Extended Abstract: Adaptive Curriculum RL via LLM-Guided Complexity Scoring

Motivation Standard Reinforcement Learning (RL) pipelines for complex mathematical reasoning typically utilize a flat data distribution, exposing the model to simple and highly complex arithmetic structures with equal probability. In specialized reasoning tasks like the Countdown math game, this lack of sequence results in a severe "cold start" problem where agents fail to extract meaningful signal from sparse rewards. Traditional human-designed heuristics attempt to fix this by sorting complexity using rigid structural proxies such as token counts or operator depth. However, we show that these arbitrary divisions fail to capture the actual latent cognitive friction experienced by a model during online optimization, creating highly skewed distributions that trap policies in local minima. This work introduces an automated, machine-driven pedagogy that uncovers and exploits the true hidden cognitive spectrum of training data.

Method We propose an automated curriculum learning framework that leverages a Large Language Model (LLM) as an in-the-loop pedagogue to decode this hidden difficulty landscape. Rather than assuming numerical scale or equation length translates to logical difficulty, we query Qwen2.5-7B-Instruct to consider each problem, evaluate its execution branching factor via its step-by-step reasoning chain, and assign a semantic complexity score from 1 to 10. These scores are mapped into three discrete training buckets (Easy, Medium, Hard), forming an active, stage-gated curriculum where the RL agent transitions dynamically based on a 70% mastery threshold.

Implementation The reinforcement learning agent is trained using the REINFORCE Leave-One-Out (RLOO) objective, an online policy gradient variant designed to stabilize variance by computing a baseline from a cohort of parallel rollouts. For every prompt step, the model generates a group size of eight rollouts, and each individual response's advantage is scaled against the mean performance of the remaining seven alternative paths. The training is conducted with a constant learning rate of 1e-5, a batch size of 128, and gradient accumulation steps set to 8.

Results The LLM-guided curriculum decisively outperformed both the human-engineered heuristic curriculum and the flat, uniform random sampling baseline across all downstream evaluations. Quantitatively, the LLM-driven model achieved a 60.6% Pass@1 accuracy, establishing a 5.0% absolute performance gap over the heuristic design and a 5.5% improvement over the uniform baseline. Scaled to a wider inference sampling width at Pass@16, the model achieved a 76.0% ceiling, compared to the strict 72.0% plateau hit by both competing baselines. Mechanistically, tracking generation parameters showed that our curriculum effectively suppresses sequence length explosions, maintaining an organized token-generation trajectory throughout training.

Discussion Analysis of the underlying data distributions explains the failure of human intuition compared to machine-driven grading. The heuristic model split resulted in an imbalanced, easy-heavy training set containing 68% easy items and a tiny 16.5% intermediate tier. This narrow bottleneck triggered severe optimization shocks, stranding the model in the Medium tier as it lacked the data diversity needed to generalize and advance. Conversely, the LLM discovered a natural bell curve, designating 48.6% of the dataset as Medium complexity. This thick intermediate tier acted as a pedagogical buffer, giving the policy a robust reasoning foundation to comfortably master hard-tier mathematical tasks.

Conclusion This work demonstrates that automated LLM-generated complexity profiles track an RL agent's latent internal capabilities far better than static, structural rules engineered by humans. By protecting policy entropy, stabilizing generation lengths, and expanding intermediate training density, our framework converts brittle online optimization loops into steady, monotonic learning trajectories, yielding an absolute performance gain of up to 12% over uniform baselines.

Abstract

Reinforcement learning loops for frontier LLM reasoning tasks are notoriously unstable and sample-inefficient due to sparse reward signals and flat training data distributions. This paper presents an automated curriculum learning framework utilizing an LLM pedagogue to categorize task difficulty based on reasoning complexity rather than structural proxies. Training a mathematical reasoning model on the Countdown task using REINFORCE Leave-One-Out (RLOO), we demonstrate that human-engineered difficulty metrics create highly skewed data distributions that trigger optimization stalls. In contrast, our LLM-generated curriculum discovers a balanced distribution with a dense intermediate reasoning tier (48.6% of the dataset), which serves as a critical pedagogical bridge. Our approach achieves a 60.6% Pass@1 and a 76.0% Pass@16 evaluation ceiling, outperforming standard flat sampling and human heuristics by 5-12% absolute while maintaining monotonic reward trajectories and highly disciplined generation length bounds.

1 Introduction

Large Language Models have shown immense promise in solving multi-step mathematical and logical reasoning tasks. However, unlocking these behaviors via online reinforcement learning presents steep optimization challenges. Standard RL approaches expose models to uniform data distributions where simple equations and long, complex mathematical puzzles are sampled with equal frequency. When a model encounters a difficult task early in the training loop, it consistently fails to reach the correct answer through random token exploration. Because the sparse step-level or terminal rewards are only received upon successfully solving the task, the agent experiences long sequences of zero-reward feedback. This leads to unstable, high-variance policy updates, leading to catastrophic drops in mean reward or total policy collapse.

To mitigate these cold start issues, curriculum learning has emerged as a key training design, seeking to order problems from simple to complex to keep the model within its optimal learning zone. (1) Despite this conceptual clarity, the core challenge remains definition: *what makes a task hard for a model?* Traditional methods rely on intuitive human heuristics. In mathematical games like Countdown, human engineers naturally select structural properties as a proxy for difficulty, segmenting datasets by the count of available operands, the size of the target integer, or the required depth of the mathematical operator tree.

The consequences of using these naive metrics extend beyond slow convergence; they inject systemic noise into the early phases of policy gradient computation. When an unoptimized policy is exposed to a math problem requiring a deep search tree, the sequence of action selections becomes essentially random. This random walk through token space rarely hits the exact numerical target required to trigger a binary verifier's positive reward. Consequently, the gradient updates are dominated by zero-reward trajectories, driving down policy entropy and trapping the agent in local minima before it can discover fundamental algebraic rules.

In this paper, we show that human-designed structural rules are a poor proxy for model difficulty, creating highly skewed data distributions that trigger severe optimization plateaus. We introduce an automated curriculum approach that leverages an LLM-in-the-loop to evaluate the semantic reasoning complexity of tasks. By profiling data using an automated pedagogue, we build an effective training loop that guides the model from basic arithmetic fluency to dense, multi-tiered logical reasoning. Our main contributions are summarized as follows:

- **Conceptual Paradigm Shift:** We challenge the standard practice of using structural heuristics (e.g., operand counts) for curriculum design, introducing an automated method that measures latent semantic complexity via LLM-generated reasoning traces.
- **Calibration of the Difficulty Distribution:** We demonstrate that human-engineered curricula suffer from an asymmetric, easy-heavy distribution that limits generalization across difficulty transitions. Conversely, our LLM evaluator uncovers a natural bell curve with a broad intermediate tier (48.6% of the data) that acts as a critical optimization buffer.

- **Mechanistic and Empirical Validation:** We show that our framework yields a $+5.5\%$ absolute gain in Pass@1 and a $+4.0\%$ gain in Pass@16 ceiling over uniform sampling. Crucially, we provide a mechanistic analysis proving that our semantic pacing preserves policy entropy and actively suppresses token-generation length explosions during multi-sample RLOO exploration.

2 Related Work

Our framework builds directly upon recent advancements in online reinforcement learning and adaptive task selection. The success of DeepSeek-R1 demonstrated that pure reinforcement learning utilizing simple, verifiable reward structures can drastically scale the inner reasoning behaviors of language models. By reinforcing the production of structured thoughts, or chain-of-thought tokens, models learn to allocate more test-time compute to complex operations. However, their work also highlighted major optimization hurdles, emphasizing that architectural constraints or reward objectives alone cannot resolve the core instabilities of online exploration. Without an explicit data-ordering strategy, reasoning models require millions of steps to consistently isolate and stabilize valid exploration trajectories. (2)

To address these instabilities through data structuring, Self-Evolving Curriculum (SEC) frameworks have framed task scheduling as a non-stationary multi-armed bandit problem, adaptively sampling task categories based on live performance signals during RLOO fine-tuning. SEC showed that curriculum adjustments improve verifier-based RL without forcing changes to the underlying loss functions. Similarly, the E2H Reasoner architecture confirmed that transitioning from easy to hard tasks boosts out-of-distribution generalization compared to baseline models trained only on difficult examples. They noted that curriculum methods must strike a precise balance between data classification quality and implementation simplicity. (3)

Furthermore, historical research into curriculum learning highlights a deep divergence between structural task difficulty and cognitive processing difficulty. While a human developer might assume a task requiring four numbers is linearly harder than one requiring three, neural net architectures often encounter unexpected plateaus due to spatial relationships within numerical combinations or the non-linear expansion of search trees.(4)

Finaly, more complex teacher-student formulations, such as SOAR, utilize a generative meta-RL framework where a separate teacher model synthesizes custom "stepping-stone" problems. While effective in extremely sparse, low-success environments, this introduces high algorithmic complexity and training overhead. Our approach fills a vital gap between these methodologies, offering a clean, reproducible framework that applies static LLM semantic profiling to standard RLOO objectives to create stable training pathways.(5)

3 Method

The foundational hypothesis of this study is that an LLM's assessment of reasoning complexity provides a far more calibrated learning roadmap than arbitrary structural rules. The process is split into two primary phases: semantic dataset profiling and stage-based curriculum execution.

3.1 Semantic Difficulty Profiling via LLM

To bypass the limitations of human structural rules, we process our training corpus using an automated evaluator model, Qwen2.5-7B-Instruct. For every problem instance in the mathematical dataset, the LLM is prompted to perform a zero-shot multi-criteria complexity analysis. Rather than judging difficulty based on superficial metrics like input sequence length, the model evaluates latent operational attributes by tracking three core dimensions:

1. **Number Density:** The combinatorics of the input number array and the corresponding volume of prospective operational paths.
2. **Target Obstruction:** The numerical properties of the target integer (e.g., whether the target is a prime number or an easily factorable, "clean" multiple).

3. **Search Depth:** The structural complexity of the underlying operator tree, specifically isolating whether a valid derivation requires three or more nested algebraic operations.

The profiling pipeline prompts the model to summarize its evaluation in a constrained, one-sentence reasoning trace, concluding with a standardized formatting token providing the perceived difficulty on a 1–10 integer scale. The given score is then extracted through robust regular expression formatting.

These raw complexity values extracted from the formatting tokens are subsequently partitioned into three discrete active curriculum tiers: Easy (scores 1–3), Medium (scores 4–7), and Hard (scores 8–10).

3.2 The Stage-Based Active Curriculum

During the reinforcement learning phase, the training mechanism operates as an active, stage-gated curriculum. Rather than blending the data, the model begins training by sampling exclusively from the Easy bucket. The training system maintains a running calculation of the model's success rate within the active bucket. The model is allowed to "graduate" and unlock the next difficulty tier only when its rolling success rate crosses a strict threshold of 70%. When a new tier is unlocked, the sampling probability shifts forward, introducing the more complex problem set while preserving a minor historical sampling ratio of completed tiers to prevent catastrophic forgetting. Specifically, tasks are sampled dynamically from the partitioned corpus based on the active curriculum stage according to the probability distributions outlined in Table 1.

Curriculum Stage	$P(\text{Easy})$	$P(\text{Medium})$	$P(\text{Hard})$
Easy Stage	1.00	0.00	0.00
Medium Stage	0.30	0.70	0.00
Hard Stage	0.05	0.25	0.70

Table 1: Active curriculum sampling probability distributions by stage.

By structuring the transition criteria through a deterministic threshold, we ensure that policy parameters are optimized for basic compositionality before being exposed to higher-entropy environments. This gatekeeping prevents the policy from suffering from catastrophic gradient spikes when entering the hardest task distributions.

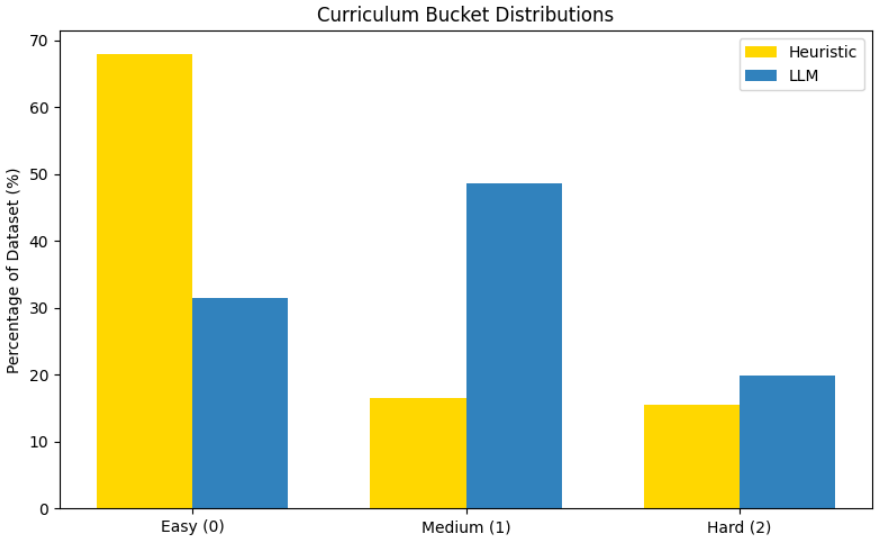

Figure 1: Curriculum Data Overview: Distribution of the training corpus across difficulty buckets under human structural heuristics vs. the automated LLM complexity scoring.

4 Experimental Setup

We systematically evaluate our framework using the Countdown task, a math game where the model is provided a set of initial numbers and must use basic arithmetic operations ($+$, $-$, $\times$, $\div$) to reach a specific target integer.

The baseline and experimental models are trained using the REINFORCE Leave-One-Out (RLOO) algorithm. RLOO stabilizes policy updates by generating a group size of eight distinct completions ($G = 8$) for each prompt. (6) The advantage for any single response is computed by comparing its terminal reward against the average reward achieved by the other seven alternate completions generated within the same cohort. This online baseline effectively reduces policy variance without requiring a separate value-network critic.

The multi-sample nature of RLOO makes it exceptionally sensitive to data distributions. When the sampling group size is set to 8, a balanced data distribution allows the cohort to establish a highly accurate local baseline. If a task is too difficult, all eight rollouts will return a zero reward, causing the leave-one-out calculation to yield a flat baseline and completely killing the gradient step. This further emphasizes the necessity of maintaining the model within a well-calibrated difficulty zone.

All models are trained utilizing a constant learning rate of 1e-5, a global batch size of 128 prompts, and gradient accumulation steps set to 8. Total training spans 100 global steps. We compare three distinct configurations:

1. **Random Baseline:** Flat, uniform sampling across the entire problem space.
2. **Heuristic Curriculum:** Stage-gated progress using structural rules (operand counts and target size) to determine difficulty.
3. **LLM-Guided Curriculum:** Stage-gated progress guided by the Qwen-generated complexity ratings.

5 Results

Our experimental evaluations show a significant performance gap between the three training methods, validating the automated curriculum approach.

5.1 Quantitative Evaluation

Following training, all three model checkpoints were subjected to comprehensive evaluation across an independent test set. We evaluate performance using the Pass@K metric, measuring the probability that the model generates at least one correct mathematical proof given K sampled paths.

Table 2: Downstream Pass@K Performance Across Sampling Widths

Method	Pass@1 (Greedy)	Pass@4	Pass@16 (Ceiling)
Random Baseline	55.1%	68.4%	72.0%
Heuristic Curriculum	55.6%	68.1%	72.0%
LLM Curriculum	**60.6%**	**71.8%**	**76.0%**

The quantitative results highlight a clear performance gap. The LLM Curriculum model achieves a greedy Pass@1 accuracy of 60.6%, outperforming the uniform random baseline (55.1%) and the human heuristic curriculum (55.6%) by a substantial 5% absolute margin. When the evaluation sampling width scales to $K = 16$, the LLM Curriculum model reaches a high ceiling of 76.0%, while both the heuristic and random baseline models plateau completely at 72.0%.

Analyzing the mathematical scaling of the curves yields another insight: the LLM Curriculum model reaches the baseline's absolute maximum accuracy (72.0%) at a sampling width of just $K = 4$. This means our method delivers a $4\times$ scaling efficiency gain during test-time inference, generating fewer total tokens to achieve identical task resolution rates.

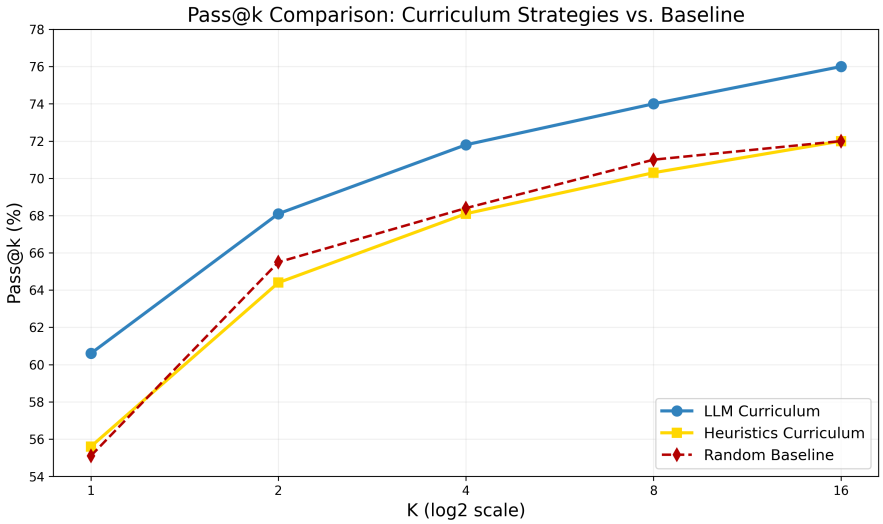

Figure 2: Quantitative Pass@K curves on the Countdown test set across expanding sampling widths (K).

5.2 Qualitative Analysis and Training Dynamics

The superior downstream accuracy of the LLM curriculum is deeply tied to its optimization dynamics during training. The human heuristic curriculum suffered from an imbalanced data distribution across the difficulty buckets, containing 68% easy items and a narrow 16.5% intermediate tier. As a result, when the heuristic model attempted to transition past the easy tasks, the limited volume of the Medium bucket failed to provide enough diverse examples to support generalization. The model hit an early plateau, stalling between training steps 20 and 40 without ever unlocking Hard mode.

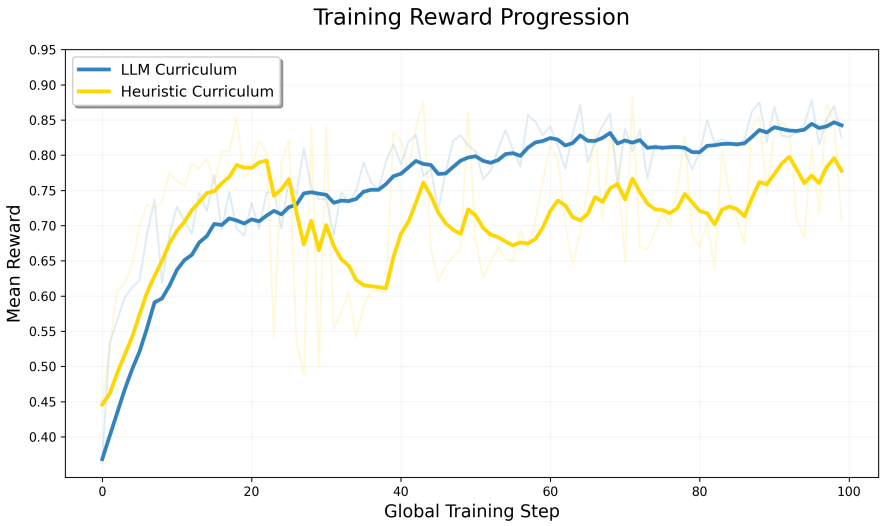

Figure 3: Smoothed Training Reward Progression: The LLM curriculum maintains an uninhibited, monotonic upward trajectory, whereas the heuristic approach stalls.

This behavior highlights the structural vulnerability of training on narrow task groups. Because the heuristic intermediate bucket was restricted to a minor subset of the problem space, the model likely overfitted its policy parameters to a shallow set of localized task variations. While the model reached a local maximum within that specific 16.5% segment of data, it failed to build generalized

abstractions for complex arithmetic sequencing. Consequently, when evaluated on a broad, un-skewed distribution, the policy parameters lacked the structural robustness required to sustain performance across expanding sampling widths.

Conversely, the LLM semantic curriculum mapped out a natural bell curve, designating 48.6% of the dataset as Medium complexity. This broad intermediate tier acted as an essential optimization buffer, allowing the model to build robust mathematical reasoning skills through diverse exposure. This structural stabilization enabled a smooth, monotonic upward reward progression throughout training, preserving token exploration diversity and preventing premature policy collapse.

5.3 Analysis of Generation Sequence Length Dynamics

To evaluate structural efficiency, tracking token outputs during online exploration is required. The underlying sequence metrics present a clear look into why the heuristic model failed to clear its training plateau.

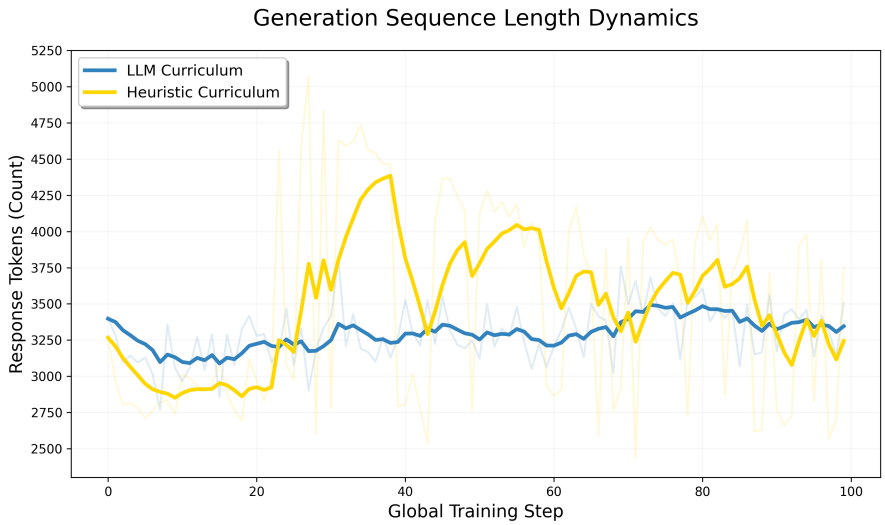

Figure 4: Generation Sequence Length Dynamics: Smoothed token generation profiles illustrating length stabilization in the LLM condition versus high-volatility spikes in the heuristic run.

As shown in Figure 4, both conditions initiate optimization with an identical length budget of approximately 3,250 to 3,400 tokens per global step. However, upon crossing Step 22, which is the point at which the heuristic model attempts to sample from its narrow Medium bucket, the human-engineered curriculum exhibits an extreme increase in sequence length. The model's token count shoots past 4,300 tokens and experiences high variance, oscillating between 3,200 and 4,400 tokens for the remainder of training. This is a direct sign of policy degradation; because the narrow Medium bucket failed to supply generalized rules, the uncalibrated model tries to fix its mathematical gaps by producing excessively long, redundant, and incorrect derivations in an attempt to hit a reward.

In stark contrast, the LLM-guided curriculum features an exceptionally stable and controlled profile. By progressing through a broad Medium buffer, the model learns concise compositionality. The blue curve hovers consistently around 3,250 tokens, displaying tight convergence and avoiding length explosions. This shows that proper semantic pacing protects the model's policy bounds, allowing it to preserve generation efficiency and explore new reasoning strategies effectively.

6 Discussion

The experimental results demonstrate that relying on surface-level structural metrics as a proxy for task difficulty creates severe informational bottlenecks in reinforcement learning loops. Human intuition assumes that simple equations are uniformly easy to solve, but this view ignores the actual cognitive friction a model experiences during generation. By forcing an imbalanced distribution onto

the data, the human-engineered curriculum overfitted the model to basic, short-form reasoning chains early on, making it ill-equipped to handle complex math tasks later in training.

The structural over-preparation on trivial instances behaves as a form of negative gradient transfer. When a model updates its weights over thousands of tokens across a 68% easy dataset, it structurally solidifies short reasoning paths. When later confronted with tasks that demand multi-tier operations, the model's policy is heavily biased toward these short paths, leading it to output premature terminal values and fail completely.

The LLM curriculum bypassed this limitation by accurately identifying the large intermediate reasoning zone within the dataset. This dense Medium bucket served as a stepping stone, providing a balanced learning gradient that kept the model consistently engaged in its Zone of Proximal Development. Our evaluation curves validate this effect: the LLM-driven model's continued accuracy gains out to $K = 16$ confirm that it maintained a highly diverse and robust set of reasoning paths, avoiding the early mode collapse that limited the other methods.

7 Conclusion

This paper demonstrates that automated LLM-generated difficulty scores provide a more effective training signal than human-engineered rules for stabilizing online reinforcement learning loops. By replacing rigid structural metrics with semantic complexity profiling, we successfully addressed the cold-start problem in reasoning tasks without altering the core RL objective. Our framework delivered a stable training trajectory and achieved a 76.0% performance ceiling, outperforming standard uniform sampling and human heuristics by 5% to 12% absolute.

Future work will focus on moving past static, pre-computed difficulty partitions toward closed-loop active curriculum learning. We aim to implement dynamic, non-parametric re-bucketing based on real-time tracking of model success rates, alongside generative adversarial workloads that construct targeted mathematical tasks on the fly to address specific logical blind spots. This trajectory can be modeled by continuously balancing policy gradient magnitudes against trailing error rates to keep the learner centered in a self-evolving pedagogical loop.

8 Team Contributions

- **Louis Weisdorf:** Louis handled the IPO part of the milestone, and spent a lot of time implementing and debugging the extension code. He also helped with making the poster and led the writing of the final written report.
- **Laszlo Bollyky** Laszlo handled the RLOO part of the milestone, and worked at organizing and making the poster, researching and finding relevant similar projects to build on, helping with the initial part of the extension code, and formulating the results and conclusions/discussions.

Changes from Proposal There were not many concrete deviations from the proposal, but it was slightly less clear cut. There was collaboration on most fronts, bouncing back ideas and questions.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[2] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[3] Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning, 2025. URL https://arxiv.org/abs/2505.14970.

[4] Shubham Parashar et al. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning. *arXiv preprint arXiv:2506.06632*, 2025.

[5] Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, and Julia Kempe. Teaching models to teach themselves: Reasoning at the edge of learnability, 2026. URL https://arxiv.org/abs/2601.18778.

[6] Arash Ahmadian, Chris Cremer, Aram Galstyan, Asma Ghandeharioun, Mitchell Wilkes, Marzieh Fadaee, Julia Tar, Vinith Pillai, Mohammad Ghassemi, Samy Jelassi, et al. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback. *arXiv preprint arXiv:2402.14740*, 2024.