, as well as across domains with distinct characteristics from BBO functions to policy search. Finally, we perform an extensive ablation that highlights the key prompt design considerations that encourage the generation of promising solutions for QD.

Introduction

Open-ended processes are systems that continuously generate novel and interesting outcomes and products. They have tremendous potential as tools for creativity or as invention generators, and could drive the next generation of scientific discovery and lead to major AI innovations (Stanley, 2019). However, designing algorithms and systems which are truly open-ended and capable of never-ending creativity and complexity still remains a challenge (Stanley, 2017). One of the challenges in open-ended search is generating solutions that are novel and better, given knowledge and solutions that already exist.

We take inspiration from how humans invent and innovate and how the creative process of invention often involves combining many diverse concepts together. Humans rely on a diversity of past solutions in the form of other inventions, artifacts and knowledge-bases such as books and code-bases to collectively and creatively generate new inventions. For example, the development and invention of a device like the smartphone was built upon a large diversity of prior generations of inventions and engineering from screens (material science), power supplies (chemistry), processing chips and many more. Provided these building blocks, the ingenuity and creativity of humans to effectively ingest and extract the key elements from this diversity of examples is also critical to making such breakthroughs and discoveries. More often than not, many of the modern technologies follow this same observation and are rarely developed in isolation and from scratch. While this recombination is characteristic of processes such as cross-over in natural evolution, these usually involve combining only two parent solutions. Effectively combining and using a large number of past inventions is not trivial to achieve. Our work looks at replicating this open-ended process of invention and innovation observed in cultural and technical evolution by (i) using foundation models to effectively ingest a large diversity of solutions to generate solutions that are both better and more novel, and (ii) using Quality-Diversity to maintain and provide these models with many diverse and high-quality examples as context for generation.

Quality-Diversity (QD) (Pugh et al., 2016; Cully and Demiris, 2017; Chatzilygeroudis et al., 2021) algorithms are an effective way to collect and maintain a diversity of the best solutions seen and has been a step towards open-ended algorithms. Instead of just searching for a single optimal solution in conventional optimization, QD searches for a diversity of high-performing solutions. A QD approach is useful when one wants a diversity of good options and solutions at the end of search, but can also be implicitly effective in the conventional optimization setting by providing stepping stones to a more optimal solution (Gaier et al., 2019; Clune, 2019), that otherwise might have been discarded. This has benefits across many applications such as robotics (Cully et al., 2015), procedural content generation (Gravina et al., 2019), engineering design (Gaier et al., 2018) and many more.

In QD, new solutions are commonly generated either through random mutations and variations, or optimization

algorithms such as evolution strategies (ES) or gradient descent (i.e. policy gradients). These generation methods use the existing archive of elites mainly as starting points, either as parents for mutations, or for initialization of optimization procedures such as gradient descent. Existing QD approaches rarely consider using the entire archive (or even multiple solutions from the archive) to generate new solutions, while having this additional information could have great benefits and be more effective for the generation process.

We propose to address this challenge by using large transformer models such as Large Language Models (LLMs). This is possible due to the in-context learning capabilities of large transformer models (Brown et al., 2020), where given prompts of arbitrary examples and demonstrations of a task, these models can extract patterns effectively to generate completions and even improvements (Brown et al., 2020; Laskin et al., 2022; Mirchandani et al., 2023). This work introduces in-context AI generators for QD (In-context QD), where new solutions are generated using an LLM by passing a representative subset of the archive of elites into the context of the model. Instead of just starting points, we use the in-context capabilities of pre-trained LLMs to ingest a larger diversity of high-performing solutions from the archive to improve the generation of solutions by leveraging both the pattern-matching and generative capabilities of LLMs. Specifically, we introduce few-shot prompting strategies for QD by providing solutions from the archive of elites as examples in the context when prompting the LLM. QD lends itself nicely to in-context learning by providing a diverse set of high-quality and diverse examples for the model to potentially generate more interesting ideas and solutions.

We show that by careful construction of template, context and queries of the prompt, In-context QD can effectively generate novel and high-quality solutions for QD search over a range of parameter search space dimensions, and archive sizes. We rigorously demonstrate this over a range of QD benchmarks with ranging characteristics such as benchmark BBOB functions, redundant robotic arm control and parameterized policy search for a robot. In most cases, In-context QD performs better than MAP-Elites across metrics especially in finding regions of high-fitness. Additionally, we conduct a thorough study over components of In-context QD and highlight the importance of the choice of context template, context structure and context size.

Background Related Work

Quality-Diversity and MAP-Elites

Drawing from natural evolution, Quality-Diversity (QD) optimization actively seeks a variety of solutions to a given problem, akin to species with diverse characteristics achieving the same objective in the natural world. In other words, instead of looking for one objective-maximizing solution, QD searches for a collection of diverse and high-performing solutions. To do so, most QD approaches rely on a fitness f that quantify the quality of solutions and a set of pre-defined characteristics that allow quantifying their diversity, referred to as features $\mathbf{d}$.

Among existing QD approaches, the most commonly used one is MAP-Elites (Mouret and Clune, 2015), which is the focus of this work. MAP-Elites maintains an archive of solutions that obtains its structure by splitting the feature space into a multidimensional archive of C cells. Each cell of the archive defines a feature-space neighborhood (i.e. niche) where one solution can be kept. The solutions kept in the cells are referred to as elites and the set of all elites is returned as the final collection of solutions of the algorithm. The use of an archive guarantees a good coverage of the feature space and thus a large diversity of the final elites according to the feature dimensions. After initialization with random solutions, a MAP-Elites loop follows the standard cycle of selection, mutation, evaluation and addition to the archive. The selection of parents is done by uniformly sampling from the existing elites of the archive, to encourage quality and diversity. The addition to the archive follows two rules: (1) if the new solution fills in an empty cell, it is always added to that cell, and (2) if there is already an elite in the cell, the new solution only replaces it if it has a higher fitness f. These conditions ensure that added solutions either improve the diversity or quality of the archive.

Variations and Optimizers. In MAP-Elites, the mutation can be done using any common mutation operator such as standard GA (Mouret and Clune, 2015) or specific operators tailored for QD (Vassiliades and Mouret, 2018). Previous work has also proposed to use gradient-informed mutation when available (Fontaine and Nikolaidis, 2021; Nilsson and Cully, 2021; Pierrot et al., 2022), or mutation based on evolution strategies (Colas et al., 2020; Fontaine et al., 2020; Flageat Lim and Cully, 2023). In comparison, our work leverages the pattern completion abilities of LLM to generate the next solution given the large diversity of the archive of elites. While LLMs have also been used as mutation operators for their ability to tackle text-based and program-based QD domains(Lehman et al., 2023; Bradley et al., 2023), we focus and demonstrate these capabilities on general QD BBO optimization problems, other than text or code and develop prompt construction strategies specifically for QD. Also close to our work, BOP-Elites (Kent and Branke, 2020) and DDE-Elites (Gaier et al., 2020) both learn a representation of the current archive content and use this representation as a mutation operator to generate new relevant offspring. They rely on a Gaussian Process and a Variational Autoencoder model respectively. While our work also uses a representation of the archive content to generate relevant offspring, it relies on the in-context capabilities of pre-trained foundation models and does not require training a model online during the optimization. We show in the following that the LLM manages to generate relevant offspring without requiring any online specific training.

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

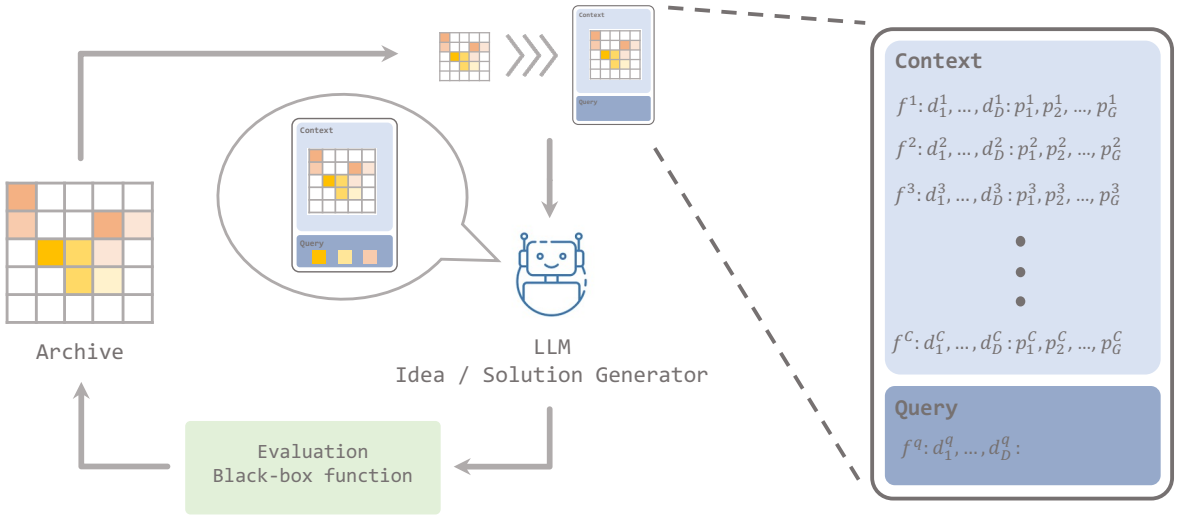


Figure 1: Overview of In-context QD using LLMs as idea and solution generators. The LLM observes the current state of the archive which contains high-quality and diverse solution examples and uses this to generate new solutions to be evaluated and considered for addition to the archive. The prompt consists of *context* and *query* portions which are concatenated together.

Model-based QD. Model-based QD are a family of QD methods that commonly use learnt models as surrogate models to increase the sample efficiency of search (Gaier et al., 2018; Keller et al., 2020; Lim et al., 2022; Zhang et al., 2023) or in domains in which model outputs are good proxies (Bradley et al., 2023; Ding et al., 2023). LLMs are also models, but instead of using these models for their predictive capabilities, our approach uses them for their generative capabilities.

Language Models as Optimizers

Inspired by evolutionary computation, LLMs have been very recently used as mutation or variation operators, especially in text-based domains, ranging from modifying sentence sentiment to generating mathematical expressions, code and programs (Lehman et al., 2023; Meyerson et al., 2023). In fact, when searching and optimizing in the space of text and prompts, using evolution-based methods is one of the more popular and effective approaches (Yang et al., 2023; Fernando et al., 2023) as they are black-box optimization algorithms. Additionally, LLMs can act as optimizers and work over sequences of integers (Mirchandani et al., 2023; Yang et al., 2023), as well as problems that can be represented as code or programs (Chen et al., 2024; Romera-Paredes et al., 2024; Trinh et al., 2024).

Prior work using LLMs as optimizers has mainly focused on either (i) conventional optimization problems where the goal is to find a single solution instead of a QD problem (Yang et al., 2023; Mirchandani et al., 2023; Meyerson et al., 2023; Zhang et al., 2023), or (ii) focuses on the QD problem but uses the LLM in a way that does not explicitly account for both the fitness and feature in the prompt and has mainly been used in subjective text-based domains (Bradley et al., 2023; Samvelyan et al., 2024). There is preliminary work which uses LLMs for more generic evolutionary optimization (Liu et al., 2023) but considers the case of conventional optimization and requires very detailed task information and descriptions. Closest to our approach is concurrently released work which also studies prompting strategies but targeting recombination operators for Evolution Strategies (ES) (Lange et al., 2024), where the LLM generates the next mean statistic of an ES which solutions are sampled from. Instead, our work considers the general QD problem which contains the added dimensions of diversity. Additionally, our approach taps into the curated solutions in the archive of elites to build the context and studies in-context prompt construction approaches for the QD problem and directly generates sample solutions.

Method

In QD, the goal is to find a diversity of high-quality solutions. This provides two dimensions of optimization, fitness and novelty/diversity, compared to conventional single-objective optimization approaches which only consider fitness. A common approach when using LLMs as optimizers is to sort the sequence in an increasing order of fitness and provide a query of higher value, allowing the model to complete the sequence (Yang et al., 2023; Mirchandani et al., 2023). However, in QD, we are not concerned with just finding a solution that has a high fitness, we are also interested in covering the feature space with high-scoring solutions. Figure 1 shows an overview of our approach to this, In-context QD. The LLM takes as input the current state of the archive, and uses this as a prompt to generate new solutions. The generated solutions are then considered for addition to grow the quality and diversity of the archive as in MAP-Elites. In the following, we

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

explain how we represent the solutions and build a prompt that consists of the *context* and the *query*. This provides a general and modular framework for which to use LLMs as AI generators for Quality-Diversity.

Prompt Template

To include both fitness and all the dimensions of features, we use a prompt template that takes the form:

$$f^i : d_1^i, ..., d_G^i : p_1^i, ..., p_D^i \quad (1)$$

where i represents the index of a solution, G represent the total dimensions of the features and D represent the dimensions of the parameter space. The fitness f, features **d** and parameters **p** are separated by a colon ":" while each value is separated by a comma ",". Each solution i is then separated by a line break "$\backslash n$". This representation allows us to consider all elements of the QD problem. We follow prior work (Mirchandani et al., 2023; Lange et al., 2024) and represent the features **d** and parameters **p** as integers. This is due to how numbers/digits are tokenized in LLMs, where representing a large combination of digits stacked together as a float/decimal might be tokenized, broken up and ingested in a non-uniform and confusing manner for the model (Millidge, 2024; Singh and Strouse, 2024).

Building Context

The solutions maintained in a QD archive lend themselves nicely to few-shot and many-shot prompting of LLMs as the diversity and quality of solutions provide a good set of examples and context to refer to in order to generate better ideas and solutions. Ideally, the entire archive could be used as the context. However, given the context size limit of many open-source LLMs, we use a sample/subset of solutions of the archive as an approximation of the archive to construct the context. If provided with models with larger context or problems with lower archive sizes, the entire archive could be used as the context. The context size is a hyper-parameter and can take values up to the archive size C. Yet, despite this temporary limitation, we can already demonstrate the effectiveness of this proposed approach.

In building the context, there are two considerations: the ***content*** and the ***structure*** of the context. As explained above, the ***content*** of the context is a subset representation of the archive. We uniformly sample solutions from the archive to form the content of the context. An increasingly larger context of solutions sampled would give us an increasingly better representation of the archive culminating in the entire archive itself. Meyerson et al. (2023) suggest that few-shot prompting can be viewed as Estimation of Distribution Algorithms (EDAs). In our QD setting, by providing more solutions from the archive in the context using our prompt template, we provide a good approximation of the patterns and the distributions of the fitness-feature space. This provides a better context for the language model especially when queried for desired features.

Configuration	Options
Prompt Template	(LMX, Fitness, Feature, QD)
Context Size	[0, C]
Context Structure	(Sort Fit., Sort Cell Dist., Random)
Fit. Query Strategy	Context Improvement
Desc. Query Strategy	(Empty, Uniform)

Table 1: Variants of context building and query strategies.

Next, we look at the ***structure*** of the context. As observed in prior work (Mirchandani et al., 2023; Yang et al., 2023) and also demonstrated in our work, the structure of the context also matters to provide a clearer heuristic and pattern to the model. There are many possible strategies for this. We introduce sorting according to distance from the query features (see next section) where solutions in the context are ranked from the furthest to the closest to provide a heuristic in the direction of the features. Alternatively, we can also opt for the tried and tested strategy of sorting the context based on the fitness scores. However, this might not always be effective in all QD domains as the search for diverse solutions is a big component of QD. Other options could include sorting the solutions randomly, or according to the certain dimensions of the features, or to other scores of interest. Through the query strategy, we will see why this makes sense as an overall approach to QD.

Query Strategy

To get the language model to generate a solution that will improve the quality and diversity of our current archive, we take the "just ask" (Jang, 2021) approach. We will "ask" the LLM for quality and diversity, or in particular, solutions that are novel w.r.t to the archive and are high-quality. To do this, we append a query fitness f^q and features $d_1^q, ..., d_G^q$ to the context prompt (see Figure 1) to elicit a pattern or sequence completion behavior from the model, based on the context.

We pick a fitness query f_q approach that complements and fits the context, which is a sampled representation of the QD archive in the style of the prompt template. We set a fitness query f_q that is incrementally larger ($\sim 20\%$) than the best fitness solution in the context. This fitness query strategy is referred to as context improvement. Such a query always ensures a reasonable increase in fitness performance given the context sequence to ensure a systematic trend and pattern in the prompt.

The selection of features used in the query d^q cannot be treated entirely the same as with the fitness. In the case of features, the goal is to cover the entire feature space and not find an optimal point in the space by moving in a specific direction of a feature. While expanding the horizon and boundary of the archive is needed, it is also just as important to fill in "holes" in the archive to maximize the coverage

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

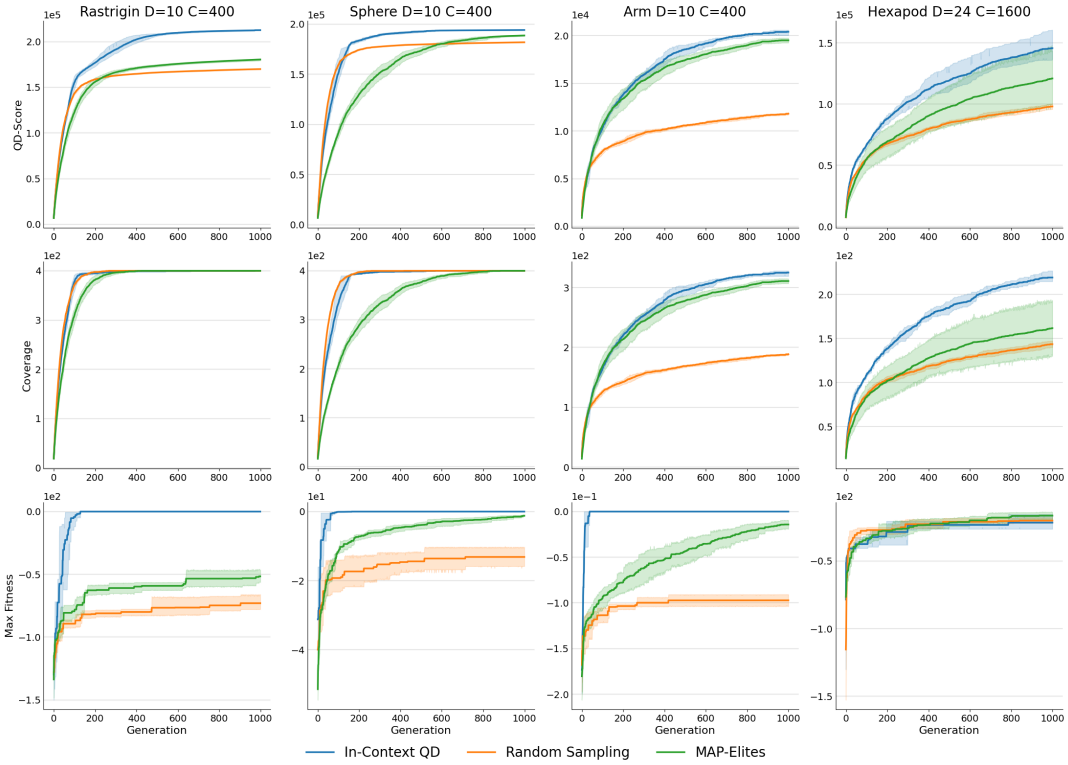


Figure 2: Performance comparison over QD-Score, Coverage and Max-Fitness across all tasks considered where D is the parameter space dimensions and C is the number of niches in the archive. The results are averaged over 5 independent runs.

of the feature space. An unbiased approach to this is to uniformly sample the centroids of the cells in the archive to obtain query features $d_1^q, ..., d_G^q$. We refer to this approach as "Uniform". We also propose a strategy which queries features from cells that are empty first, after which we default to sampling features uniformly from the cells in the archive when the number of empty cells becomes less than the batch size. The centroids of the empty cells and sampled cells are used for the queries. We refer to this strategy as "Empty".

Overall, this proposed context-building approach is modular, composable and easily amendable to new templates, context-building or query strategies. Table 1 summarizes the different options available when applying In-context QD.

Experimental Setup

We evaluate our approach on a range of benchmark QD optimization tasks. As In-context QD relies heavily on pattern and sequence matching and completion, we highlight characteristics of how the parameters are related to the fitness and feature in each task. We first evaluate on two popular functions from the Black-Box Optimization Benchmark (BBOB) (Hansen et al., 2010): the **Rastrigin** (Rosenbrock function) and **Sphere** tasks as done commonly in QD literature (Mouret and Clune, 2015; Fontaine et al., 2020). The features in these synthetic QD tasks are directly tied to the parameter space making exploration of the feature space relatively simple, where random uniform sampling over the parameter space can result in high coverage. Next, we consider the redundant planar robotic **Arm** task (Mouret and Clune, 2015; Cully et al., 2015) where the parameters represent the angles of each joint in the system. Here, the features are the final end-effector position, which is less trivial and cannot be directly inferred from the parameters. For these three tasks, we evaluate our approach across different parameters space dimensions $D \in \{5, 10\}$. Finally, we also consider a policy search task where search is over $D = 24$ parameters, of a locomotion controller for a **Hexapod** robot (Cully and Mouret, 2013; Chatzilygeroudis et al., 2018).

We compare in-context QD against a Random Sampling baseline which samples the parameter space uniformly and MAP-Elites (Mouret and Clune, 2015) with the iso-line variation operator (Vassiliades and Mouret, 2018). We evaluate the (i) maximum fitness score found throughout search, (ii) the coverage, measured by the number of filled cells in the archive and (iii) the QD score which is the sum of fitness of all the solutions in the archive. For all our algorithms and baselines, we use a batch size of 10 at each generation, and run for 1000 generations. We also investigate the performance across different feature space sizes $C \in \{400, 1600\}$ by changing the resolution of the MAP-Elites archive $C \in \{20^2, 40^2\}$ to ensure the generality of our approach. To be able to run in-context QD on just a single GPU device, we use Mistral-

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

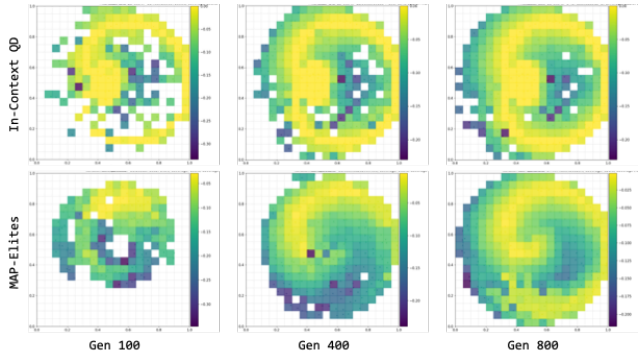


Figure 3: Evolution of archives across generations showing the different optimization paths and dynamics of In-context QD which can quickly exploit patterns for regions of high-performance.

7B-v0.1 (Jiang et al., 2023), an open-source 7B parameter language model.

Results

Figure 2 shows that In-context QD is capable of performing QD optimization and outperforms both Random Sampling and MAP-Elites baselines on the synthetic BBOB tasks. This is interesting as both Random Sampling and MAP-Elites have shown to be very effective in low-dimensional optimization problems despite their simplicity, especially on these synthetics tasks (Mouret and Clune, 2015). We can observe the effectiveness of In-context QD in finding the global optima in these QD problems, quickly extracting the patterns that lead to high fitness scores. This result suggests that In-context QD maintain the ability to find the best solution in the conventional optimization settings as observed in other work (Yang et al., 2023; Mirchandani et al., 2023; Lange et al., 2024). More interestingly, our QD prompting strategy allows the LLM to also effectively extract patterns in the feature distribution to generate more solutions that are diverse and cover the feature space more efficiently. In these BBOB functions, the coverage performance across all algorithms is rather similar. This can be attributed to the ease of covering the feature space on these synthetic BBOB task where the feature dimensions are directly linked to the parameter dimensions. This makes Random Sampling and MAP-Elites strong upper baselines on these tasks.

The Arm task poses another challenge as there is an unclear pattern connecting features and parameters, and as some regions of the feature space are unreachable. Despite these challenges, In-context QD competitively performs QD optimization on this task (third column Figure 2). In the Arm, the solutions with high fitness (i.e. solutions with low variance in joint angles) have a clear pattern in which similar parameter values would give high fitness. In-context QD very quickly identifies and extracts this pattern from the provided archive context and finds the region of high-performing solutions on this task (see third row on Figure 2 Max Fitness). In terms of coverage of the feature space, In-context QD manages to reach performance similar to MAP-Elites, showing its ability to discover the non-trivial pattern connecting parameters to features in this task. We also find that, as In-context QD uses the entire archive and generates solutions based on pattern matching, its optimization dynamics and paths are very different from those of conventional QD algorithms. This interesting observation is most evident in this Arm task as shown in Figure 3.

Lastly, we demonstrate that In-context QD is also an effective QD approach on the Hexapod (policy search) task, where both fitness and features are not in any direct way translatable from the parameter space. This task also consists of $D = 24$ parameters which is more than double the size of dimensions of previous tasks showing that In-context QD can also be effective at higher dimensions. Overall, In-context QD provides an interesting perspective and promising approach for the effective generation of solutions in QD by considering an AI generator that can utilize the knowledge and context of the entire state of the archive.

In addition to the variety of different QD tasks and domains, the results presented in Figure 4 show that In-context QD is a general QD approach and is also robust across both archive sizes $C \in \{400, 1600\}$ and parameter search space dimensions $D \in \{5, 10\}$, performing competitively against compared baselines.

Effect of prompt template

We study the importance of the prompt template used for In-context QD by comparing to existing methods and baselines for this as below:

- **LMX** (Meyerson et al., 2023) which prompt contains only the parameters of each solution.
- **Fitness** (Yang et al., 2023; Mirchandani et al., 2023; Lange et al., 2024) which prompt contains the fitness score followed by the parameter. This template is generally used in conventional single-objective optimization.
- **Feature** which prompt consists of feature characterizations before the parameter vector.
- **QD** which prompt consists of both the fitness scores and the feature as explained in the methods.

Table 2 provides examples of the templates used for each of the strategies. For all of the above methods in our study, we use the same context-building and query strategies. However, for LMX, the query does not affect or contribute to the prompt as it does not rely on queries.

Figure 5 shows the QD-score result of this comparison where we see the importance of both the feature and descriptor in the template of the prompt. As expected, while the fitness-only template does well at finding the best solution, it has no incentive to generate solutions with new descriptors, leading to low coverage and overall poor QD performance. The score and label agnostic approach of LMX does better

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

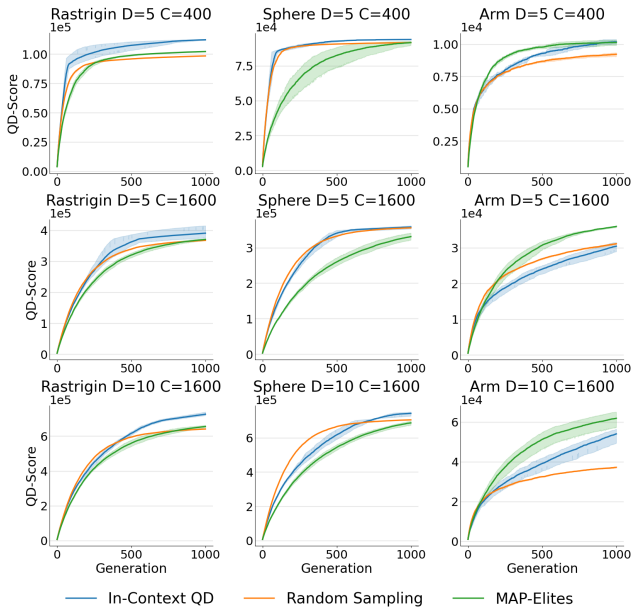


Figure 4: QD-Score of In-context QD across a variety of parameter space dimensionality D and archive sizes C.

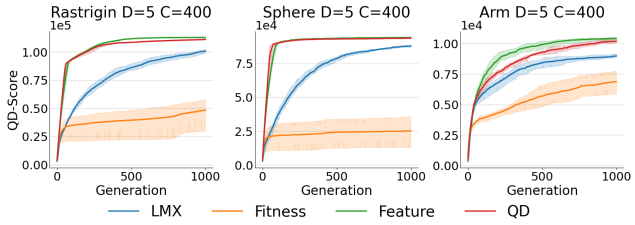


Figure 5: Performance comparison between different prompt templates (see Table 2).

as it is not biased by any values, relying on its context to estimate a distribution from the parameter space and generating solutions from this distribution (Meyerson et al., 2023). In the context of QD and our experiments, since the context consists of solutions and elites from the archive, this distribution is most likely that of the hyper-volume of elites (Mouret and Clune, 2015; Vassiliades and Mouret, 2018). Most importantly, we observe that the presence of the features **d** in the template is critical to the performance of In-context QD. Along with the querying strategy, templates with features **d** in the prompt significantly do better on QD problems.

Effect of context structure

Next, we also consider how the structure of the context affects the performance of In-context QD. In Figure 6, we can observe sorting according to the cell distance from the query feature performs most consistently well across different tasks. This structure complements the feature **d** in the template providing a clear trend and pattern which enables good exploration. Interestingly, we find that the conventional sorting by fitness approach struggles in some domains especially when there is a clear pattern for high-performing solutions such as

Strategy	Template
LMX	$p_1^i, ..., p_D^i$
Fitness	$f^i : p_1^i, ..., p_D^i$
Feature	$d_1^i, ..., d_G^i : p_1^i, ..., p_D^i$
QD	$f^i : d_1^i, ..., d_G^i : p_1^i, ..., p_D^i$

Table 2: Comparison of prompt templates where f is the fitness score, **d** are the features and **p** are the parameters of the solution.

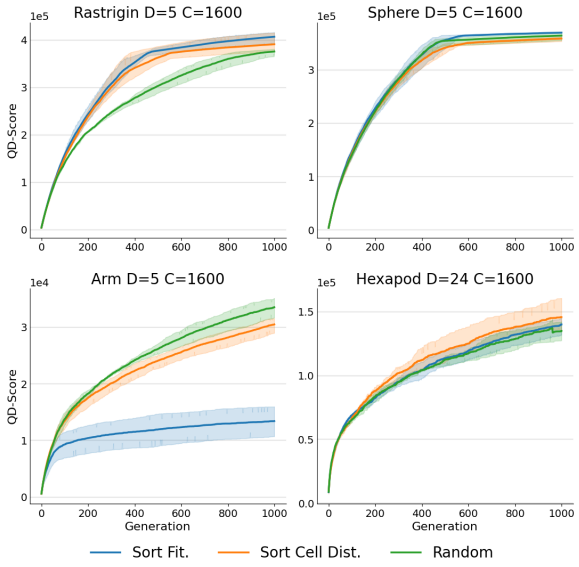


Figure 6: Effect of using different methods to sort and arrange the context structure on the QD-score performance of In-context QD.

in the Arm task. Such a structure causes the LLM to exploit the patterns of high fitness resulting in no exploration of the feature space.

Effect of context size

A critical component of our In-context QD is the use of the entire archive for the generation of the next solutions. Ideally, we would like to provide AI generators with the entire diversity of the archive as context to be able to more effectively understand the distribution of the space of solutions and best generate novel and better-performing solutions w.r.t the current archive. However, as explained in the methods, the context length of current LLMs limits the number of solutions that can fit into the context size. Here, we investigate the effect of the different context sizes on In-context QD when sampling is done to approximate the entire archive of elites. Figure 7 shows that increasing the context size always helps to improve performance. This is particularly important when the archive has a large number of niches. We can observe in Figure 7 that the difference in performance between a context size of 10 and 20 is more significant when $C = 1600$ compared to when $C = 400$. At higher archive sizes, coverage is

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

more difficult, hence getting a better distribution estimate of the feature space through more examples is beneficial. This observation corroborates the hypothesis that sampling from the archive provides an approximation of the archive of elites, and we can get increased performance as this approximation gets better. However, when sampling, these results indicate that there is a minimum context size that is critical to good performance and dependent on the maximum archive size C.

Discussion and Conclusion

In this work, we introduce In-context QD which combines the large diversity of the QD archive to more effectively generate new solutions for QD by leveraging the in-context learning capabilities of pre-trained LLMs. We demonstrate that this allows for strong and competitive performance on a variety of QD tasks ranging from BBO to policy search, while also showing its robustness and generality across a range of archive sizes and parameter space dimensions. Finally, we highlight and show that the choice of the different prompt components such as the prompt template, context structure, context size, and query strategy is critical to using LLMs for QD.

In our work, we utilize LLMs for what these models thrive at and were trained to do, which are their capabilities as general pattern matching machines (Brown et al., 2020; Mirchandani et al., 2023) to spot patterns in the sequences and potentially complete and improve them. Using them in this manner allows us to treat them as idea or solution generators that rely on examples that are high-quality and diverse provided through the library of solutions from QD. Using them in this optimization loop also means that these "ideas" can be "verified" when applied and tested on the task or problem. This is in contrast to many other applications where they are attempted to be used as sources of truth for factual answers, where they behave more like approximate retrieval machines which can at times lead to what is commonly referred to as "hallucinations". In our setting, we rely on these "hallucinations" to dream and generate novel and better ideas or solutions given a set of examples, for which we can then test and experiment on the problem, similar to an engineer, inventor or scientist.

While we focus on the general pattern-matching abilities of LLMs for general optimization problems, they are commonly used for its abilities on mediums such as language or code that it possesses from its training data. The strengths of these models in these domain-specific capabilities are complementary to our proposed approach and we hope to explore these applications. This further demonstrates the versatility of pattern-matching based LLM optimization on a wide set of settings. However, it is important to note that despite the ease and versatility of In-context QD and its rapid exploitation of patterns, it can also struggle where patterns in the fitness, features and descriptors are unclear. Our work highlights generating solutions using in-context learning with LLMs

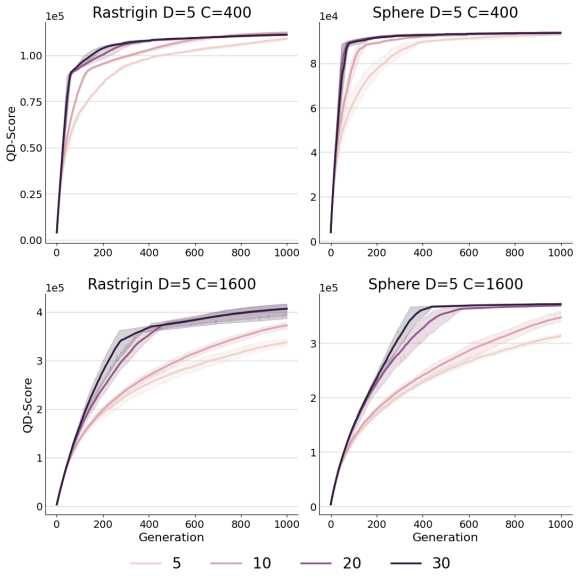


Figure 7: QD-Score performance over a range of context sizes of In-context QD. We find that increasing context size helps performance.

as just an alternative form of optimization based on noisy pattern-matching rather than gradients or random un-directed mutations that can be useful in certain settings, and not something that will dominate and eat up all QD approaches. A more complete approach points to perhaps using In-context QD to complement a multi-emitter setting (Cully, 2021), where the LLM generator can be an emitter which exploits patterns in problems as it arises.

Limitations and Future Work. A limitation of our current approach is that we assume a defined feature space where we can query desired features. This is true and a controllable aspect of many QD algorithms and applications. However, to extend this to a fully open-ended setting where we get continuously evolving and new features, an interesting option would be to get the LLMs themselves to output the query features especially in environments and tasks that are fully open-ended that don't have a specified feature space. We hope to explore this in future work.

Additionally, we are also still limited to optimization problems of smaller parameter dimensions due to the context size of these models. Solutions to these could include models that have increasingly large context-lengths up to a million tokens (Reid et al., 2024) or dimension batched querying (Lange et al., 2024). Yet, we believe the general principles of this approach apply and extend to QD with context-based generative models as solution generators. Potential future work also includes other query and context building strategies due to its simplicity or using LLM-based single-objective optimization (Yang et al., 2023; Mirchandani et al., 2023; Lange et al., 2024) with novelty objectives or curiosity objectives as in novelty search (Lehman and Stanley, 2011; Conti et al., 2018).

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

References

Bradley, H., Dai, A., Teufel, H., Zhang, J., Oostermeijer, K., Bellagente, M., Clune, J., Stanley, K., Schott, G., and Lehman, J. (2023). Quality-diversity through ai feedback. *arXiv preprint arXiv:2310.13032*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chatzilygeroudis, K., Cully, A., Vassiliades, V., and Mouret, J.-B. (2021). Quality-diversity optimization: a novel branch of stochastic optimization. In *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*, pages 109–135. Springer.

Chatzilygeroudis, K., Vassiliades, V., and Mouret, J.-B. (2018). Reset-free trial-and-error learning for robot damage recovery. *Robotics and Autonomous Systems*, 100:236–250.

Chen, A., Dohan, D., and So, D. (2024). Evoprompting: Language models for code-level neural architecture search. *Advances in Neural Information Processing Systems*, 36.

Clune, J. (2019). Ai-gas: Ai-generating algorithms, an alternate paradigm for producing general artificial intelligence. *arXiv preprint arXiv:1905.10985*.

Colas, C., Madhavan, V., Huizinga, J., and Clune, J. (2020). Scaling map-elites to deep neuroevolution. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 67–75.

Conti, E., Madhavan, V., Petroski Such, F., Lehman, J., Stanley, K., and Clune, J. (2018). Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in neural information processing systems*, 31.

Cully, A. (2021). Multi-emitter map-elites: improving quality, diversity and data efficiency with heterogeneous sets of emitters. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 84–92.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.

Cully, A. and Demiris, Y. (2017). Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259.

Cully, A. and Mouret, J.-B. (2013). Behavioral repertoire learning in robotics. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 175–182.

Ding, L., Zhang, J., Clune, J., Spector, L., and Lehman, J. (2023). Quality diversity through human feedback. *arXiv preprint arXiv:2310.12103*.

Fernando, C., Banarse, D., Michalewski, H., Osindero, S., and Rocktäschel, T. (2023). Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.

Flageat Lim, Manon, B. and Cully, A. (2023). Multiple hands make light work: Enhancing quality and diversity using map-elites with multiple parallel evolution strategies. *arXiv preprint arXiv:2303.06137*.

Fontaine, M. and Nikolaidis, S. (2021). Differentiable quality diversity. *Advances in Neural Information Processing Systems*, 34.

Fontaine, M. C., Togelius, J., Nikolaidis, S., and Hoover, A. K. (2020). Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 94–102.

Gaier, A., Asteroth, A., and Mouret, J.-B. (2018). Data-efficient design exploration through surrogate-assisted illumination. *Evolutionary computation*, 26(3):381–410.

Gaier, A., Asteroth, A., and Mouret, J.-B. (2019). Are quality diversity algorithms better at generating stepping stones than objective-based search? In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 115–116.

Gaier, A., Asteroth, A., and Mouret, J.-B. (2020). Discovering representations for black-box optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 103–111.

Gravina, D., Khalifa, A., Liapis, A., Togelius, J., and Yannakakis, G. N. (2019). Procedural content generation through quality diversity. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE.

Hansen, N., Auger, A., Finck, S., and Ros, R. (2010). Real-Parameter Black-Box Optimization Benchmarking 2010: Experimental Setup. Research Report RR-7215, INRIA.

Jang, E. (2021). Just ask for generalization. *evjang.com*.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Keller, L., Tanneberg, D., Stark, S., and Peters, J. (2020). Model-based quality-diversity search for efficient robot learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9675–9680. IEEE.

Kent, P. and Branke, J. (2020). Bop-elites, a bayesian optimisation algorithm for quality-diversity search. *arXiv preprint arXiv:2005.04320*.

Lange, R. T., Tian, Y., and Tang, Y. (2024). Large language models as evolution strategies. *arXiv preprint arXiv:2402.18381*.

Laskin, M., Wang, L., Oh, J., Parisotto, E., Spencer, S., Steigerwald, R., Strouse, D., Hansen, S. S., Filos, A., Brooks, E., et al. (2022). In-context reinforcement learning with algorithm distillation. In *The Eleventh International Conference on Learning Representations*.

Lehman, J., Gordon, J., Jain, S., Ndousse, K., Yeh, C., and Stanley, K. O. (2023). Evolution through large models. In *Handbook of Evolutionary Machine Learning*, pages 331–366. Springer.

Lehman, J. and Stanley, K. O. (2011). Novelty search and the problem with objectives. In *Genetic programming theory and practice IX*, pages 37–56. Springer.

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024

Lim, B., Grillotti, L., Bernasconi, L., and Cully, A. (2022). Dynamics-aware quality-diversity for efficient learning of skill repertoires. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5360–5366. IEEE.

Liu, S., Chen, C., Qu, X., Tang, K., and Ong, Y.-S. (2023). Large language models as evolutionary optimizers. *arXiv preprint arXiv:2310.19046*.

Meyerson, E., Nelson, M. J., Bradley, H., Moradi, A., Hoover, A. K., and Lehman, J. (2023). Language model crossover: Variation through few-shot prompting. *arXiv preprint arXiv:2302.12170*.

Millidge, B. (2024). Integer tokenization is insane.

Mirchandani, S., Xia, F., Florence, P., Ichter, B., Driess, D., Arenas, M. G., Rao, K., Sadigh, D., and Zeng, A. (2023). Large language models as general pattern machines. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Nilsson, O. and Cully, A. (2021). Policy gradient assisted map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 866–875.

Pierrot, T., Macé, V., Chalumeau, F., Flajolet, A., Cideron, G., Beguir, K., Cully, A., Sigaud, O., and Perrin-Gilbert, N. (2022). Diversity policy gradient for sample efficient quality-diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1075–1083.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40.

Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.-b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., Ruiz, F. J., Ellenberg, J. S., Wang, P., Fawzi, O., et al. (2024). Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Samvelyan, M., Raparthy, S. C., Lupu, A., Hambro, E., Markosyan, A. H., Bhatt, M., Mao, Y., Jiang, M., Parker-Holder, J., Foerster, J., et al. (2024). Rainbow teaming: Open-ended generation of diverse adversarial prompts. *arXiv preprint arXiv:2402.16822*.

Singh, A. K. and Strouse, D. (2024). Tokenization counts: the impact of tokenization on arithmetic in frontier llms. *arXiv preprint arXiv:2402.14903*.

Stanley, K. O. (2017). Open-endedness: The last grand challenge you've never heard of. https://www.oreilly.com/radar/open-endedness-the-last-grand-challenge-youve-never-heard-of/.

Stanley, K. O. (2019). Why Open-Endedness Matters. *Artificial Life*, 25(3):232–235.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. (2024). Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Vassiliades, V. and Mouret, J.-B. (2018). Discovering the Elite Hypervolume by Leveraging Interspecies Correlation. In *GECCO 2018 - Genetic and Evolutionary Computation Conference*, Kyoto, Japan.

Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., and Chen, X. (2023). Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Zhang, M. R., Desai, N., Bae, J., Lorraine, J., and Ba, J. (2023). Using large language models for hyperparameter optimization. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/isal2024/36/48/2461219/isal_a_00771.pdf by Imperial College London user on 11 November 2024