DÉJÀQ: Open-Ended Evolution of Diverse, Learnable and Verifiable Problems

Willem Röpke*
AI Lab, Vrije Universiteit Brussel
Belgium
willem.ropke@vub.be

Samuel Coward *
FLAIR, University of Oxford
United Kingdom
scoward@robots.ox.ac.uk

Andrei Lupu
FLAIR, University of Oxford
United Kingdom

Thomas Foster
FLAIR, University of Oxford
United Kingdom

Tim Rocktäschel
University College London
United Kingdom

Jakob Foerster
FLAIR, University of Oxford
United Kingdom

Abstract

Recent advances in reasoning models have yielded impressive results in mathematics and coding. However, most methods depend on static datasets, which promote memorisation and hinder generalisation. We introduce DÉJÀQ, a framework that departs from this paradigm by jointly evolving a diverse set of synthetic mathematical problems alongside model training. This dataset continuously adapts to the model's abilities, generating problems at an appropriate level of difficulty throughout training. We find that models at the 7B scale increasingly generate novel and interesting problems, while smaller models (1.5B-3B) struggle to sustain meaningful evolution. Our results underscore the potential of dynamically evolving training data for improving mathematical reasoning and suggest a promising direction for other domains, which we will facilitate by open-sourcing our code.

1 Introduction

Post-training of large language models (LLMs) is a highly active area of research, with recent methods focusing on designing training recipes that leverage real or synthetically generated datasets to enhance instruction-following ability [Ouyang et al., 2022, Wang et al., 2023b], coding performance [Nijkamp et al., 2023, Lozhkov et al., 2024], and mathematical reasoning [Shao et al., 2024, Hendrycks et al., 2021]. Two key limitations are the scarcity of high-quality data and the substantial compute required for training. We approach both challenges through the following research question:

How can we dynamically generate learnable and diverse training data that enables LLMs to bootstrap their own post-training?

One of the central motivations for this question is the need to obtain training data that remains well-suited to the model's current capabilities. A commonly observed issue is the prevalence of training examples with (near-)zero variance, which provide little to no learning signal and introduce noise into gradient updates [Foster and Foerster, 2025, Yu et al., 2025]. This not only hinders learning but also

*Equal contribution

Preprint. Under review.

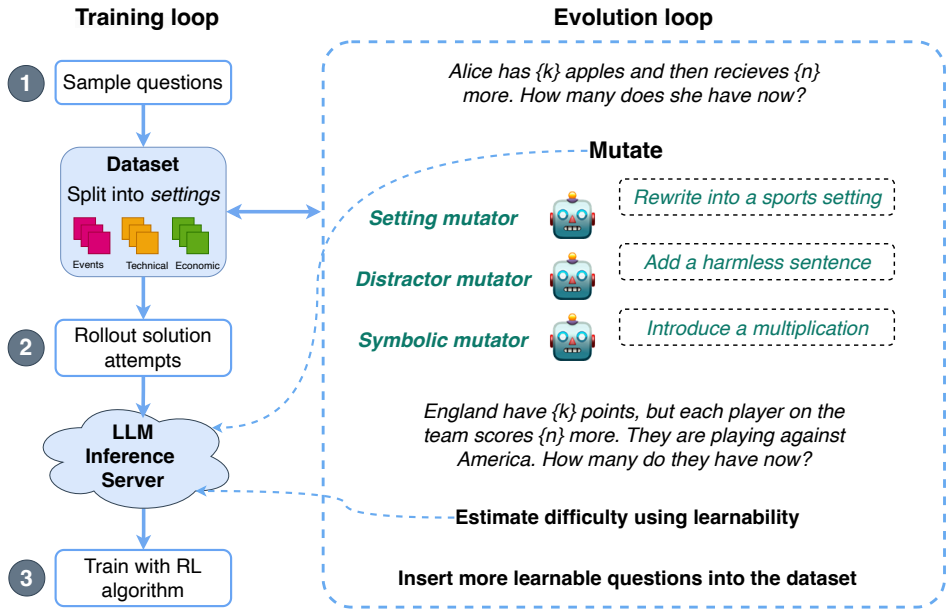


Figure 1: Overview of DÉJÀQ. We maintain an archive of question templates, organised by the setting each question applies to. Training data for RLVR is sampled from this archive, which is continuously updated through various *mutators* applied to existing templates. The *setting mutator* changes the setting (e.g., from Personal Life to Events), the *distractor mutator* introduces irrelevant information, and the *symbolic mutator* alters the underlying mathematical structure. Each question is scored by its *learnability* and retained or replaced accordingly.

wastes valuable compute. Although such examples can be filtered manually, this only underscores the broader issue of limited and ineffective training data. In this work, we introduce DÉJÀQ, a method that evolves a dataset of challenging yet solvable problems, explicitly optimised to maximise the model's learning progress.

The design of DÉJÀQ builds on three complementary ideas that have proven effective in reinforcement learning, including to some extent LLM post-training. From ACCEL [Parker-Holder et al., 2022], we adopt the principle of evolving training data jointly with model optimisation, rather than relying on a fixed dataset. From RAINBOW TEAMING [Samvelyan et al., 2024], we incorporate the use of MAP-Elites to maintain a structured archive of diverse training problems, and apply LLM-guided mutations to generate new high-quality examples in sparsely populated regions of the search space. From *learnability-based training* [Foster and Foerster, 2025], we take *learnability* as a proxy metric for the expected utility of a datapoint during training. DÉJÀQ unifies these components into a single framework that evolves a dataset of verifiable question–answer pairs through quality-diversity search for LLM post-training. The model continuously evaluates its own performance on newly generated problems, and those deemed sufficiently learnable are added to an archive for future training. Samples are retained based on their estimated potential to improve the model, thereby enabling open-ended bootstrapping without reliance on external supervision.

A core challenge in realising this framework is generating problems that are both *verifiable*, with ground-truth answers available by construction, and *skill-appropriate*, meaning they are neither trivial nor beyond the model's current capabilities. To address this, we introduce two complementary mutation strategies. The first is a template-based strategy, inspired by GSM-Symbolic [Mirzadeh et al., 2024], in which symbolic templates are instantiated with varying parameters to control difficulty. The second is an LLM-guided strategy, in which the model rewrites existing problems either by modifying their contextual framing or by altering their structure in a controlled way. Structural changes include the insertion of distractors, which are semantically coherent sentences that do not affect the solution, as well as symbolic modifications to the underlying operations in the solution.

We evaluate DÉJÀQ on models from the Qwen family [Yang et al., 2024a,b] at 1.5B, 3B, and 7B parameter scales. While these models are relatively small compared to current state-of-the-art models,

they remain of high interest for research under limited compute budgets. We train these models with DÉJÀQ and find strong evidence that both the introduction of synthetic reasoning data and the adaptivity of the curriculum improve performance. Notably, although LLM-guided mutations may theoretically introduce errors, our mutation operators are well-equipped to avoid such failures, and the learnability-based scoring acts as a natural safeguard against degenerate data.

We summarise our main contributions below and provide a visual overview of our method in Fig. 1:

1. **DÉJÀQ - Synthetic Data Evolution:** An evolutionary framework for constructing a dataset of highly learnable, verifiable question-answer pairs tailored to reasoning models.
2. **Different Mutation Strategies:** We propose four mutators, ranging from simple resampling to LLM-guided mutations, designed to maintain verifiability while increasing diversity and complexity.
3. **Efficient Bootstrapping:** The same model is used for both data generation and training, enabling a fully bootstrapped setup that leverages shared infrastructure.
4. **Empirical Validation:** We present a detailed empirical study showing that DÉJÀQ generates diverse and learnable problems for model training, particularly at larger model scales.

2 Background

2.1 Reinforcement Learning with Verifiable Rewards

Post-training of LLMs often involves a reinforcement learning (RL) phase, where a token-level Markov decision process (MDP) is defined by treating each token as an action and transitions as the concatenation of tokens to the existing context. Within this framework, the model is optimised using RL. A key requirement of this setup is a reward function, which in RLHF is typically learned from human feedback [Christiano et al., 2017, Ouyang et al., 2022]. Reinforcement Learning with Verifiable Rewards (RLVR) eliminates the need for such feedback by relying on verifiable reward signals [Lambert et al., 2024]. In mathematics, this may correspond to checking against ground-truth answers; in code generation, to evaluating against a comprehensive test suite. Formally, RLVR maximises the objective,

$$\mathbb{E}_{y\sim\pi_\theta(x)}\left[r_{\text{RLVR}}(x,y) - \beta D_{\text{KL}}(\pi_\theta(y \mid x) \parallel \pi_{\text{ref}}(y \mid x))\right] \tag{1}$$

where $r_{\text{RLVR}}(x,y) \in \{0,1\}$ denotes a verifiable binary reward, and the second term penalises deviation from a reference policy, weighted by the regularisation parameter β. Recently, the Group Relative Policy Optimisation (GRPO) algorithm has shown strong performance in mathematical domains [Shao et al., 2024]. Unlike its predecessor, PPO [Schulman et al., 2017], GRPO avoids reliance on a learned value network by sampling multiple generations and estimating advantages directly from them, offering both simplicity and improved stability.

2.2 MAP-Elites

To co-evolve a dataset of challenging yet solvable questions for the LLM to train on, we adopt a quality-diversity algorithm [Cully and Demiris, 2018], namely MAP-Elites [Mouret and Clune, 2015]. MAP-Elites maintains an archive of items $x \in \mathcal{X}$, where each item is assigned a feature descriptor via a mapping $d : \mathcal{X} \to \mathbb{R}^n$, and scored by a fitness function $f : \mathcal{X} \to \mathbb{R}$. In our setting, the feature space is discretised into a finite grid by assuming that each dimension of $d(x)$ is categorical. The archive is initially populated with a set of seed items $\{x_1, \ldots, x_k\}$, each inserted into its corresponding cell. Thereafter, the algorithm proceeds iteratively: at each step, an item $x \in \mathcal{X}$ is sampled from the archive and modified by a mutation operator $q : \mathcal{X} \to \mathcal{X}$, yielding a new item $x' = q(x)$. The mutated item x' is then assigned to a cell via $d(x')$, and scored using $f(x')$. Let y denote the current occupant of that cell. If the cell is empty or if $f(x') > f(y)$, then x' replaces y in the archive. Through repeated application of this procedure, MAP-Elites constructs an archive that is both diverse and high-quality.

3 Related Work

Curricula for LLMs. Training large language models (LLMs) typically consists of two phases, pre-training and post-training, both of which require substantial data and compute. To maximise

the utility of a fixed training budget, the design of effective learning curricula has emerged as a key strategy. In pre-training, Jin et al. [2023] introduce a sequence-length-based curriculum to improve efficiency, while Pouransari et al. [2024] apply a similar approach to address inefficiencies related to how documents are concatenated and chunked. Lin et al. [2024] propose Selective Language Modelling, which restricts loss computation to informative tokens. In post-training, recent state-of-the-art models have adopted hand-crafted curricula to guide training [Yu et al., 2025]. Beyond manual design, adaptive curriculum learning has gained traction. Foster and Foerster [2025] propose upsampling examples with high *learnability*, a proxy for how likely an input is to improve model performance. Similarly, Qi et al. [2025] apply evolution to web-based LLM agents, progressively generating more complex tasks to drive continual improvement. Finally, Shi et al. [2025] propose selecting training samples based on their proximity to a dynamically determined target difficulty, encouraging the model to focus on examples that are neither too easy nor too hard.

Reasoning Models. The pursuit of deploying large language models (LLMs) in domains such as mathematics and coding has led to the development of reasoning models that are explicitly trained to solve complex problems by generating coherent intermediate reasoning steps. Techniques like Chain-of-Thought (CoT) [Wei et al., 2022], Tree-of-Thought (ToT) [Yao et al., 2023] and Self-Consistency [Wang et al., 2023a] prompt models to articulate reasoning traces prior to producing answers. To further strengthen this capability, several iterative schemes have been proposed in which models generate reasoning samples, fine-tune on them, and repeat the process [Zelikman et al., 2022, Hosseini et al., 2024]. More recently, reinforcement learning (RL) approaches have shown that effective reasoning strategies can emerge without explicit instruction [Shao et al., 2024, DeepSeek-AI et al., 2025]. These methods often follow the inference-time compute paradigm, accepting increased computational cost during inference in exchange for improved downstream performance [Snell et al., 2024, Wu et al., 2025]. Among related approaches, WizardMath [Luo et al., 2023] uses GPT-4 both to evolve its training dataset and to supervise the reasoning process of student models, effectively outsourcing both problem generation and supervision to a static external oracle. This imposes an inherent ceiling on model performance, bounded by the capabilities of the supervising model. It also does not adopt a principled approach to dataset diversity, leaving the breadth and representativeness of its training distribution unclear. In contrast, our method constructs a verifiable dataset explicitly optimised for both diversity and learnability, enabling self-improvement without dependence on external supervision or fixed data sources.

4 Open-Ended Evolution of Diverse and Learnable Verifiable Problems

Our objective is to continually evolve a dataset of highly learnable reasoning problems alongside model training, while maintaining verifiability and diversity within the archive. To this end, we propose DÉJÀQ a novel LLM post-training method that curates a stream of challenging yet solvable problems tailored to the current capabilities of the reasoning model.

DÉJÀQ consists of two processes that are executed asynchronously: a dataset evolution loop and model post-training via RLVR. To evolve the dataset, we adopt the MAP-Elites algorithm [Mouret and Clune, 2015], a quality-diversity method that maintains an archive of word-problem templates indexed by a feature descriptor. Each cell in the archive retains the highest-scoring template encountered during evolution. While this work focuses on mathematical reasoning, DÉJÀQ is readily applicable to other domains, such as code generation, provided that suitable problem featurisations and mutation operators are defined. Full pseudocode is provided in Appendix B.

4.1 Initial Archive Population

To instantiate the MAP-Elites archive, we require a seed dataset $\mathcal{D}_0$ and a description function d that maps each datapoint to a set of features or categories. For the seed dataset, we draw inspiration from GSM-Symbolic introduced by Mirzadeh et al. [2024], who argue that existing frontier models are likely overfit on GSM8K [Cobbe et al., 2021] and propose a template-based alternative. We thus include all such templates in our initial dataset.

To define the descriptor function d, we manually inspect the templates and devise a classification scheme based on their *problem setting*, such as Professional, Economic, or Recreational. Each template is then assigned a category along this axis by instantiating a concrete problem and prompting a language model, Qwen2.5-32B-Instruct [Yang et al., 2024a] in our case, to generate a chain-of-

thought rationale followed by a final classification. The exact setting categories we consider are listed in Appendix B, and the prompt used for classification is provided in Appendix G.

4.2 Problem Scoring

To evolve a dataset that meaningfully advances the model's capabilities, we require a principled fitness function. We adopt *learnability* as the central scoring criterion. For a given problem instance x and model parameters θ, learnability is defined as $l_\theta(x) = p_\theta(x)(1 - p_\theta(x))$ [Tzannetos et al., 2023], where $p_\theta(x)$ denotes the probability that the model solves x correctly. This measure is grounded in learning theory and has been successfully applied in RL and LLM post-training for mathematical domains [Parker-Holder et al., 2022, Rutherford et al., 2024, Foster and Foerster, 2025].

In practice, the true probability $p_\theta(x)$ is unknown and must be estimated. To this end, we use a model inference server to generate K completions for each problem and compute the empirical success rate $\hat{p}_\theta(x)$ as the fraction of correct completions. Learnability is then estimated using the unbiased estimator $\hat{l}_\theta(x) = \frac{K}{K-1}\hat{p}_\theta(x)(1 - \hat{p}_\theta(x))$. An intuitive benefit of this scoring function is that malformed or unsolvable problems naturally receive low learnability scores and are therefore unlikely to be retained in the archive.

4.3 Mutating Problems While Preserving Verifiability

Base Template and Mutations

A fog bank rolls in from the ocean to cover a city. It takes `{t}` minutes to cover every `{d}` miles of the city. If the city is `{y}` miles across from the oceanfront to the opposite inland edge, how many minutes will it take for the fog bank to cover the whole city?
Solution: $t \cdot \frac{y}{d}$

— **Resample Mutator** (Instantiate template) —

A fog bank rolls in from the ocean to cover a city. It takes 256 minutes to cover every 9 miles of the city. If the city is 72 miles across, how many minutes will it take?
Solution: 2048

— **Setting Mutator** (Retain solution) —

Scientists are observing an air mass moving inland. It takes `{t}` minutes for the air mass to cover every `{d}` miles of the research area. If the research area is `{y}` miles across, how many minutes will it take for the air mass to cover the whole area?
Target category: Scientific

— **Distractor Mutator** (Retain solution) —

A fog bank rolls in from the ocean to cover a city. It takes `{t}` minutes to cover every `{d}` miles of the city. Starting from the oceanfront, the fog slowly spreads inland, covering the city block by block. If the city is `{y}` miles across from the oceanfront to the opposite inland edge, how many minutes will it take for the fog bank to cover the whole city?

— **Symbolic Mutator** (Modify solution) —

A fog bank rolls in from the ocean to cover a city. It takes `{t}` minutes to cover every `{d}` miles of the city. If the city is y miles across from the oceanfront to the opposite inland edge, how many minutes will it take for the fog bank to cover the whole city? The fog bank first needs to cover `{a}` miles on the outer edge of the city before it can start moving towards the center, and then it needs to cover the remaining `{b}` miles to finish covering the whole city.
Solution: $t \cdot \frac{(a+b)}{d}$

Figure 2: Example mutations of a fog coverage template under the operators used in DÉJÀQ. Shown are real generations produced by the 7B base model and obtained using the same prompts as applied during training.

A central challenge in constructing synthetic datasets for reasoning domains such as mathematics is balancing expressivity with verifiability. Models require access to sufficiently challenging and

diverse training data, yet the solutions to these problems must remain accessible to ensure meaningful supervision. Prior work circumvents this issue by relying on stronger teacher models to generate and validate data [Luo et al., 2023]. In contrast, our goal is to enable continual self-improvement without dependence on external oracles. To this end, we introduce four mutation operators within DÉJÀQ, each designed to expand the space of problem variations while aiming to preserve verifiability. All mutators are illustrated in Fig. 2.

Resample Mutator. As the archive is initialised from symbolic templates, a natural mutation strategy is to *resample* new problem instances without altering the underlying structure. Given a desired candidate category from the archive, we sample a fresh instantiation from the seed dataset. This ensures verifiability, as solutions can be directly computed from the template. However, expressivity is constrained by the diversity of available templates. In particular, several descriptor classes are represented by a single template, increasing the risk of overfitting within those regions.

Setting Mutator. To overcome the expressivity limits of resampling, we introduce an LLM-guided setting mutator, inspired by Samvelyan et al. [2024]. This mutator first identifies a category in the archive with low learnability and prompts an LLM to rewrite a high-quality parent problem to match that category. This facilitates exploration beyond manually defined templates, enabling more diverse and targeted problem generation.

Distractor Mutator. In addition to contextual rewrites, we propose a *distractor mutator*, which adds semantically irrelevant sentences to a problem. These distractors may reuse existing variables within the template as long as they do not interfere with the core reasoning steps. This mutator is designed to vary surface form without affecting the underlying solution.

Symbolic Mutator. While the setting and distractor mutators change the presentation of the problem, its underlying symbolic interpretation does not change. Consequently, we propose a final mutator that modifies the solution to a given template in a structured manner to also improve symbolic diversity. Two mutation types are defined: *root* and *leaf*. Root mutations append an additional computation to the final answer, effectively deepening the solution path. Leaf mutations modify an intermediate variable by redefining it through a new expression, altering the internal structure of the problem. We procedurally generate which operation to apply and with what operators and operands and subsequently prompt the LLM to adapt the question by introducing a new sentence in a predefined location.

Pitfalls of Evolution. While evolutionary mutation expands the problem space, it also introduces several challenges. Frequent reuse of high-quality parents can reduce diversity and increase the likelihood of errors, so we decrease the selection probability of archive items as parents based on their mutation depth. To further mitigate error accumulation, we periodically resample candidates from the seed templates, ensuring a steady influx of verifiable problems. To encourage substantive variation through our mutations, we adopt the filtering approach from RAINBOW TEAMING and use BLEU [Papineni et al., 2002] to admit candidates only if their surface form diverges sufficiently from the parent. Finally, to address staleness in learnability estimates [Parker-Holder et al., 2022], we apply a time-based decay factor and make use of the rollouts performed by GRPO to reset the scores.

4.4 LLM Inference Server Integration

A key consideration in our setup is the computational overhead introduced by estimating learnability and performing LLM-guided mutations, compared to training on static templated data. To avoid additional cost, we leverage the same LLM inference server already used during GRPO training. As our evolutionary framework only requires online generation, this shared infrastructure can be directly integrated. Implementation details are provided in Appendix B.

5 Experiments

We experimentally evaluate DÉJÀQ using the Qwen2.5(-Math) family of models at 1.5B, 3B, and 7B parameter scales [Yang et al., 2024a,b]. Our code is implemented on top of TRL [von Werra et al., 2020] for RL fine-tuning on the LLMs and vLLM [Kwon et al., 2023] for the model inference server.

Table 1: Mean accuracy on Qwen2.5-* models (prefix Qwen2.5- omitted). Bold indicates the best model on a given evaluation per parameter size. For s.e.m and significance testing see Appendix F.

		GSM-Symbolic (%)			Other Benchmarks (%)	
Model	Method	Symbolic	P1	P2	MATH-500	GPT-Eval
Math-1.5B-Instruct	Base	**80.4**	67.2	49.5	55.2	81.0
	DR	79.8	67.9	**50.3**	54.5	82.0
	DÉJÀQ-R	79.7	68.0	50.1	53.1	82.3
	DÉJÀQ-S	80.3	67.7	49.9	54.4	82.3
	DÉJÀQ-A	79.9	**68.0**	49.8	**55.3**	**84.5**
3B-Instruct	Base	76.1	64.2	40.1	44.8	**78.0**
	DR	76.2	64.6	39.6	43.7	75.0
	DÉJÀQ-R	**78.0**	**65.9**	**40.5**	**46.5**	74.3
	DÉJÀQ-S	77.4	65.5	40.3	46.3	73.7
	DÉJÀQ-A	77.1	65.1	40.1	45.9	74.3
7B-Instruct	Base	87.4	76.8	61.7	**55.4**	76.0
	DR	83.7	70.7	48.8	49.4	77.0
	DÉJÀQ-R	**91.0**	**80.8**	**65.4**	53.8	77.5
	DÉJÀQ-S	90.9	80.4	64.6	53.6	**81.0**
	DÉJÀQ-A	90.5	80.3	64.8	53.3	78.5

Methods. As baselines, we include the original base model as well as *domain randomisation* (DR), which samples uniformly from the set of available templates and instantiates them with valid parameters. We compare these against progressively more expressive variants of DÉJÀQ: the *resample* mutator (DÉJÀQ-R), the *setting* mutator (DÉJÀQ-S), and the full combination of *setting, distractor, and symbolic* mutators (DÉJÀQ-A). We do not evaluate the distractor or symbolic mutators in isolation, as they lack the ability to mutate across categories.

Benchmarks. Each model is evaluated on the Symbolic, P1, and P2 subsets of the GSM-Symbolic test set [Mirzadeh et al., 2024], as well as on the out-of-distribution MATH-500 benchmark [Hendrycks et al., 2021, Lightman et al., 2024]. Additionally, we compiled a new dataset of 100 diverse grade-level mathematics problems using GPT-4o [OpenAI et al., 2024], dubbed GPT-Eval. We introduce this dataset curation process in Appendix E.

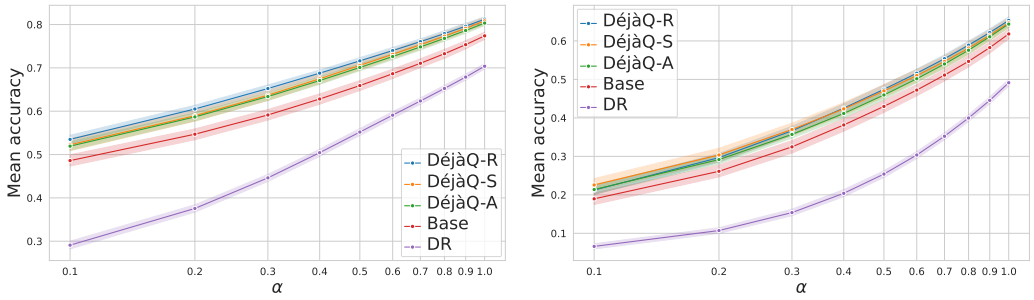
5.1 Insights on Evaluation Accuracy

Table 1 presents mean accuracy for the base model, the domain-randomisation (DR) baseline, and the three DÉJÀQ variants. Bold values indicate the best-performing method at each model scale.

Smaller models struggle. At the 1.5B and 3B scales, performance differences between the base model, domain randomisation, and DÉJÀQ variants are minimal. This likely reflects the limited ability of smaller models to guide expressive mutations effectively. In DÉJÀQ-S and DÉJÀQ-A, which apply setting, symbolic, and distractor mutators, the model often fails to preserve the structure of the original template, resulting in malformed or incorrect problems. This is supported by the learnability trends in Fig. 4a, where DÉJÀQ-R maintains substantially higher learnability over time.

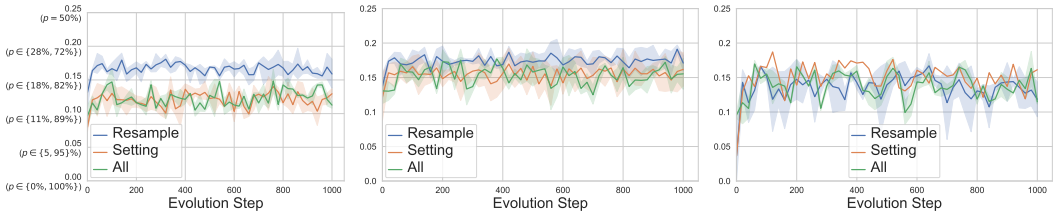
7B as a threshold. At the 7B scale, performance differences between methods become more apparent. DÉJÀQ-R achieves the highest accuracy on the in-distribution evaluation sets, Symbolic, P1, and P2, likely because its mutations remain within the space of known templates. In contrast, DÉJÀQ-A performs best on GPT-Eval, suggesting that the broader exploration enabled by LLM-guided mutations improves generalisation. These results indicate that 7B marks a critical capacity threshold beyond which models can effectively leverage an evolving training distribution. Notably, DÉJÀQ achieves substantial gains on in-distribution tasks without compromising performance on the out-of-distribution MATH-500 benchmark.

We further evaluate performance using the Conditional Value at Risk (CVaR) metric, proposed by Rutherford et al. [2024], which measures the expected success rate over the hardest α-proportion of tasks. Specifically, for a given $\alpha \in (0, 1]$, CVaR computes the mean success on the lowest



(a) P1 (b) P2

Figure 3: Conditional Value at Risk (CVaR) at varying α levels on the P1 and P2 test sets for the 7B model. We find that DÉJÀQ improves robustness for the more challenging problems.



(a) 1.5B (b) 3B (c) 7B

Figure 4: Mean learnability with 95% confidence interval of the complete archive over the first 1000 evolution steps, across different model sizes (1.5B, 3B, 7B) and mutation strategies (DÉJÀQ-R, DÉJÀQ-S, DÉJÀQ-A). Larger models generate more learnable examples and benefit more from complex mutations. Shaded regions indicate standard deviation across runs.

α-percentile of task performances. This emphasises robustness by focusing on difficult examples, unlike standard averages, which may obscure poor performance in challenging regions of the task distribution. We evaluate all model scales on the P1 and P2 subsets and show the results for the 7B models in Fig. 3 and all other results in Appendix F.

The results reveal a substantial gap between the domain randomisation models and all other variants. This is expected, as the base models have already been trained on data from within the same distribution, making further improvement through challenging without curating usable data. In contrast, all variants of DÉJÀQ consistently outperform the baseline, likely due to the increased diversity and learnability of the training data they generate.

Learnability matters. For both 3B and 7B models, variants that use learnability-guided sampling consistently outperform the base model and the domain randomisation (DR) baseline on most evaluation sets. This supports the idea that focusing training on examples that the model is close to learning can speed up progress [Tzannetos et al., 2023, Rutherford et al., 2024, Foster and Foerster, 2025]. In contrast, DR starts to fall behind at larger scales. At 7B, it performs worse than the base model on all metrics except GPT-Eval, suggesting that uniform sampling from a fixed set of templates adds noise and weakens learning. This shows the importance of adaptive sampling strategies that stay aligned with the model's current abilities.

5.2 Open-Ended Learning through Evolution

Fig. 4 tracks the mean learnability of the evolving archive over the first 1000 steps for three model sizes and mutation strategies.

Learnability over time. We observe that smaller models (1.5B, 3B) quickly plateau in learnability, especially under DÉJÀQ-R, suggesting that simple template instantiations are sufficient to challenge them. More complex mutations often exceed their capabilities or produce malformed examples, which are filtered out during evolution and result in lower average learnability. At 3B, the gap between

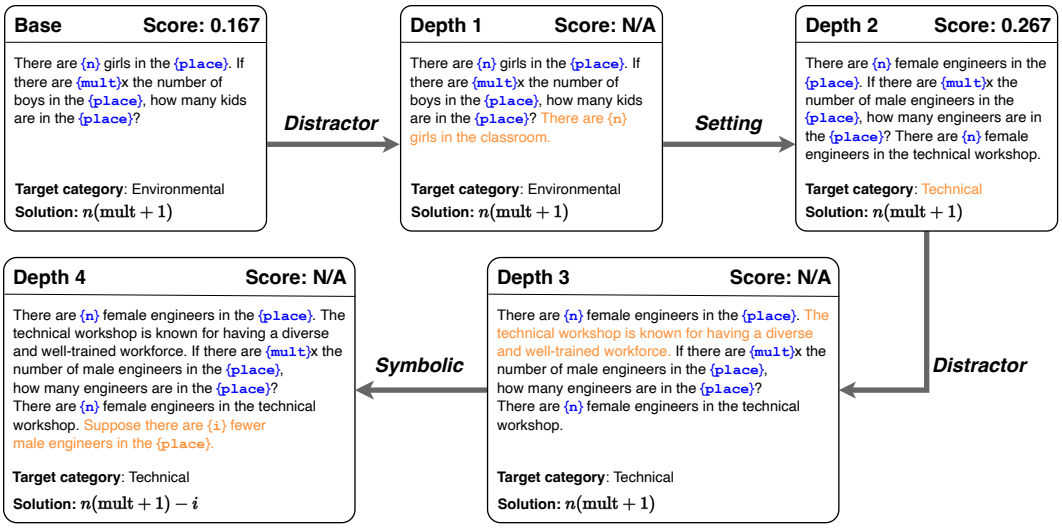


Figure 5: An example sustained chain of successful mutations generated by the 7B-Instruct model using our full method. Learnability is only computed after applying all mutators to the sample.

strategies narrows, and DÉJÀQ-S and DÉJÀQ-A show improved performance relative to the 1.5B scale. The 7B model behaves more dynamically, with learnability stabilising around 0.14 and exhibiting pronounced fluctuations, reflecting cycles of exploration and recovery. This indicates that larger models can evaluate and benefit from their own mutations more effectively. Overall, these results show that learnability enables open-ended curriculum evolution, though its effectiveness depends on the model's capacity when used to guide mutations. Importantly, current learnability remains well below the theoretical maximum of 0.25, leaving substantial room for improving mutation strategies and archive composition.

A little bit too open-ended? We designed the featurisation of GSM-symbolic templates to reflect a range of real-world domains we considered relevant. However, since DÉJÀQ does not enforce hard constraints on the types of problems that can be generated, we found that the model occasionally introduced new axes of variation. In particular, smaller models sometimes rewrote problems from English into Spanish, producing well-formed math questions that occasionally combined both languages. While training on these examples does not improve performance on our current (English-only) benchmarks, we hypothesise that the model becomes more robust along dimensions not captured by our limited standard evaluations. This highlights the importance of developing evaluation sets that better reflect the diversity and open-ended nature of real-world problems, or, if the goal is to stay within a more constrained domain, incorporating auxiliary filtering mechanisms such as a judge model as done in RAINBOW TEAMING.

Genealogy. Figure 5 shows a sustained chain of successful mutations generated at the 7B model scale, demonstrating that DÉJÀQ can move well beyond the original base templates. The base template, shown in the top left, contains three variables, two of which are used in the solution formula. It is classified under the Environmental setting, likely due to specific instantiations of the `place` variable. The first mutation appends a distractor sentence that reuses an existing variable `n` without altering the problem's structure. The second rewrites the problem to fit a different setting, Technical in this case, and is scored for its learnability, which is notably higher than that of its root template.[2] The third mutation inserts a harmless sentence after the opening, which blends naturally into the context without introducing new reasoning steps. Finally, the symbolic mutator appends a sentence that subtracts a new variable from the total. This example combines all LLM-guided mutation types and highlights the model's capacity to generate coherent and increasingly complex training data through iterative mutations. In Appendix E we provide more complete examples of the mutation chains and their respective learnability that highlight the open-ended nature of our method.

[2]Although learnability lies in the range $[0, 0.25]$, the unbiased estimator used in our case extends this range to $[0, 0.3]$ with $K = 6$.

6 Conclusion

We introduce DÉJÀQ, an evolutionary framework that leverages a suite of mutators to incrementally evolve a dataset of diverse and learnable problems tailored for reinforcement learning with verifiable rewards. Our method builds on MAP-Elites to maintain an archive of synthetic problems, selecting and retaining examples based on their learnability under the current model. To balance diversity and verifiability, we combine template resampling with LLM-guided mutations that reframe the question context (setting mutator), introduce irrelevant information (distractor mutator), and alter the underlying mathematical structure (symbolic mutator). Empirical results show that while smaller models struggle to produce coherent and useful mutations, models at the 7B scale begin to generate novel and learnable problems that sustain meaningful training. These findings underscore the potential of adaptive data generation as a foundation for scalable post-training and open-ended learning.

References

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

A. Cully and Y. Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2018. doi: 10.1109/TEVC.2017.2704781.

DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, and S. S. Li. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948.

T. Foster and J. Foerster. Learning to reason at the frontier of learnability, 2025.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S.-K. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, Virtual*, 2021.

A. Hosseini, X. Yuan, N. Malkin, A. Courville, A. Sordoni, and R. Agarwal. V-STaR: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*, 2024.

H. Jin, X. Han, J. Yang, Z. Jiang, C.-Y. Chang, and X. Hu. GrowLength: Accelerating llms pretraining by progressively growing training length. *CoRR*, abs/2310.00576, 2023. doi: 10.48550/ARXIV.2310.00576.

A. Kazemnejad, M. Aghajohari, E. Portelance, A. Sordoni, S. Reddy, A. Courville, and N. L. Roux. VinePPO: Unlocking RL potential for LLM reasoning through refined credit assignment, 2024.

W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with PagedAttention. In J. Flinn, M. I. Seltzer, P. Druschel, A. Kaufmann, and J. Mace, editors, *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM, 2023. doi: 10.1145/3600006.3613165.

N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, Y. Gu, S. Malik, V. Graf, J. D. Hwang, J. Yang, R. L. Bras, O. Tafjord, C. Wilhelm, L. Soldaini, N. A. Smith, Y. Wang, P. Dasigi, and H. Hajishirzi. TÜLU 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124, 2024. doi: 10.48550/ARXIV.2411.15124.

H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Z. Lin, Z. Gou, Y. Gong, X. Liu, Y. Shen, R. Xu, C. Lin, Y. Yang, J. Jiao, N. Duan, and W. Chen. Rho-1: Not all tokens are what you need. *CoRR*, abs/2404.07965, 2024. doi: 10.48550/ARXIV.2404.07965.

A. Lozhkov, R. Li, L. B. Allal, F. Cassano, J. Lamy-Poirier, N. Tazi, A. Tang, D. Pykhtar, J. Liu, Y. Wei, T. Liu, M. Tian, D. Kocetkov, A. Zucker, Y. Belkada, Z. Wang, Q. Liu, D. Abulkhanov, I. Paul, Z. Li, W.-D. Li, M. Risdal, J. Li, J. Zhu, T. Y. Zhuo, E. Zheltonozhskii, N. O. O. Dade, W. Yu, L. Krauß, N. Jain, Y. Su, X. He, M. Dey, E. Abati, Y. Chai, N. Muennighoff, X. Tang, M. Oblokulov, C. Akiki, M. Marone, C. Mou, M. Mishra, A. Gu, B. Hui, T. Dao, A. Zebaze, O. Dehaene, N. Patry, C. Xu, J. McAuley, H. Hu, T. Scholak, S. Paquet, J. Robinson, C. J. Anderson, N. Chapados, M. Patwary, N. Tajbakhsh, Y. Jernite, C. M. Ferrandis, L. Zhang, S. Hughes, T. Wolf, A. Guha, L. von Werra, and H. de Vries. StarCoder 2 and the stack v2: The next generation, 2024.

H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, and D. Zhang. WizardMath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *CoRR*, abs/2308.09583, 2023. doi: 10.48550/ARXIV.2308.09583.

S.-I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, and M. Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. *CoRR*, abs/2410.05229, 2024. doi: 10.48550/ARXIV.2410.05229.

J.-B. Mouret and J. Clune. Illuminating search spaces by mapping elites. *CoRR*, abs/1504.04909, 2015.

E. Nijkamp, B. Pang, H. Hayashi, L. Tu, H. Wang, Y. Zhou, S. Savarese, and C. Xiong. CodeGen: An open large language model for code with multi-turn program synthesis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae,

A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph. Gpt-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135.

J. Parker-Holder, M. Jiang, M. Dennis, M. Samvelyan, J. N. Foerster, E. Grefenstette, and T. Rocktäschel. Evolving curricula with regret-based environment design. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 17473–17498. PMLR, 2022.

H. Pouransari, C.-L. Li, J.-H. R. Chang, P. K. A. Vasu, C. Koc, V. Shankar, and O. Tuzel. Dataset decomposition: Faster LLM training with variable sequence length curriculum. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Z. Qi, X. Liu, I. L. Iong, H. Lai, X. Sun, J. Sun, X. Yang, Y. Yang, S. Yao, W. Xu, J. Tang, and Y. Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

A. Rutherford, M. Beukman, T. Willi, B. Lacerda, N. Hawes, and J. N. Foerster. No regrets: Investigating and improving regret approximations for curriculum discovery. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

M. Samvelyan, S. C. Raparthy, A. Lupu, E. Hambro, A. H. Markosyan, M. Bhatt, Y. Mao, M. Jiang, J. Parker-Holder, J. Foerster, T. Rocktäschel, and R. Raileanu. Rainbow teaming: Open-ended generation of diverse adversarial prompts. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300.

T. Shi, Y. Wu, L. Song, T. Zhou, and J. Zhao. Efficient reinforcement finetuning via adaptive curriculum learning, 2025.

C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314, 2024. doi: 10.48550/ARXIV.2408.03314.

G. Tzannetos, B. G. Ribeiro, P. Kamalaruban, and A. Singla. Proximal curriculum for reinforcement learning agents. 2023, 2023.

L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, K. Rasul, and Q. Gallouédec. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`, 2020.

X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023a.

Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics, 2023b. doi: 10.18653/V1/2023.ACL-LONG.754.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Y. Wu, Z. Sun, S. Li, S. Welleck, and Y. Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*, 2025.

A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024a. doi: 10.48550/ARXIV.2412.15115.

A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, K. Lu, M. Xue, R. Lin, T. Liu, X. Ren, and Z. Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024b. URL `https://arxiv.org/abs/2409.12122`.

S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, T. Fan, G. Liu, L. Liu, X. Liu, H. Lin, Z. Lin, B. Ma, G. Sheng, Y. Tong, C. Zhang, M. Zhang, W. Zhang, H. Zhu, J. Zhu, J. Chen, J. Chen, C. Wang, H. Yu, W. Dai, Y. Song, X. Wei, H. Zhou, J. Liu, W.-Y. Ma, Y.-Q. Zhang, L. Yan, M. Qiao, Y. Wu, and M. Wang. DAPO: An open-source LLM reinforcement learning system at scale, 2025.

E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman. STaR: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

A Limitations & Broader Impact

DÉJÀQ introduces a framework for jointly evolving a dataset of verifiable problems and answers, optimised for learnability. While our approach demonstrates strong potential, especially at larger model scales, it also presents several limitations. Smaller models often struggle to generate meaningful mutations and fail to benefit from more expressive mutators during training. Although we introduced a distractor and symbolic operator to increase flexibility, these can still lead to malformed or off-distribution examples when models cannot apply them reliably.

A key challenge arises from the open-ended nature of LLM-guided mutation. Because DÉJÀQ does not constrain the type of problems that can be produced, the model may explore axes of variation not originally intended by the designer, for instance, generating valid math problems in other languages. While such behaviour may reflect progress in robustness or generalisation, it complicates alignment with fixed evaluation sets. This highlights the need for more comprehensive benchmarks that better capture open-ended learning, or for auxiliary mechanisms that constrain mutations to task-relevant domains, such as judge models or domain classifiers.

Although our work focuses on mathematical reasoning, we believe DÉJÀQ can be extended to other domains, such as code generation. A generalised framework that evolves domain-specific curricula from a rule-based seed dataset could reduce dependence on manual data curation. However, as with any automated data generation system, careful safeguards are needed when applying it to sensitive or high-stakes domains.

B DÉJÀQ Implementation Details

In this section we provide all remaining implementation details for DÉJÀQ.

B.1 Setting Categorisation

The following table presents the setting categorisation used in our DÉJÀQ experiments and was derived through a combination of manual analysis and LLM-assisted inspection.

Table 2: The setting categories used in our DÉJÀQ experiments.

Name	Description
Personal Life	Scenarios from everyday personal experiences involving home life, family, school, food, health habits, or individual routines.
Professional	Contexts involving occupations, productivity, workplace responsibilities, or services rendered as part of a job or trade.
Economic	Situations involving money, costs, purchases, income, trade, markets, or financial decision-making.
Recreational	Scenarios focused on hobbies, play, sports, games, or other leisure activities pursued for enjoyment.
Events	Social or organised occasions such as birthdays, holidays, celebrations, school fairs, or community gatherings.
Scientific	Problems involving biological, chemical, or physical concepts, including natural processes and scientific observations.
Technical	Scenarios involving machines, devices, or engineered systems where understanding tools, parts, or operational constraints is essential.
Environmental	Scenarios involving ecosystems, weather, agriculture, conservation, or interactions between humans and the natural world.

B.2 LLM Inference Server Integration

As noted in Section 4.4, our approach integrates dataset curation into the same inference infrastructure used for training. We elaborate here on why the additional inference cost is justified and how this integration can be made efficient in practice.

First, prior work has shown that training on low-information samples can negatively impact model performance by slowing down overall training and introducing noise into the gradient updates [Yu et al., 2025, Foster and Foerster, 2025]. Filtering out such instances in advance can therefore result in more effective gradient updates, offsetting the added inference cost. Second, recent RLVR methods employed in LLM post-training, such as GRPO [Shao et al., 2024] and VinePPO [Kazemnejad et al., 2024], already rely on fast, online sampling. These methods typically use a separate inference server such as vLLM [Kwon et al., 2023] to generate rollouts in real time. Importantly, this server is often underutilised during phases such as backpropagation or data staging.

By integrating dataset curation into the same inference infrastructure, we make more efficient use of available resources without incurring additional overhead. In our implementation, we employ an agnostic scheduling strategy that queries the inference server opportunistically—whether for training, scoring, or data generation. Identifying an optimal schedule that maximises throughput while avoiding interference with training remains a non-trivial engineering challenge and an open direction for future work.

B.3 Pseudocode

Algorithm 1 The DÉJÀQ algorithm. Shared components are highlighted in blue.

Require: Initial model parameters θ_0, seed dataset $\mathcal{D}_0$, mutation operator q, and training budget T
Ensure: A post-trained reasoning model with parameters θ_T

Initialise LLM inference server
Initialise MAP-Elites archive $\mathcal{A} \leftarrow \emptyset$
Populate $\mathcal{A}$ with seed problems from $\mathcal{D}_0$ and compute learnability scores $l(x; \theta_0)$

Launch two asynchronous processes:
`(1) Model Training Loop`
for $t = 1$ to T **do**
    Sample training batch $\mathcal{B}$ from $\mathcal{A}$
    Update model via RLVR: ▷ Uses LLM inference server to sample generations

$$\theta_t \leftarrow \arg\max_{\theta} \mathbb{E}_{y \sim \pi_\theta(x)} \left[r_{\text{RLVR}}(x, y) - \beta \, \mathrm{D}_{\text{KL}} \left(\pi_\theta(y \mid x) \,\|\, \pi_{\text{ref}}(y \mid x) \right) \right]$$

end for

`(2) Dataset Evolution Loop`
while training is running **do**
    Sample $x \sim \mathcal{A}$
    Generate mutant $x' \leftarrow q(x)$ ▷ Uses LLM inference server to propose mutations
    if x' is correctly formatted **then**
        Compute score $s' \leftarrow l(x'; \theta_t)$ ▷ Uses LLM inference server to estimate learnability
        Assign descriptor $d \leftarrow d(x')$
        if $d \notin \mathcal{A}$ **or** $s' > l(\mathcal{A}[d]; \theta_t)$ **then**
            $\mathcal{A}[d] \leftarrow x'$
        end if
    end if
end while
return θ_T

C Generating the GPT-Eval Test Data

As described in Section 5, we construct a synthetic evaluation set of grade-school-level mathematical problems using a frontier model as the data generator [OpenAI et al., 2024]. Specifically, we prompt GPT-4o to generate batches of 10 questions and corresponding answers based on a fixed specification (see Appendix G for the full prompt). This process is repeated until a dataset of 100 examples is obtained.

To validate the dataset, we use GPT-o3, a reasoning model with strong mathematical capabilities, to evaluate the generated question-answer pairs. For each problem that is not verified by GPT-o3, we additionally request a corrected answer. We manually reviewed a sample of both verified and corrected problems and found no errors in GPT-o3's judgments.

D Experiment Details

In this section, we provide the hyperparameters and computational resources used to perform our experiments.

D.1 Hyperparameters

Table 3: Combined Configuration Parameters for *training*, and *evolution*.

Parameter	Value
Training Parameters	
reward_funcs	cos_correctness, format
reward_weights	2.0, 1.0
algorithm	GRPO
learning_rate	1.0e-06
adam_beta1 / beta2	0.9 / 0.99
weight_decay	0.1
warmup_ratio	0.1
lr_scheduler_type	cosine_with_min_lr
lr_scheduler_kwargs	min_lr_rate: 0.1
optim	paged_adamw_8bit
gradient_accumulation_steps	4
gradient_checkpointing	true
gradient_checkpointing_kwargs	use_reentrant: false
num_generations	6
scale_rewards	true
max_prompt_length	512
max_completion_length	2048
per_device_train/eval_batch_size	6 / 6
num_iterations	1
max_steps	1000
use_vllm	true
Evolution Parameters	
cell_size	3
ignore_top_k	6
score_decay	0.95
score_alpha	0.5
bleu_threshold	0.6
resample_prob	0.25
structure_probs	distractor: 0.4, symbolic: 0.4, both: 0.2, none: 0.0
max_tries	5
mutation_batch_size	16
sample_decay	0.5

D.2 Computational Resources

Our experiments were conducted on a compute cluster consisting of nodes equipped with NVIDIA A40 and NVIDIA L40S GPUs, each with 48 GB of VRAM. For the 1.5B and 3B models, we used one GPU for vLLM inference and one GPU for training. For the 7B models, one GPU was allocated to vLLM and three GPUs were used for training. Depending on cluster load, each run completed within one to two days.

E Genealogy Illustration

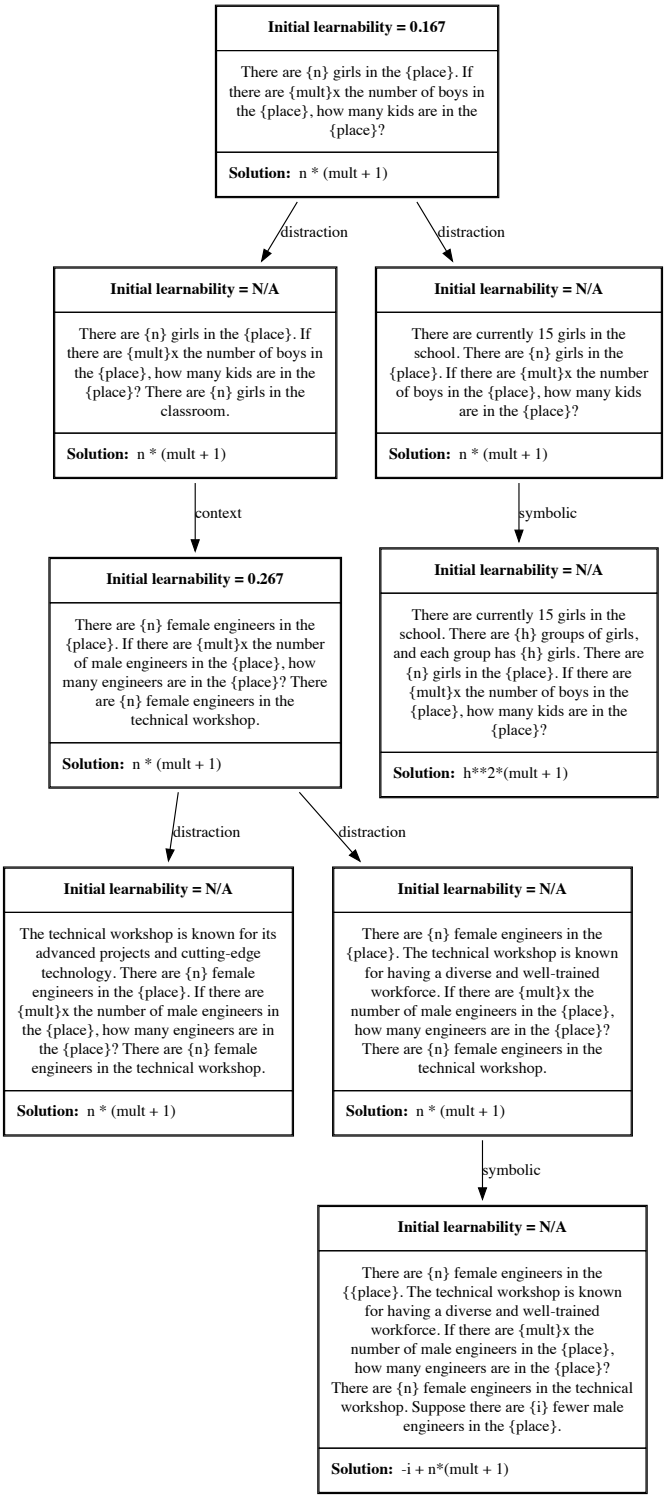
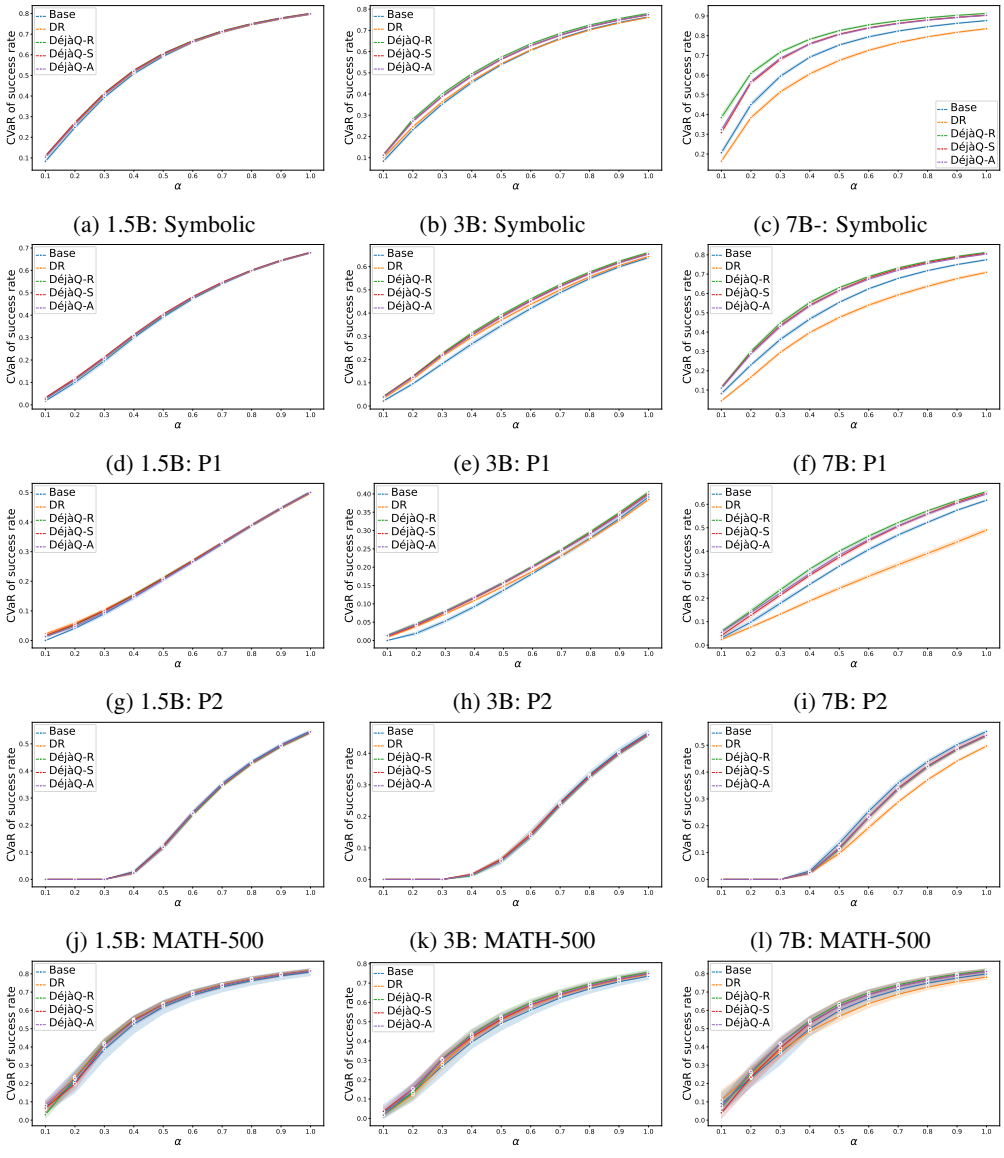


Figure 6: Extended genealogy of the example from Fig. 6. Each template includes a problem string, a solution formula, and the learnability score at the time it was first admitted to the archive. Templates with learnability marked as N/A were not scored, as they represent intermediate mutations during DÉJÀQ-All. Arrows indicate the applied mutation type.

F Additional Results

We present the CVaR results for the 1.5B and 3B models, along with the results for the 7B models on the remaining evaluation datasets not presented in the main paper. Furthermore, in Table 4 we also provide standard errors for the accuracy table presented in Section 5.



(a) 1.5B: Symbolic (b) 3B: Symbolic (c) 7B-: Symbolic

(d) 1.5B: P1 (e) 3B: P1 (f) 7B: P1

(g) 1.5B: P2 (h) 3B: P2 (i) 7B: P2

(j) 1.5B: MATH-500 (k) 3B: MATH-500 (l) 7B: MATH-500

(m) 1.5B: GPT-Eval (n) 3B: GPT-Eval (o) 7B: GPT-Eval

Figure 7: CVaR of success across varying α values for each evaluation dataset. "1.5B" refers to Qwen2.5-Math1.5B-Instruct, "3B" to Qwen2.5-3B-Instruct, and "7B" to Qwen2.5-7B-Instruct.

Table 4: Mean accuracy on Qwen2.5-* models (prefix Qwen2.5- omitted). **Bold** indicates the best model on a given evaluation per parameter size. † indicates results from a single training seed or a pretrained base model; for non-base models this was due to resource constraints.

		GSM-Symbolic (%)			Other Benchmarks (%)	
Model	Method	Symbolic	P1	P2	MATH-500	GPT-Eval
Math-1.5B-Instruct	Base†	79.6 ± 0.12	67.8 ± 0.17	$\mathbf{50.2} \pm 0.21$	$\mathbf{54.7} \pm 0.24$	81.0 ± 0.94
	DR	79.7 ± 0.07	67.8 ± 0.08	49.8 ± 0.13	54.2 ± 0.19	$\mathbf{81.9} \pm 0.34$
	DéjàQ-R	$\mathbf{80.1} \pm 0.07$	68.0 ± 0.12	50.0 ± 0.15	54.2 ± 0.22	81.7 ± 0.43
	DéjàQ-S	80.0 ± 0.07	$\mathbf{68.1} \pm 0.06$	50.1 ± 0.12	54.4 ± 0.18	81.6 ± 0.40
	DéjàQ-A	79.8 ± 0.09	67.9 ± 0.09	50.0 ± 0.20	54.4 ± 0.29	81.6 ± 0.54
3B-Instruct	Base†	76.4 ± 0.12	63.9 ± 0.20	39.2 ± 0.29	$\mathbf{46.6} \pm 0.40$	73.6 ± 0.89
	DR	76.1 ± 0.08	64.4 ± 0.16	38.5 ± 0.16	45.9 ± 0.23	74.6 ± 0.53
	DéjàQ-R	$\mathbf{78.0} \pm 0.08$	$\mathbf{66.1} \pm 0.11$	$\mathbf{40.6} \pm 0.13$	45.8 ± 0.24	$\mathbf{75.6} \pm 0.58$
	DéjàQ-S	77.4 ± 0.10	65.6 ± 0.11	40.1 ± 0.12	46.2 ± 0.20	74.9 ± 0.56
	DéjàQ-A	77.4 ± 0.10	65.4 ± 0.13	39.8 ± 0.16	45.9 ± 0.21	74.9 ± 0.64
7B-Instruct	Base†	87.6 ± 0.11	77.5 ± 0.15	61.7 ± 0.19	$\mathbf{55.1} \pm 0.31$	79.9 ± 0.84
	DR	83.5 ± 0.16	70.9 ± 0.27	49.0 ± 0.43	49.7 ± 0.24	78.1 ± 0.63
	DéjàQ-R	$\mathbf{91.3} \pm 0.10$	$\mathbf{81.3} \pm 0.12$	$\mathbf{65.4} \pm 0.14$	53.5 ± 0.20	$\mathbf{81.8} \pm 0.50$
	DéjàQ-S†	90.4 ± 0.10	80.8 ± 0.20	64.8 ± 0.23	53.8 ± 0.47	81.2 ± 0.77
	DéjàQ-A	90.3 ± 0.06	80.4 ± 0.09	64.4 ± 0.25	53.6 ± 0.16	81.2 ± 0.63

G Prompts

Qwen Math System Prompt

Please reason step by step, and put your final answer within `\boxed{}`.

Teacher System Prompt

You are a knowledgeable and patient mathematics teacher. Aim to develop the student's intuition and problem-solving skills. You will be given math problems along with specific instructions, and your task is to revise or adapt the problems to best meet those instructions.

Setting Mutate Prompt Template

You will receive:

- **candidate_context**: The target setting you must adapt the template into, e.g., "Personal life".
- **template**: A word-problem template that contains **placeholders enclosed by single curly braces**, e.g., ‘name‘, ‘a‘, ‘unit‘.

—

Your task: Rewrite the template to fit the *candidate_context*.

Strictly follow these rules:

1. **Do not change the mathematical structure or variables.**
   - The **exact same set of placeholders (identified by the text inside the curly braces)** **must appear in the rewritten template—no more, no less**.
   - **Placeholders are case-sensitive. Do not rename, add, or remove any of them.**
2. **Adapt the contextual details** so the story naturally fits the *candidate_context*.
   - **Even if the template already matches the candidate_context, you must still rewrite it** into a different story or situation within the same context.

- The new version should be clearly distinct while remaining fully appropriate for the candidate_context.

3. Ensure the rewritten template is **clear, grammatically correct, and fully self-contained**.

4. **Always write in English.**
   - The rewritten template must be fully in English, including names, descriptions, and story elements.
   - Do not switch to other languages, even partially.

—

Output format

First, reason step by step before writing your final answer.
- Keep the reasoning brief and focused—just a few sentences.
- Identify the current general setting of the template.
- Describe how the *candidate_context* differs (if it differs) and what contextual changes are needed.
- Specify what must remain unchanged in the template (placeholders and mathematical structure).
- Plan how you will adapt the story to fit the *candidate_context*, creating a clearly distinct version even if the context remains the same.

Only after this reasoning, output the result as the JSON object below—no extra text or explanation:

```
json
{
  "mutated_template": "<the rewritten template>"
}
```

—

Inputs:

[candidate_context]
{{ candidate }}

[template]
{{ question_template }}

Distractor Mutate Prompt Template

You will receive:
- **template_before**: The part of the word problem **before** the insertion point.
- **template_after**: The part of the word problem **after** the insertion point.

—

Your task: Insert a harmless sentence that adds detail without changing the answer.

Strictly follow these rules:
- You may **reuse existing placeholders**, but must **not change their meaning or mathematical role**.
- **Do not introduce any new variables** that affect the solution or reasoning.
- The inserted sentence can add context or narrative colour, but **must not alter any quantities, relationships, or the logic required to solve the problem**.
- Ensure the inserted text is **clear, grammatically correct, and fully self-contained**.

- **Always write in English.**

—

Output format

First, reason step by step before writing your final answer.

- Identify which placeholders you reused.
- Justify why the inserted sentence does not affect the original solution.

Only after this reasoning, output the result as the JSON object below—no extra text or explanation:

```
json
{
  "inserted_text": "<the harmless sentence you inserted>"
}
```

—

Inputs:

[template_before]
{{ template_before }}

[template_after]
{{ template_after }}

Symbolic Mutate Prompt Template

You will receive:

- **question_before**: The segment of the word problem **before** the insertion point.
- **question_after**: The segment of the word problem **after** the insertion point.
- **change**: change — a symbolic description of how the mathematical structure must be updated.

—

Your task
Write **one or two concise sentences** to be inserted between *question_before* and *question_after* so that the whole problem now realises **change**.

Global writing rules

- Touch **only** the placeholders mentioned in **change**; do **not** invent others.
- Ensure the new text is **clear, grammatical, self-contained,** and blends with the surrounding story.
- Keep everything in the **same language (English)**; avoid personal names unless they already appear.

{% if modify_root and new_var %}

Instructions – *root* modification with a **new** variable
‘change‘ defines a **new overall answer** ‘new_answer‘ by applying the indicated operation to the current answer (‘old_answer‘) and a **brand-new placeholder**. Introduce that new placeholder, describe how it combines with ‘old_answer‘, and keep the narrative consistent. {% elif modify_root and not new_var %} ### Instructions – *root* modification with an **existing** variable
‘change‘ defines a **new overall answer** ‘new_answer‘ by combining ‘old_answer‘ with an **existing placeholder** via the specified operation. Reference the existing placeholder explicitly and describe the operation.
{% else %}
Instructions – *leaf* modification (expand a placeholder)
‘change‘ rewrites **one existing placeholder** (call it ‘target‘) as an expression
target = left $\oplus$ right

where ‘left‘ and ‘right‘ are **exactly two fresh placeholders** and $\oplus$ is the given operation.

Write an insertion that:

- **Mentions the two new placeholders** and gives them concrete meaning in the story.
- **States or clearly implies** the relation between them and ‘target‘ exactly as in ‘change‘.
- Connects smoothly to what comes before and after (you may refer to ‘target‘ if helpful).

Tip: Think of telling the reader how the quantity ‘target‘ *came to be* by using ‘left‘ and ‘right‘.
{% endif %}

—

Output format
First think with the scaffold below. Then output **only** the JSON block.

Scaffold

- *Placeholders*: reused $\rightarrow$ [. . .], new $\rightarrow$ [. . .]
- *Draft*: "<insertion sentence(s)>"
- *Why it fits*: . . .

Final visible output

```
json
{
  "inserted_text": "<your final insertion>"
}
```

—

Inputs

[question_before]
{{ question_before }}

[question_after]
{{ question_after }}

[change]
{{ change }}

—

Illustrative examples

{% if modify_root %}

Example 1 (addition, *root-new*)

[question_before]
Sam saved {k} dollars last month.

[question_after]
How much money does Sam have after last month’s savings?

[change]
‘{new_answer} = {old_answer} + {new_var}‘

inserted_text: This month, he received a performance bonus of {new_var} dollars. How much money does Sam have now?

```
json
{ "inserted_text": "This month, he received a performance bonus of {new_var}
dollars. How much money does Sam have now?" }
```

—

Example 2 (multiplication, *root-existing*)

[question_before]
The factory produced {a} widgets yesterday. It plans to repeat this output for {days} more days.

[question_after]
How many widgets will the factory produce in total?

[change]
‘{new_answer} = {old_answer} * {days}‘

inserted_text: Oh, but then it will repeat that output for {days} consecutive days.

```
json
{ "inserted_text": "Oh, but then it will repeat that output for {days}
consecutive days." }
```

—

{% else %}
Example 1 (division, *leaf*, gift story)

[question_before]
{name} has {target} apples but eats {n}. How many does she have left?

[question_after]
How many apples does she have left?

[change]
‘{target} = { new_var_1} / { new_var_2}‘

inserted_text: The {target} apples came from her mother, who split { new_var_1} apples evenly among her { new_var_2} children.

```
json
{ "inserted_text": "The {target} apples came from her mother, who split
{ new_var_1} apples evenly among her { new_var_2} children." }
```

—

Example 2 (addition, *leaf*)

[question_before]
Julia completed a weekend hike totalling {target} kilometres.

[question_after]
How many kilometres did she hike over the weekend?

[change]
‘{target} = { new_var_1} + { new_var_2}‘

inserted_text: She walked { new_var_1} km on Saturday and { new_var_2} km on Sunday.

```
json
{ "inserted_text": "She walked { new_var_1} km on Saturday and
{ new_var_2} km on Sunday." }
```

—

Example 3 (subtraction, *leaf*)

[question_before]
After a spill, the bag now contains {target} marbles.

[question_after]
How many marbles are in the bag now?

[change]
‘{target} = { new_var_1} - { new_var_2}‘

inserted_text: The bag originally held { new_var_1} marbles, but { new_var_2} rolled away.

```
json
{ "inserted_text": "The bag originally held { new_var_1} marbles,
but { new_var_2} rolled away." }
```

—

Example 4 (multiplication, *leaf*)

[question_before]
A rectangular field covers {target} square metres. It consists of identical plots.

[question_after]
What is the total area of the field?

[change]
‘{target} = { new_var_1} * { new_var_2}‘

inserted_text: Each plot measures { new_var_1} metres by { new_var_2} metres.

```
json
{ "inserted_text": "Each plot measures { new_var_1} metres by
{ new_var_2} metres." }
```

—

Example 5 (division, *leaf*)

[question_before]
The car covered {target} kilometres in {k} hours.

[question_after]
What was its average speed?

[change]
‘{target} = { new_var_1} / { new_var_2}‘

inserted_text: This trip was the first of a total of { new_var_2} in a road trip totalling { new_var_1} kilometers.

```
json
{ "inserted_text": "This trip was the first of a total of { new_var_2} in
a road trip totalling { new_var_1} kilometers." }
```

—

{% endif %}

Algebra Dataset Generator Prompt

You are a dataset generator for algebra word problems. Your task is to create a dataset of 10 diverse and well-structured math word problems.

Each problem should:

- Be a realistic **word problem** involving algebraic reasoning.

- Require multiple of the basic arithmetic operations: **addition (+), subtraction (-), multiplication (*), or division (/)**.
- Include **numerical answers** that are either **integers** or **simple decimal values** (e.g. at most two decimal places).
- Design each question to be meaningfully challenging, requiring multiple steps of reasoning or intermediate calculations to reach the correct answer.

Diversity Requirements:

- Ensure **broad topical diversity** across a wide range of real-world domains.
- Ensure **diversity in problem structure**: some can be direct, others can include extraneous information or require intermediate steps.
- Vary **question phrasing** and not just content — avoid repetitive templates.
- Aim for a **balanced distribution** across the four operations and their combinations.

Output Format:
Return the dataset as a **Python list of dictionaries**. Each dictionary must have the following keys:

- "problem": the text of the word problem (as a string)
- "answer": the correct numerical answer (as a number, not a string)

Example (with 3 entries):

```
python
[
  {
    "problem": "A delivery truck makes 3 stops. At the first stop, it unloads
    15% of its 200-item cargo. At the second stop, it delivers 40 more items.
    At the third stop, it unloads half of what remains. How many items are
    left in the truck after all three stops?",
    "answer": 30
  },
  {
    "problem": "A biologist observes that a bacteria population triples every
    hour. If the initial population is 200 and after 2 hours 120 bacteria are
    removed, what is the population at the end of the second hour?",
    "answer": 1680
  },
  {
    "problem": "A conference room has 12 rows of chairs with 15 chairs in each
    row. If 3 rows are reserved and only 80% of the remaining seats are
    occupied, how many people are seated?",
    "answer": 108
  }
]
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction present DÉJÀQ, list its four contributions and do not claim results that are not substantiated by the empirical evaluation.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated "Limitations & Broader Impact" section (Appendix A) that discusses key limitations of the approach. Additional limitations are also addressed in Section 5, particularly in the context of model behaviour and failure modes observed during training.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The work is entirely empirical; no new theorems or proofs are proposed.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: An anonymised repository in the supplementary materials contains all code, fixed seeds, an environment file, and step-by-step scripts that regenerate every main table and figure.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The full codebase will be released under a permissive license upon acceptance, and the anonymised clone already accompanies the submission.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide a tabular overview of all relevant hyperparameters in Appendix D. Furthermore, the same values are hard-coded in the released code configuration files.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report 95% confidence intervals for the CVaR plots in Section 5 and Appendix F and also provide them for the learnability plots in Fig. 4. For the accuracy evaluations, we do not include statistical significance tests due to the limited number of seeds. We also emphasise that the main contribution of the paper lies in the qualitative analysis of the data generation process and the design of the mutator framework, rather than in specific evaluation metrics.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the computational resources needed for experiments in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conform to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix A outlines positive impacts (e.g. improved mathematical capabilities) and negative risks (e.g. misuse).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No high-risk assets (model weights or private datasets) are released, so additional safeguards are unnecessary.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Each external dataset and software package is cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper introduces no new datasets or pretrained models.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The study does not involve human participants or crowdsourced data.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human-subject research is conducted; IRB review is therefore not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The method fine-tunes open-weight LLMs; model sources, training recipe, and usage constraints are detailed in the methodology section and Appendix.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.