UpSkill: Mutual Information Skill Learning for Structured Response Diversity in LLMs

Devan Shah* **Owen Yang***
Daniel Yang **Chongyi Zheng** **Benjamin Eysenbach**
Princeton University
{devan.shah, owen.yang, daniel.yang, chongyiz, eysenbach}@princeton.edu

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has improved the reasoning abilities of large language models (LLMs) on mathematics and programming tasks, often by maximizing `pass@1` correctness. However, optimizing single-attempt accuracy can inadvertently suppress response diversity across repeated attempts, narrowing exploration and overlooking underrepresented strategies. We introduce UpSkill, a training time method that adapts *Mutual Information Skill Learning* (MISL) to LLMs to induce *structured response diversity*: a discrete latent z selects a reproducible "strategy" that steers the token distribution toward distinct modes. We propose a novel reward that we implement within Group Relative Policy Optimization (GRPO): a *token-level* mutual information (MI) reward that encourages trajectory specificity to z. Experiments on GSM8K with three open-weight models, Llama 3.1–8B, Qwen 2.5–7B, and R1-Distilled–Qwen2.5–Math–1.5B show that UpSkill improves multi-attempt metrics for Qwen 2.5–7B, yielding gains of $\sim$4% in `pass@k` and $\sim$4% in `plurality@k` without degrading `pass@1`. Additionally, we prove that improvements in `pass@k` are closely tied to the mutual information objective, providing a theoretical justification for UpSkill.

Keywords: Language Models, Mutual Information, Reasoning, Response Diversity, RLVR

1 Introduction

LLMs excel at verifiable reasoning tasks such as mathematical problem solving and code generation [21]. However, repeated sampling often yields highly similar outputs [42]. This is detrimental in multi-attempt settings where just one correct completion solves the problem at hand, such as code generation with tests [5] or formal proofs in Lean [54], as a lack of diversity reduces the effective number of independent attempts. Therefore, for these or other objectives evaluated by `pass@k`, or the probability that at least one of k completions will be correct, more deterministic output decreases the chance that some sampled attempt will succeed. Furthermore, recent work has found that post-training that optimizes single-attempt correctness suppresses response variation across attempts [7, 9], creating a discrepancy between how models are trained and how they are used and evaluated.

The challenge of balancing diversity and accuracy, or exploration and exploitation [51], has primarily been studied in prior works that change how decoding is done. Methods such as temperature sampling [39], nucleus sampling [24], and prompt perturbations [48] can inject variety, but they require manual tuning [12] and are brittle across domains [47, 37]. Separately, prior training-time methods do not expose a controllable basis of strategies and require a complex training regime to properly balance exploration and exploitation [52, 7]. We seek a training-time mechanism that *(i)*

*Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Scaling Environments for Agents (SEA).

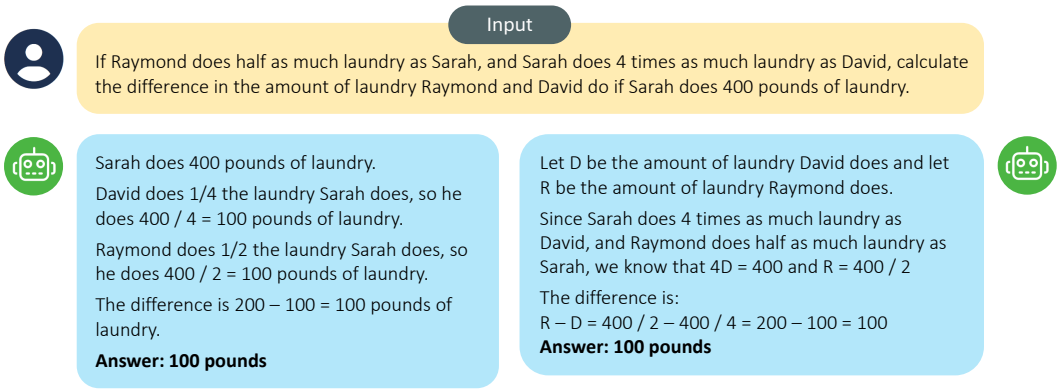

Figure 2: UpSkill is an unsupervised method for training LLMs to produce diverse responses. After training, different latent vectors z (blue boxes above) correspond to different response strategies. Because of space constraints, the figure shows summarized responses from UpSkill; we report the full responses in Appendix K.

increases diversity in a controlled manner, *(ii)* produces semantically distinct and reproducible modes of reasoning, and *(iii)* preserves single-attempt verifiable accuracy.

We introduce UpSkill, a training-time approach that induces *structured response diversity* without prompt engineering. The key idea behind UpSkill is to introduce an input token z that structures the response, so that different values of z correspond to different responses. Formally, we will model LLM attempts on verifiable reasoning tasks as a token-level Markov decision process. We can then adopt prior work from reinforcement learning on learning *skills*, which learn a policy conditioned on a latent variable z. These methods [14, 19, 1, 44, 16] include a loss term that maximizes the mutual information between z and the policy's behavior. Precisely, we adapt the CSF method [65] to LLMs: the model conditions its response on a discrete latent $z \in \{1, \ldots, N\}$, and training encourages behaviors whose distribution depends strongly on z. Intuitively, each z should correspond to a reproducible strategy, and the set of strategies should span a broad range of behaviors.

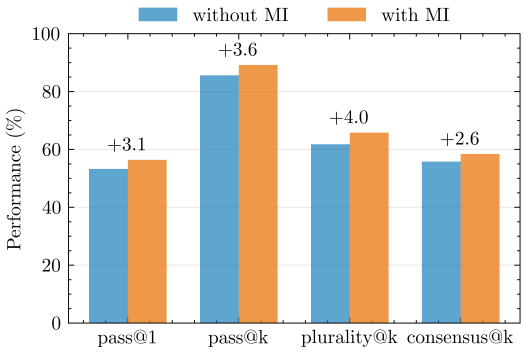

Figure 1: Token-level MISL improves multi-attempt accuracy without hurting single-attempt accuracy on GSM8K for the Qwen 2.5-7B model (See Sec. 5.2).

The main contribution of our paper is a method for training LLMs to produce diverse responses. Our method implements mutual information skill learning by applying GRPO [43] with a novel reward term: a token-level mutual information reward, which encourages diversity in completions. Finally, we sketch a theoretical link between $\mathcal{I}(\tau; z \mid x)$ and `pass@k`: the improvement of `pass@k` after training is related to $\mathcal{I}(\tau; z \mid x)$, In summary, our contributions are as follows:

- UpSkill achieves gains of +3.6% in `pass@k` and +4.0% in `plurality@k` on GSM8K on Qwen 2.5–7B using RL fine-tuning with LoRA adapters on 2,000 problems, with preserved `pass@1` accuracy. We additionally test two other open-weight models, although we do not observe improved performance.
- In an arithmetic puzzle environment, UpSkill improves `pass@5` by +10% by mitigating response variation collapse and developing a collection of diverse and complementary skills.

- We prove that `pass@k` improvement closely corresponds to the mutual information $\mathcal{I}(\tau; z \mid x)$, showing that large improvements in multi-attempt accuracy require — and are limited by — sufficient mutual information.
- We provide an effective and reproducible method for token-level MI, and will release an open-source implementation focused on practical performance.

Training and evaluation code is open-sourced at https://github.com/dshah02/upskill.

2 Background and Related Work

2.1 Multi-attempt evaluation, redundancy, and why diversity matters

For verifiable tasks, we often consider the probability of success across *multiple* completions rather than a single attempt [7]. Let x denote the input and τ a sampled completion from policy $\pi(\cdot \mid x)$. Let $Y(\tau) \in \{0, 1\}$ indicate correctness under a deterministic verifier. For k attempts, the standard metric

$$\text{pass@}k(x) \;=\; 1 - \Pr\left(\bigcap_{i=1}^{k} \{Y(\tau_i) = 0\} \,\middle|\, x \right) \tag{1}$$

is the complement of the joint failure probability across k i.i.d. draws $\tau_{1:k} \sim \pi(\cdot \mid x)$ [5]. Letting $p = \Pr(Y(\tau) = 1 \mid x)$, we therefore have $\text{pass@}k(x) = 1 - (1 - p)^k$.

In practice, identical prompts with fixed decoding hyperparameters can yield strongly correlated trajectories, especially for deterministic or near-deterministic samplers [56]. A useful lens is to consider an "effective number of attempts" k_{eff} that discounts k by a correlation term (analogous to design effects in sampling) [27]. If completions have pairwise correlation ρ in the binary success indicators, a heuristic adjustment gives $k_{\text{eff}} \approx k/(1 + (k-1)\rho)$: as $\rho \to 1$, additional attempts contribute little; as $\rho \to 0$, $k_{\text{eff}} \to k$. Although crude, this highlights the central point: reducing dependence among attempts is as important as raising per-attempt accuracy. Structured diversity aims to decrease redundancy so that the joint failure probability decreases faster in k. For Gaussian random variables, correlation and mutual information are closely related (as intuitively, correlated variables have information on each other) [28]. However, as text correctness cannot easily be framed as a Gaussian distribution, mutual information is more a natural measurement.

Beyond `pass@k`, `plurality@k` and `consensus@k` measure agreement among completions, examining robustness and internal consistency of the model's reasoning [57]. In many workflows, agreement acts as a proxy for confidence while still benefiting from diversity to escape shared failure modes [23].

2.2 RL on language models: token-level MDPs, RLVR, and GRPO

Autoregressive LLMs can be cast as policies over a Markov decision process (MDP), where the state is the token prefix and the action is the next token [3, 36]. This setup, often referred to as the token-level MDP [66], allows reinforcement learning algorithms to directly optimize model behavior for correctness on verifiable tasks such as math or code.

Reinforcement Learning from Verifiable Rewards (RLVR) leverages automatically checkable signals (e.g., exact numeric answers, unit tests) as rewards, with the goal being to improve the pass rate of policies while ensuring the new policy remains close to a base model [33, 11]. Let π_θ denote the trainable policy and π_{base} a frozen reference. A common form of the per-trajectory reward is

$$r_{\text{RLVR}}(\tau) \;=\; r_{\text{correctness}}(\tau) \;-\; \beta\, D_{\text{KL}}\big(\pi_\theta(\cdot \mid x) \,\big\|\, \pi_{\text{base}}(\cdot \mid x)\big)\,, \tag{2}$$

where $\beta > 0$ controls deviation from the base model [60].

Group Relative Policy Optimization (GRPO) [43] adapts PPO-style updates to reasoning by sampling multiple completions per prompt x as a *group*. Within-group baselines reduce variance and increase the relative difference between completion rewards. Concretely, for each x one draws C trajectories $\{\tau_i\}_{i=1}^{C}$, computes verifiable rewards and a group baseline (e.g., a rank or mean-normalized signal), and updates π_θ with clipped policy ratios as in PPO [41]. GRPO typically improves `pass@1` on math/code under RLVR [43]. However, absent any explicit term for diversity, it can *reduce* variation across attempts as the policy sharpens around locally high-reward regions [9].

As some intuition for this distribution change, suppose that the model is attempting to predict the correct answer in a setting where it believes that the answer is `Yes` with probability 70% and `No` with probability 30%. Cross-entropy loss encourages a model to predict the correct distribution of 70% `Yes` and 30% `No`; on the other hand, GRPO training would cause the model to collapse its output distribution towards predicting 100% `Yes`, as it maximizes the `pass@1`. Empirically, this can shrink the entropy of the completion distribution and heighten redundancy among attempts, limiting `pass@k` improvements even as `pass@1` increases [9].

2.3 Mutual Information and Skill Discovery

Maximizing mutual information (MI) between latent variables and observed behavior has been a recurring tool for learning structured, controllable representations [53, 26, 50].

In generative modeling, InfoGAN [6] augments GAN training with a variational lower bound on $\mathcal{I}(c; x)$ to make latent codes c predictably control semantic factors (e.g., stroke thickness for MNIST). In variational autoencoders, InfoVAE [63] adds an explicit MI term to counteract posterior collapse and preserve informative latents even with expressive decoders.

In sequential decision making, MI has been used to discover diverse, reusable behaviors without external rewards. Early work such as VIC [20] and DIAYN [13] maximizes $\mathcal{I}(s; z)$ or $\mathcal{I}(\tau; z)$, encouraging skills z whose rollouts visit different parts of state or trajectory space and remain identifiable from observations. InfoGAIL [32] extends this to imitation learning by maximizing MI between a latent intention and trajectories to capture multi-modal expert behavior. Subsequent methods bias the MI objective toward long-horizon distinctiveness to avoid trivial short-term variation [45, 22]. Additional related work on MI is available in Appendix B.

Unsupervised skill discovery in RL can be viewed as maximizing the MI between a latent "skill" variable and observed trajectories [20, 13]. Let $z \in \mathcal{Z}$ index a skill and let τ denote a trajectory. These methods maximize

$$\max_{\pi} \mathcal{I}(\tau; z \mid x) \;=\; \mathbb{E}\Big[\log p_\pi(\tau \mid x, z) - \log p_\pi(\tau \mid x)\Big] \;=\; \mathcal{H}(\tau \mid x) - \mathcal{H}(\tau \mid x, z). \tag{3}$$

This decomposition clarifies the pressure on the policy: *(i)* to increase marginal entropy $\mathcal{H}(\tau \mid x)$ so that trajectories cover more of the solution space; and *(ii)* to decrease conditional entropy $\mathcal{H}(\tau \mid x, z)$ so that each z induces a reproducible, stable mode. The net effect is a set of distinct, consistent behaviors indexed by z that together span diverse solution strategies.

Our setting is closest in spirit to unsupervised skill discovery (e.g., DIAYN, VIC) and to mutual information-based skill learning [65], but differs in applying these techniques to language models with RLVR training. We develop an approach for maximizing MI tailored to LLM reasoning, and also connect `pass@k` performance with the mutual information objective.

2.4 Other techniques for response diversification

Beyond MI-based training, several other approaches aim to increase output diversity.

At inference time, decoding-time diversification alters sampling: increasing temperature, switching to nucleus/top-k sampling [24, 15], or perturbing prompts [37]. While simple, these approaches face limitations: *(i)* they often fail to explore qualitatively distinct solution paths [35, 39]; *(ii)* they require domain-specific tuning [59]; and *(iii)* they can trade off against correctness and coherence [35]. Prompt-cycling can inject domain knowledge (e.g., "try algebra" vs. "try geometry"), but it burdens users with prompt engineering and saturates well below human diversity [48].

Determinantal point processes (DPPs) provide another path by rewarding sets of outputs that span high-volume regions in embedding space [30, 56, 34, 58]. In reinforcement learning, determinant-based rewards can encourage agents to explore trajectories that span complementary regions of state space [2, 62]. Compared to MI, which directly couples a latent z with trajectories to ensure reproducible modes, DPP-based diversity is distribution-free: it treats a set of samples as diverse if they occupy a high-volume region in representation space, regardless of whether the same diversity is reproducible under repeated sampling.

Finally, training-time diversification has also been studied through explicit `pass@k`-based objectives. [52] proposed an unbiased estimator for generic k-attempt objectives, showing overall improved

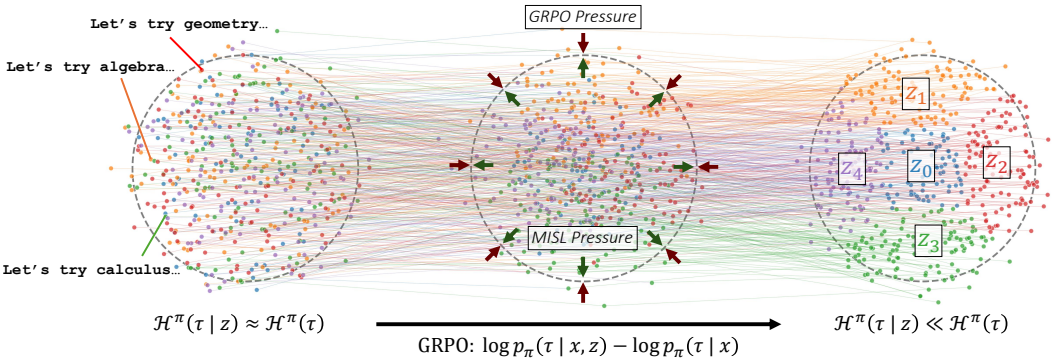

Figure 3: Example illustration of how the MISL reward improves `pass@k` performance. Before MISL (left), the trajectory distribution is independent of the latents z, so the conditional entropy is close to the marginal. MISL training prevents distribution collapse due to `pass@1` training (middle). Adding the token-level MI reward (right) yields well-separated clusters indexed by z, reducing conditional entropy while preserving high marginal entropy. At inference, fixing different z values produces consistent and diverse solution strategies.

model efficacy. Extending this, [7] argue that simply training on `pass@1` falls victim to a local maximum of over-exploitation and reduced exploration. They find that `pass@k` training naturally focusing optimization efforts on harder problems producing significant improvements in both `pass@k` and `pass@1`. Outside of verifiable domains, DivPO [31] alters preference optimization by contrasting diverse high-quality responses with common low-quality ones using a predefined diversity objective, yielding large diversity gains on creative and instruction-following tasks.

As we provide an orthogonal method to improve `pass@k` and diversity, our approach may complement that of [7], [52], and [31].

3 Optimizing LLM Diversity with Mutual Information

Given an input x and an autoregressive policy $\pi(\cdot \mid x)$ that produces a completion (trajectory) $\tau = (y_1, \ldots, y_T)$, we introduce a *discrete* latent $z \in \{1, \ldots, N\}$ via a lightweight prompt prefix (e.g., `Strategy {z} |`), yielding conditional policies $\pi(\cdot \mid x, z)$. During training, z is drawn uniformly at random from the set $\{1, \ldots, N\}$. At inference, one selects $k \leq N$ distinct values of z and generates one completion per value, producing k semantically distinct attempts.

3.1 Objective

We would like to encourage *structured response diversity* by maximizing the conditional mutual information $\mathcal{I}(\tau; z \mid x)$. Intuitively, maximizing mutual information makes the outputs of different strategies distinguishable, ensuring that each z induces a reliably different mode. By querying each strategy once, we obtain k semantically distinct attempts. Formally, this corresponds to maximizing

$$\max_\pi \ \mathcal{I}(\tau; z \mid x) \ = \ \mathbb{E}\Big[\log p_\pi(\tau \mid x, z) - \log p_\pi(\tau \mid x)\Big], \tag{4}$$

which increases the overall entropy of trajectories while reducing the conditional entropy within each z-mode, ensuring diverse yet reproducible strategies. The term $p_\pi(\tau \mid x) = \frac{1}{N}\sum_{z'=1}^{N} p_\pi(\tau \mid x, z')$ is a uniform mixture over skills. Maximizing mutual information encourages *(i)* high marginal entropy of trajectories, promoting broad coverage; and *(ii)* low conditional entropy given z, so that each response is distinct and determined by z. Figure 3 provides an overview of the relevant dynamics.

Algorithm 1 UpSkill: A method for training LLMs to produce diverse responses with mutual information.

Inputs: base policy π_{base}, trainable policy π, latent count N, completions per group C, weights $(\alpha_1, \alpha_2, \beta)$
repeat
  Sample a minibatch of prompts $\{x\}$
  for each x in the minibatch **do**
    Sample $z \sim \text{Unif}(\{1, \ldots, N\})$; generate C completions $\{\tau_i\}_{i=1}^{C}$ with $\pi(\cdot \mid x, z)$
    Compute $r_{\text{corr}}(\tau_i)$, $r_{\text{TMI}}(\tau_i; x, z)$, and $\Delta_{\text{KL}}(\tau_i)$ as above
  end for
  Form per-sample rewards via (6); compute advantages; update π with GRPO
until convergence

3.2 Token-level mutual information reward

We now focus on implementing the mutual information as a token-level reward. For each pair (x, z), let $\{\tau_i\}_{i=1}^{C}$ be C completions sampled from $\pi(\cdot \mid x, z)$. We define a *per-sample* token-level score

$$r_{\text{TMI}}(\tau_i; x, z) \;=\; \sum_{t=1}^{C} \Big[\log p_\pi(y_t \mid x, z, y_{<t}) - \log p_\pi(y_t \mid x, y_{<t}) \Big], \tag{5}$$

where the second term is the uniform mixture

$$p_\pi(y_t \mid x, y_{<t}) \;=\; \frac{1}{N} \sum_{z'=1}^{N} p_\pi(y_t \mid x, z', y_{<t}).$$

Log-probabilities are computed by π on the realized τ_i. In our implementation the mixture is computed *exactly* across all N skills; this is feasible for the N used in our experiments (Section 5). Since $\frac{1}{C} r_{\text{TMI}}(\tau_i; x, z)$ is a Monte Carlo estimator of $\mathcal{I}(\tau; z \mid x)$, we make this our main reward term with UpSkill, with the other reward term being considered in ablation experiments. Appendix D discusses an alternative based on semantic mutual information.

3.3 Combined RL objective

Let $r_{\text{corr}}(\tau_i) \in \mathbb{R}$ denote the verifiable correctness reward (often binary) and define the per-trajectory KL penalty as

$$\Delta_{\text{KL}}(\tau_i) \;=\; \sum_{t=1}^{|\tau_i|} \log \frac{\pi(y_t \mid x, z, y_{<t})}{\pi_{\text{base}}(y_t \mid x, z, y_{<t})}.$$

The per-sample scalar reward is

$$r(\tau_i; x, z) \;=\; r_{\text{corr}}(\tau_i) \;-\; \beta\, \Delta_{\text{KL}}(\tau_i) \;+\; \alpha_1\, r_{\text{TMI}}(\tau_i; x, z), \tag{6}$$

with $\alpha_1, \beta \geq 0$. We apply GRPO to optimize the sum of the combined rewards.

3.4 Training procedure

We fine-tune a trainable policy π_θ with GRPO while injecting a discrete strategy variable $z \in \{1, \ldots, N\}$. At each step, we draw a minibatch of prompts x and, for each x, sample a strategy z uniformly and generate C completions $\tau_{1:C} \sim \pi_\theta(\cdot \mid x, z)$ under fixed decoding. For every completion τ, we compute: *(i)* a verifiable correctness reward $r_{\text{corr}}(\tau)$ from the task's deterministic checker; *(ii)* the token-level MISL term r_{TMI} that measures how specific the trajectory is to the chosen strategy; and *(iii)* a KL control term toward a frozen base policy. We then update the policy with GRPO on this reward.

3.5 Inference

Given a budget of k attempts, we choose k distinct latents from $\{1, \ldots, N\}$ and generate one completion per latent under fixed decoding hyperparameters. Aggregation (e.g., majority vote) can optionally be applied. Because each completion is produced by a trained, distinct mode, conditional success probabilities remain larger than with redundant samplings, improving multi-attempt metrics.

4 Theoretical Connection Between pass@k Improvement and Mutual Information

Our main theoretical result shows that the mutual information objective is closely tied to pass@k. In particular, we will show that the mutual information objective is a lower bound on *improvement* in the pass@k objective, so maximizing mutual information provably results in an increased (lower bound on) pass@k. Our theoretical results will require the following assumptions:

1. k-uniform mixture model: Assume that the marginal distribution over the skills is identical to the base model.
2. Distributional impact: Let a_z be the probability of success of strategy z and a be the probability of success of the base model. Assume that for all $x \in \mathcal{X}$ there exists $\eta > 0$ such that for all $z \in [k]$, $|a_z - a| \geq \eta\delta(\pi_{M,z}(\cdot \mid x), \pi_B(\cdot \mid x))$, where δ is the total variation distance.

The second assumption says that the distribution shifts induced by UpSkill correspond to different problem approaches, and, as a result, will have different probabilities of success. A more precise definition of k-uniform mixture models, additional justification for the assumptions, and the statement and proof of the lemma are in Appendix C.

Lemma 1. Let $\texttt{pass@k}_B$ be the pass@k score of the base model on prompt x and $\texttt{pass@k}_M$ be the pass@k score of the mixture model on prompt x. Under the above assumptions, we show that:

$$1 - \exp\left(-C_1\eta^2\mathcal{I}(\tau; z \mid x)^2\right) \leq \frac{\texttt{pass@k}_M - \texttt{pass@k}_B}{1 - \texttt{pass@k}_B} \leq 1 - \exp\left(-C_2\mathcal{I}(\tau; z \mid x)\right)$$

where C_1 depends on k and C_2 depends on k and $\max_z a_z$.

The quantity in the middle can be interpreted as the fraction of possible improvement from the base model that is realized by the mixture model. Since monotonically increasing functions in $\mathcal{I}(\tau; z \mid x)$ provide both lower and upper bounds on how much the mixture model improves over the base model, it makes sense to optimize directly for the mutual information. UpSkill explicitly increases $\mathcal{I}(\tau; z \mid x)$ during training, ensuring diversity across skills and giving a guaranteed improvement in pass@k over the base model.

5 Experiments

We evaluate whether conditioning on a discrete latent z and training with our token-level mutual-information reward (5) improves multi-attempt metrics. We present results in two settings: (1) a controlled arithmetic environment that allows for fully verifiable evaluation and direct inspection of distributional effects, and (2) the GSM8K benchmark across three open-weight models. We then report ablations on the number of strategies N and on adding a semantic-MI surrogate.

5.1 Arithmetic Environment

The arithmetic environment consists of prompts with three single-digit integers and a latent skill index $z \in \{0, \ldots, N-1\}$. A small transformer model chooses one of the integers to be a target and is required to produce a simple arithmetic expression with the other two digits that evaluates to this target. We use an automatically verified correctness reward, and training uses GRPO without a KL penalty. Full details of the environment, model, and training procedure are provided in Appendix E.

Figure 4 illustrates that in the control condition (α_1=0), training quickly collapses to a single deterministic strategy: by the end of training, pass@1 and pass@5 are indistinguishable (0.793), offering no benefit from multiple attempts. In contrast, with UpSkill (α_1=0.5, MI reward cap at 1.0), the model maintains diverse trajectories, yielding a substantially higher multi-attempt accuracy (pass@5 $= 0.897$) despite a lower single-attempt accuracy (pass@1 $= 0.390$). This difference aligns with the entropy dynamics: UpSkill preserves broad output distributions (token entropy changed from 0.723 to 0.797 over training), sustaining diverse strategies and higher pass@5. By contrast, the control run—while substantially improving pass@1—collapses to near-deterministic outputs (with entropy 0.723 changing to 0.030), leaving pass@5 identical to pass@1 (see Appendix E.5 for more details).

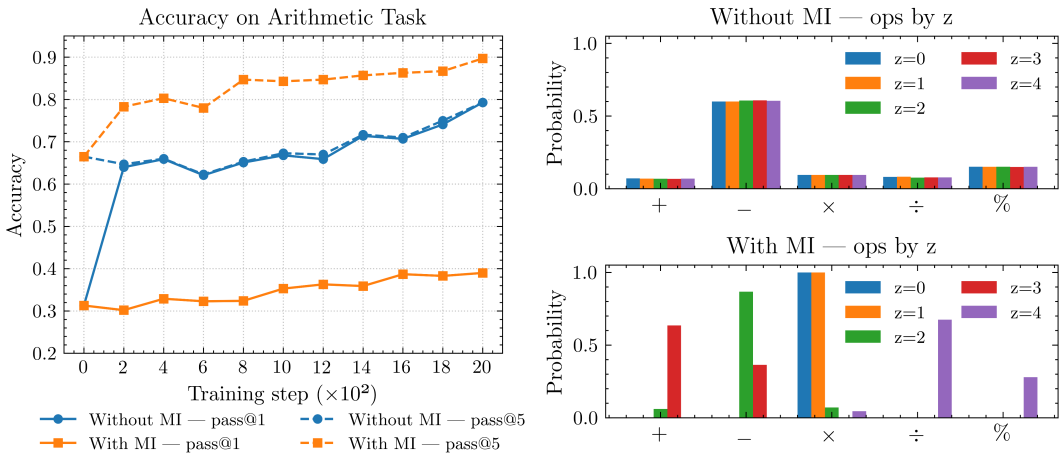

Figure 4: Arithmetic environment results. Training curves show that under GRPO alone (blue), `pass@1` and `pass@5` converge together, indicating that multiple attempts provide little benefit. With MISL (orange; N=5), `pass@5` improves substantially while `pass@1` remains modest, demonstrating that different latents yield complementary solutions. Operator distributions further highlight this effect: without MISL, they are nearly identical across z, reflecting a lack of specialization, whereas with MISL, distinct latents focus on different operators, producing diverse strategies that drive multi-attempt gains.

Figure 4 illustrates that under UpSkill, different z values yield distinct distributions over operators and digits, whereas the control produces nearly identical distributions across z. In this small-scale environment, we can directly observe the learned strategies, and notably $z = 1$ and $z = 2$ converge to risky yet common modes, whereas other values of z cover the remaining operations, improving multi-attempt success with a strategy infeasible for optimizing a `pass@1` objective. The distributions over the first digit are available in Appendix E.7.

We additionally ablate the impact of starting model capabilities, the coefficient of KL penalty, and GRPO parameters (see Appendix F for full details and results). KL penalty β discourages entropy collapse by ensuring the new policy remains close to the initial policy, thereby improving performance. On models with $\beta \in [0.05, 0.10]$, we test $\alpha_1 \in [0.1, 0.3, 0.5]$ and find that well-chosen MI-reward parameters increase `pass@k` by an average of 3% for the weaker base model. However, for the stronger base model, we find the opposite trend. It is always best to choose $\alpha_1 = 0$, with the best choice of $\alpha_1 \in [0.1, 0.3, 0.5]$ still leading to a 1.2% performance decrease. Our theoretical results in Lemma 1 suggest that UpSkill improvement is negatively related to `pass@1` (equivalently $\texttt{pass@k}_B$) capability, and thus, although surprising, this result is in line with our theoretical analysis. We separately conjecture that β, which corresponds with a decrease in exploration from the base policy, conflicts with the mutual information incentive to explore.

5.2 GSM8K

We next evaluate on GSM8K [8], a dataset of grade-school arithmetic word problems. We use 2,000 training problems and a held-out set of 500 questions. All experiments are conducted in a zero-shot setting with a maximum sequence length of 1024 tokens. We train LoRA adapters (approximately 80M trainable parameters) on top of three open-weight backbones: Llama 3.1–8B [18], Qwen 2.5–7B [38], and R1-Distilled–Qwen2.5–Math–1.5B [10]. As before, we apply GRPO with a correctness reward and with default KL penalty, and UpSkill is applied at the token level as in Eq. (6). At inference, we fix k distinct skill indices and generate one completion per skill. More details and ablations are available in Appendices G, H, and I. Figure 5 shows our performance on the withheld evaluation set. Token-level MISL improves results on Qwen 2.5–7B on all metrics. The other tested models are generally unaffected. Interestingly, these results do not entirely parallel the arithmetic environment, as UpSkill's improvement on Qwen 2.5–7B has not come at the cost of `pass@1`. We hypothesize this is due to the existence of a larger set of correct reasoning approaches, and thus MISL does not necessarily come at a cost of `pass@1`. Chen et al. [7] have separately found that `pass@k`

training methods can improve `pass@1` performance. We include summarized outputs from Qwen in Figure 2 and the isolated Qwen results in Figure 1.

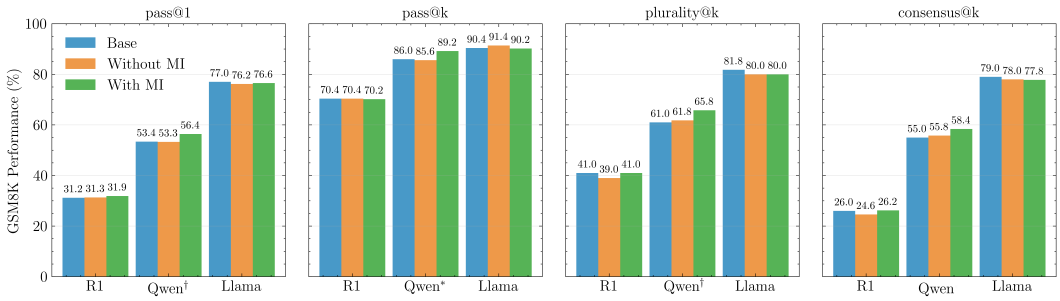

Figure 5: Performance on 500 held-out problems with N=5 strategies. We observe gains on all metrics for the Qwen model. `Base` refers to the model before GRPO training, `Without MI` refers to after GRPO training without token MI, and `With MI` refers to training with GRPO and token MI. An asterisk denotes $p<0.02$ and † denotes $p<0.06$ from McNemar's test on the hypothesis '`With MI`' outperforms '`Without MI`'.

5.3 GSM8K Ablations

We conduct two ablations to probe the robustness of the approach. First, increasing the number of skills beyond N=5 produces mixed results. Many GSM8K problems admit only a limited number of distinct solution paths, so larger N values fragment the capacity into modes that do not translate into additional gains. Second, we study the effect of replacing the token-level MI with a semantic MI, which we formally introduce in Appendix D. This semantic MI occasionally yields further improvements but introduces instability due to estimator variance in high dimensions. We provide more details in Appendix J. Therefore, our main results use token-level MISL only, with semantic variants left as a direction for future work.

6 Conclusion

Our experiments show that UpSkill provides a simple and effective way to induce strategy-level diversity in LLMs, leading to gains on multi-attempt metrics such as `pass@5` and `plurality@5`. By conditioning on discrete latent variables, the model learns reproducible modes of reasoning that reduce redundancy across attempts and increase the likelihood of success. Beyond these empirical findings, UpSkill provides a principled training-time approach for improving response diversity, and our analysis links $\mathcal{I}(\tau; z \mid x)$ to upper bounds in `pass@k` improvement in training. We hope this stimulates research on robust semantic diversity signals and theoretical ties between information-theoretic objectives and multi-attempt success.

Limitations. There are a few notable limitations with our work. Assumption 1 is rather limiting and difficult to enforce empirically, so we hope to find a natural way to either incorporate it into our training method or find another assumption that can also prove Lemma 1. As future work, we hope for further experimentation across domains and with larger models, to better study how our method is affected by KL penalty and base model performance, and to approach the theoretical guarantees from the perspective of Tsallis entropy [17]. The Llama and R1 models had unstable RL training, causing performance to degrade from the base model; we will switch to a more reliable RL framework like verl [46] and use a more robust prompting method than an already semantically meaningful prefix string.

Reproducibility. We make several efforts towards encouraging reproducibility of our work and results. **We have included all experiment code at https://github.com/dshah02/upskill**, along with instructions to reproduce our results. We use only models under permissive licenses. For our theoretical results, we provide a description of assumptions and the complete proof of claims in the appendix. Additionally, for various experiments whose full results could not fit in the main paper, we include the full results in the appendix.

References

[1] Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

[2] Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. Gone Fishing: Neural Active Learning with Fisher Embeddings, December 2021. URL http://arxiv.org/abs/2106.09675. arXiv:2106.09675 [cs].

[3] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An Actor-Critic Algorithm for Sequence Prediction, March 2017. URL http://arxiv.org/abs/1607.07086. arXiv:1607.07086 [cs].

[4] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: Mutual information neural estimation, 2021. URL https://arxiv.org/abs/1801.04062.

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code, July 2021. URL http://arxiv.org/abs/2107.03374. arXiv:2107.03374 [cs].

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets, 2016. URL https://arxiv.org/abs/1606.03657.

[7] Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models, 2025. URL https://arxiv.org/abs/2508.10751.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Xingyu Dang, Christina Baek, J Zico Kolter, and Aditi Raghunathan. Assessing diversity collapse in reasoning. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025. URL https://openreview.net/forum?id=AMiKsHLjQh.

[10] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu,

Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[11] Shihan Dou, Yan Liu, Haoxiang Jia, Limao Xiong, Enyu Zhou, Junjie Shan, Caishuang Huang, Wei Shen, Xiaoran Fan, Zhiheng Xi, Yuhao Zhou, Tao Ji, Rui Zheng, Qi Zhang, Xuanjing Huang, and Tao Gui. StepCoder: Improve Code Generation with Reinforcement Learning from Compiler Feedback, February 2024. URL http://arxiv.org/abs/2402.01391. arXiv:2402.01391 [cs] version: 1.

[12] Weihua Du, Yiming Yang, and Sean Welleck. Optimizing Temperature for Language Models with Multi-Sample Inference, February 2025. URL http://arxiv.org/abs/2502.05234. arXiv:2502.05234 [cs] version: 1.

[13] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is All You Need: Learning Skills without a Reward Function, October 2018. URL http://arxiv.org/abs/1802.06070. arXiv:1802.06070 [cs].

[14] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

[15] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation, 2018. URL https://arxiv.org/abs/1805.04833.

[16] Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1704.03012*, 2017.

[17] Shigeru Furuichi. Information theoretical properties of Tsallis entropies. *Journal of Mathematical Physics*, 47(2):023302, February 2006. ISSN 0022-2488, 1089-7658. doi: 10.1063/1.2165744. URL http://arxiv.org/abs/cond-mat/0405600. arXiv:cond-mat/0405600.

[18] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal

Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra,

Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[19] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

[20] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational Intrinsic Control, November 2016. URL http://arxiv.org/abs/1611.07507. arXiv:1611.07507 [cs].

[21] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[22] Steven Hansen, Guillaume Desjardins, Kate Baumli, David Warde-Farley, Nicolas Heess, Simon Osindero, and Volodymyr Mnih. Entropic Desired Dynamics for Intrinsic Control. In *Advances in Neural Information Processing Systems*, volume 34, pages 11436–11448. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/5f7f02b7e4ade23430f345f954c938c1-Abstract.html.

[23] Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A Sober Look at Progress in Language Model Reasoning: Pitfalls and Paths to Reproducibility, April 2025. URL http://arxiv.org/abs/2504.07086. arXiv:2504.07086 [cs].

[24] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

[25] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer, 2018. URL https://arxiv.org/abs/1808.04339.

[26] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, December 2013. URL http://arxiv.org/abs/1312.6114. arXiv:1312.6114 [stat].

[27] Leslie Kish. *Survey Sampling*. Wiley, 1965.

[28] Peter Krafft. Correlation and mutual information. https://lips.cs.princeton.edu/correlation-and-mutual-information/, February 2013. Laboratory for Intelligent Probabilistic Systems, Princeton University Department of Computer Science.

[29] Alexander Kraskov, Harald Stoegbauer, and Peter Grassberger. Estimating Mutual Information. *Physical Review E*, 69(6):066138, June 2004. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.69.066138. URL http://arxiv.org/abs/cond-mat/0305641. arXiv:cond-mat/0305641.

[30] Alex Kulesza. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012. ISSN 1935-8245. doi: 10.1561/2200000044. URL http://dx.doi.org/10.1561/2200000044.

[31] Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization, 2025. URL https://arxiv.org/abs/2501.18101.

[32] Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations, 2017. URL https://arxiv.org/abs/1703.08840.

[33] Jiate Liu, Yiqin Zhu, Kaiwen Xiao, Qiang Fu, Xiao Han, Wei Yang, and Deheng Ye. RLTF: Reinforcement Learning from Unit Test Feedback, November 2023. URL http://arxiv.org/abs/2307.04349. arXiv:2307.04349 [cs].

[34] Clara Meister, Martina Forster, and Ryan Cotterell. Determinantal Beam Search, June 2023. URL http://arxiv.org/abs/2106.07400. arXiv:2106.07400 [cs].

[35] Minh Nhat Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs, June 2025. URL http://arxiv.org/abs/2407.01082. arXiv:2407.01082 [cs] version: 7.

[36] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, March 2022. URL http://arxiv.org/abs/2203.02155. arXiv:2203.02155 [cs].

[37] Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. Prompt Perturbation Consistency Learning for Robust Language Models. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1357–1370, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.91/.

[38] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[39] Matthew Renze and Erhan Guven. The Effect of Sampling Temperature on Problem Solving in Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7346–7356, 2024. doi: 10.18653/v1/2024.findings-emnlp.432. URL http://arxiv.org/abs/2402.05201. arXiv:2402.05201 [cs].

[40] Igal Sason and Sergio Verdú. f-divergence Inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006, November 2016. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2016.2603151. URL http://arxiv.org/abs/1508.00335. arXiv:1508.00335 [cs].

[41] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[42] Sagi Shaier, Mario Sanz-Guerrero, and Katharina von der Wense. Asking Again and Again: Exploring LLM Robustness to Repeated Questions, March 2025. URL http://arxiv.org/abs/2412.07923. arXiv:2412.07923 [cs].

[43] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[44] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

[45] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-Aware Unsupervised Discovery of Skills, February 2020. URL http://arxiv.org/abs/1907.01657. arXiv:1907.01657 [cs].

[46] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[47] Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. A Thorough Examination of Decoding Methods in the Era of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8601–8629, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.489. URL https://aclanthology.org/2024.emnlp-main.489.

[48] Michal Shur-Ofry, Bar Horowitz-Amsalem, Adir Rahamim, and Yonatan Belinkov. Growing a Tail: Increasing Output Diversity in Large Language Models, November 2024. URL http://arxiv.org/abs/2411.02989. arXiv:2411.02989 [cs].

[49] Greg Ver Steeg. gregversteeg/npeet, May 2025. URL https://github.com/gregversteeg/NPEET. original-date: 2014-10-10T19:57:02Z.

[50] Karl Stratos and Sam Wiseman. Learning Discrete Structured Representations by Adversarially Maximizing Mutual Information, July 2020. URL http://arxiv.org/abs/2004.03991. arXiv:2004.03991 [cs].

[51] Richard S Sutton and Andrew G Barto. Reinforcement Learning: An Introduction. 2015.

[52] Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Rémi Munos. Optimizing language models for inference time objectives using reinforcement learning, 2025. URL https://arxiv.org/abs/2503.19595.

[53] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, April 2000. URL http://arxiv.org/abs/physics/0004057. arXiv:physics/0004057.

[54] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[55] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

[56] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models, October 2018. URL http://arxiv.org/abs/1610.02424. arXiv:1610.02424 [cs].

[57] Eric Wallace, Olivia Watkins, Miles Wang, Kai Chen, and Chris Koch. Estimating Worst-Case Frontier Risks of Open-Weight LLMs, August 2025. URL http://arxiv.org/abs/2508.03153. arXiv:2508.03153 [cs] version: 1.

[58] Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. Diversity Measurement and Subset Selection for Instruction Tuning Datasets, February 2024. URL http://arxiv.org/abs/2402.02318. arXiv:2402.02318 [cs].

[59] Gian Wiher, Clara Meister, and Ryan Cotterell. On Decoding Strategies for Neural Text Generators, March 2022. URL http://arxiv.org/abs/2203.15721. arXiv:2203.15721 [cs].

[60] Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative Preference Learning from Human Feedback: Bridging Theory and Practice for RLHF under KL-Constraint, May 2024. URL http://arxiv.org/abs/2312.11456. arXiv:2312.11456 [cs].

[61] Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. Adversarially regularized autoencoders, 2018. URL https://arxiv.org/abs/1706.04223.

[62] Kaiyan Zhao, Yiming Wang, Yuyang Chen, Xiaoguang Niu, Yan Li, and Leong Hou U. Efficient Diversity-based Experience Replay for Deep Reinforcement Learning, October 2024. URL http://arxiv.org/abs/2410.20487. arXiv:2410.20487 [cs] version: 1.

[63] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders, 2018. URL https://arxiv.org/abs/1706.02262.

[64] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders, 2017. URL https://arxiv.org/abs/1703.10960.

[65] Chongyi Zheng, Jens Tuyls, Joanne Peng, and Benjamin Eysenbach. Can a misl fly? analysis and ingredients for mutual information skill learning. *arXiv preprint arXiv:2412.08021*, 2024.

[66] Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. Dpo meets ppo: Reinforced token optimization for rlhf, 2025. URL https://arxiv.org/abs/2404.18922.

A Use of LLMs

Large language models were used in the preparation of this work for writing assistance (including polishing, improving presentation of concepts, and restructuring of text), for retrieval and discovery of related work, and for support in producing experimental code and figures. Language models were additionally used for feedback on the paper and to formalize mathematical arguments. All analysis, experimental and method design, and final interpretations are our own. LLM outputs were always rigorously reviewed.

B Extended Mutual Information Related Work

Estimating MI reliably is challenging in high dimensions. Variational bounds (Barber–Agakov) optimize a classifier or regressor $q_\phi(z \mid \cdot)$ as a proxy for the intractable posterior [55]. Contrastive bounds such as InfoNCE [55] reduce MI estimation to noise-contrastive classification and have become standard due to their stability. Neural MI estimators like MINE [4] directly optimize a Donsker–Varadhan bound but can suffer from bias/variance trade-offs and training instability. Nonparametric kNN estimators (KSG) [29] avoid parametric critics but require many samples and are sensitive to dimension, motivating careful batching and normalization when used inside policy gradients. In text generation, MI-style objectives have been used to prevent latent collapse and enable controllable generation, e.g., by encouraging informative latents in variational text models [64, 61] or aligning codes with style attributes [25]. These approaches typically maximize MI between prompts or attributes and latent variables, rather than between a discrete strategy and the full trajectory distribution, and are optimized with supervised losses rather than RL.

Conceptually, our objective reconciles two desiderata emphasized in prior MI work: *coverage* (high marginal entropy over trajectories) and *control* (low conditional entropy given z). Whereas decoding-time diversity manipulates token entropy without guarantees about identifiable modes, MI-based diversification learns *reusable, reproducible* modes indexed by a small discrete latent. This makes diversity a first-class, training-time property that can be cleanly exercised at inference by selecting distinct z values.

C Statement and derivation of theoretical bounds

C.1 Problem setup and assumptions

Let $\mathcal{X}$ be the set of all possible prompts. The statement of Lemma 1 applies to the general class of *k-uniform mixture models*.

Definition. A k-uniform mixture model (M, B) is defined to be an ordered pair of a *mixture model*, which is a set of k different policies for generating trajectories, which we will call $\pi_{M,z}(\cdot \mid x)$ for a

prompt $x \in \mathcal{X}$ and strategy $z \in [k]$, along with a *base model* $\pi_B(\cdot \mid x)$ for generating trajectories subject to the condition that

$$\frac{1}{k}\sum_{z=1}^{k} \pi_{M,z}(\cdot \mid x) = \pi_B(\cdot \mid x) \,\forall\, x \in \mathcal{X}.$$

This definition can be interpreted as follows: π_B is the trajectory distribution of the original, non-strategy conditioned language model. If we weigh each strategy as being equally important, we sample once from the mixture model by randomly choosing one strategy. In this case, the joint distribution of trajectories from the mixture is

$$\pi_M(\cdot \mid x) := \frac{1}{k}\sum_{z=1}^{k} \pi_{M,z}(\cdot \mid x).$$

The condition essentially means that the joint distribution of trajectories over picking a strategy uniformly at random must be the same as the original distribution. Therefore, in essence, the mixture model partitions π_B into k different policies that together average back to π_B.

In practice, this condition imposes undue constraints on the strategy distribution, and thus, for the practical implementation, this is not enforced. Also, while in practice one may actually train $N > k$ different strategies and then randomly sample k different strategies so that they still have equal probabilities of being selected, here we make the simplifying assumption that $N = k$, which is true for all of our experiments.

For ease of notation, let $a = \Pr(Y_x(\tau_{B,1}) = 1 \mid x)$ and $a_z = \Pr(Y_x(\tau_{M,z}) = 1)$ for $z \in [k]$. Because the trajectories $\tau_{M,z}$ are sampled independently, we have that

$$\pi_M(\cdot \mid x) = \pi_B(\cdot \mid x) \implies \frac{1}{k}\sum_{z=1}^{k} a_z = a. \tag{7}$$

We provide additional justification for the second assumption made in section 4. First, if the strategies differ in more than just style and contain meaningful semantic differences, we expect that the difference in success should be proportional to how different these two distributions are. The total variation distance measures this distance in trajectory space, while the constant η controls how sensitive the success probabilities are to changes in the distribution shift.

C.2 Extending `pass@k` to k-uniform mixture models

We now extend the traditional definition of `pass@k` to fit the setting of k-uniform mixture models to leverage the fact that we now have a natural structure for querying k different strategies by varying z. This is notably different from the setting in the consistency assumption, which can be interpreted as querying just one strategy uniformly at random.

For a given prompt x and deterministic verifier $Y_x(\tau)$ outputting 1 if τ is a valid output on prompt x and 0 otherwise, and k-uniform mixture model (M, B), define $\texttt{pass@k}_M$ be the probability that querying each of these strategies independently exactly once (see Sec. 3.5) results in at least one correct answer, and define $\texttt{pass@k}_B$ to be the probability that querying π_B independently k times results in at least one correct answer. Writing this out mathematically, for each $z \in [k]$ we sample $\tau_{M,z} \sim \pi_{M,z}(\cdot \mid x)$; then

$$\texttt{pass@k}_M \;=\; 1 - \Pr\left(\bigcap_{z=1}^{k} \{Y_x(\tau_{M,z}) = 0\} \,\middle|\, x\right) = 1 - \prod_{z=1}^{k} \Pr(Y_x(\tau_{M,z}) = 0 \mid x). \tag{8}$$

While we use the same definition of `pass@k` for B as in standard literature, we include it for the sake of completeness; similarly, when sampling $\tau_{B,z} \sim \pi_B(\cdot \mid x)$ independently for each $z \in [k]$, we find that

$$\texttt{pass@k}_B \;=\; 1 - \Pr\left(\bigcap_{z=1}^{k} \{Y_x(\tau_{B,z}) = 0\} \,\middle|\, x\right) = 1 - \Pr(Y_x(\tau_{B,1}) = 0 \mid x)^k \tag{9}$$

where we simplify the product into the RHS of (9) with the independence of samples. Then

$$\texttt{pass@k}_B = 1 - (1-a)^k,\ \texttt{pass@k}_M = 1 - \prod_{z=1}^{k}(1-a_z).$$

We now give the precise statement of the main result.

Lemma 2 (pass@k Improvement for k-uniform Mixture Models, Full Statement)**.** Let $u = \max_{z\in[k]} a_z$ and let $\varphi : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}$ be defined as $\varphi(x) = \frac{x\log x}{x-1}$ for $x \neq 0, 1$ and $\varphi(0) = 0, \varphi(1) = 1$. Then

$$1 - \exp\left(-\frac{k\eta^2\mathcal{I}(\tau; z \mid x)^2}{2\varphi(k)^2}\right) \leq \frac{\texttt{pass@k}_M - \texttt{pass@k}_B}{1 - \texttt{pass@k}_B} \leq 1 - \exp\left(-\frac{k\mathcal{I}(\tau; z \mid x)}{4(1-u)^2}\right).$$

C.3 Derivation of Lower Bound

Using Taylor's Theorem on $f(y) = \log(1-y)$ gives the equations $f(a_z) = f(a) + (a_z - a)f'(a) + \frac{1}{2}(a_z - a)^2 f''(\xi_z)$ where ξ_z lies in between a_z and a, for $z \in [k]$. Summing all of these equations, the linear terms cancel due to (7). Then

$$\sum_{z\in[k]} \log(1-a_z) = k\log(1-a) + \sum_{z\in[k]} \frac{1}{2}(a_z - a)^2 f''(\xi_z). \tag{10}$$

Let A be the random variable that takes value a_z for each $z \in [k]$ with probability $\frac{1}{k}$. Since $f''(y) = -\frac{1}{(1-y)^2}$ is a decreasing function in the interval $[0, 1)$, we find that $f''(\xi_z) \leq f''(0) = -1$ for all $z \in [k]$, i.e.

$$\sum_{z\in[k]} \log(1-a_z) \leq k\log(1-a) - \sum_{z\in[k]} \frac{1}{2}(a_z - a)^2$$

$$\implies k\log(1-a) - \sum_{z\in[k]} \log(1-a_z) \geq \frac{k\operatorname{Var}(A)}{2} \implies \frac{1 - \texttt{pass@k}_B}{1 - \texttt{pass@k}_M} \geq \exp\left(\frac{k}{2}\operatorname{Var}(A)\right)$$

$$\implies \texttt{pass@k}_M - \texttt{pass@k}_B = (1 - \texttt{pass@k}_B)\left(1 - \frac{1 - \texttt{pass@k}_M}{1 - \texttt{pass@k}_B}\right)$$

$$\geq (1 - \texttt{pass@k}_B)\left(1 - \exp\left(-\frac{k\operatorname{Var}(A)}{2}\right)\right).$$

We now find a lower bound for $\operatorname{Var}(A)$ in terms of the mutual information to finish off the proof. Let $\delta(\cdot, \cdot)$ represent the total variation distance between two distributions. We have that $\operatorname{Var}(A) \geq \mathbb{E}[|A|]^2$ by Jensen's Inequality, so using the distributional impact assumption,

$$\operatorname{Var}(A) \geq \left(\mathbb{E}_{z\sim\text{Unif}\{1,2,\ldots,k\}}[|a_z - a|]\right)^2 = \left(\frac{1}{k}\sum_{z=1}^{k}|a_z - a|\right)^2 \geq \left(\frac{1}{k}\sum_{z=1}^{k}\eta\delta(\pi_{M,z}, \pi_B)\right)^2$$

$$= \eta^2\left(\mathbb{E}_{z\sim\text{Unif}\{1,2,\ldots,k\}}[\delta(\pi_{M,z}, \pi_B)]\right)^2.$$

Note that by the assumption that $\pi_M(\cdot \mid x) = \pi_B(\cdot \mid x)$ we have that $0 \leq \pi_{M,z}(\cdot \mid x) \leq k\pi_B(\cdot \mid x)$ for all $z \in [k]$. Therefore,

$$\frac{d\pi_{M,z}}{d\pi_B}(\tau) \in [0, k].$$

Applying Theorem 26 from Sason and Verdú [40] with $\beta_2 = 0, \beta_1 = \frac{1}{k}$ where the constants are from the assumption on bounded likelihood ratio, we have that $D_{\text{KL}}(\pi_{M,z} \parallel \pi_B) \leq \varphi(k)\delta(\pi_{M,z}, \pi_B)$. Summing over $z \in [k]$ and dividing by k yields

$$\mathcal{I}(\tau; z \mid x) = \mathbb{E}_{z\sim\text{Unif}\{1,2,\ldots,k\}}[D_{\text{KL}}(\pi_{M,z} \parallel \pi_B)] \leq \varphi(k)\mathbb{E}_{z\sim\text{Unif}\{1,2,\ldots,k\}}[\delta(\pi_{M,z}, \pi_B)].$$

Putting both bounds together, we finally find that

$$\operatorname{Var}(A) \geq \left(\frac{\eta\mathcal{I}(\tau; z \mid x)}{\varphi(k)}\right)^2$$

$$\implies \texttt{pass@k}_M - \texttt{pass@k}_B \geq (1 - \texttt{pass@k}_B)\left(1 - \exp\left(-\frac{k\eta^2\mathcal{I}(\tau; z \mid x)^2}{2\varphi(k)^2}\right)\right)$$

as desired.

C.4 Derivation of Upper Bound

Let $u = \max_z a_z < 1$. In the interval $[\min(a, a_1, a_2, \ldots, a_k), \max(a, a_1, a_2, \ldots, a_k)]$ f'' achieves its minimum at $f''(u) = -\frac{1}{(1-u)^2}$. Then from (10)

$$\sum_{z\in[k]} \log(1-a_z) \geq k\log(1-a) - \frac{1}{2(1-u)^2}\sum_{z\in[k]}(a_z-a)^2$$

$$\implies \prod_{z=1}^{k}(1-a_z) \geq (1-a)^k \exp\left(-\frac{1}{2(1-u)^2}\sum_{z\in[k]}(a_z-a)^2\right)$$

$$\implies 1-\texttt{pass@k}_M \geq (1-\texttt{pass@k}_B)\exp\left(-\frac{1}{2(1-u)^2}\sum_{z\in[k]}(a_z-a)^2\right). \tag{11}$$

This places an upper bound on how much we can possibly improve `pass@k` compared to our original trajectory distributions. Using Pinsker's Inequality, we find that for all $i \in [k]$,

$$|a_i - a| = |\Pr[Y(\tau_{M,i})=1] - \Pr[Y(\tau_{B,i})=1]| = \left|\sum_{Y(\tau')=1}\pi_{M,i}(\tau') - \sum_{Y(\tau')=1}\pi_{B,i}(\tau')\right|$$

$$\leq \sum_{Y(\tau')=1}|\pi_{M,i}(\tau') - \pi_{B,i}(\tau')| \leq \sum_{\tau'}|\pi_{M,i}(\tau') - \pi_{B,i}(\tau')|$$

$$\leq \delta(\pi_{B,i}, \pi_{M,i}) \leq \sqrt{\frac{1}{2}D_{\mathrm{KL}}(\pi_{M,i} \parallel \pi_{B,i})}$$

$$\implies (a_i-a)^2 \leq \frac{1}{2}D_{\mathrm{KL}}(\pi_{M,i} \parallel \pi_{B,i}) \implies \sum_{z\in[k]}(a_z-a)^2 \leq \frac{k}{2}\cdot\frac{1}{k}\sum_{z\in[k]}D_{\mathrm{KL}}(\pi_{M,i} \parallel \pi_{B,i})$$

$$= \frac{k}{2}\mathbb{E}_{z\sim\mathrm{Unif}\{1,2,\ldots,k\}}[D_{\mathrm{KL}}(\pi_{M,i} \parallel \pi_B)] = \frac{k}{2}\mathcal{I}(\tau; z \mid x).$$

Combining this with (11) yields

$$1-\texttt{pass@k}_M \geq (1-\texttt{pass@k}_B)\exp\left(-\frac{k}{4(1-u)^2}\mathcal{I}(\tau; z \mid x)\right). \tag{12}$$

As a result, if $\mathcal{I}(\tau; z \mid x)$ is too small, then our theoretical upper bound on improvement in `pass@k` between steps 0 and T will also be very small.

Rearranging (12) to bound the improvement $\Delta := \texttt{pass@k}_M - \texttt{pass@k}_B$, we obtain

$$\Delta = (1-\texttt{pass@k}_B) - (1-\texttt{pass@k}_M)$$

$$\leq (1-\texttt{pass@k}_B)\left(1-\exp\left(-\frac{k}{4(1-u)^2}\mathcal{I}(\tau; z \mid x)\right)\right)$$

$$\leq (1-\texttt{pass@k}_B)\cdot\frac{k}{4(1-u)^2}\cdot\mathcal{I}(\tau; z \mid x),$$

where the final inequality uses $1 - e^{-x} \leq x$.

D Semantic mutual information reward

One observation that we made empirically is that token-level differences tend to reflect formatting or paraphrasing, rather than semantically distinct strategies. To bias toward more meaningful differences, we test an alternative method for measuring mutual information by embedding completions with a fixed encoder $\psi(\tau) \in \mathbb{R}^d$ and estimating the mutual information between embeddings and skills for a *single prompt* x:

$$\widehat{\mathcal{I}}(\psi(\tau); z \,|\, x) \tag{13}$$

using the KSG k-nearest-neighbor estimator [29], implemented with the library NPEET [49]. Concretely, for each x we collect the set of embeddings across strategies and samples, $\mathcal{B}(x) = \{(\psi(\tau_i^{(z)}), z) : z \in \{1, \ldots, N\}, i = 1, \ldots, C\}$, and apply KSG to $\mathcal{B}(x)$ to obtain a single scalar $r_{\mathrm{SMI}}(x)$.

E Arithmetic Environment

E.1 Task

Each problem instance consists of three integers $a, b, c \in \{0, \ldots, 9\}$. A small transformer model chooses one of the integers to be a target and is required to produce a simple arithmetic expression with the other two digits that evaluates to this target. Valid operators are $\{+, -, \times, \div, \mathrm{mod}\}$, with division and modulo defined only when results are integers and denominators are nonzero. A latent skill index $z \in \{0, \ldots, N-1\}$ is provided as part of the prompt, conditioning the model on which strategy to adopt.

E.2 Prompt format and conditioning

The input is formatted as

```
[z] a b c
```

where `[z]` encodes the latent skill id and a, b, c are the three digits. The model is required to generate exactly three tokens in the order `(digit, operator, digit)`. This restriction enforces that every completion corresponds to a candidate arithmetic expression of the form $L \, o \, R$ with $L, R \in \{a, b, c\}$. The verifier deterministically evaluates the completion, awarding a reward of 1 if the output matches the designated target and 0 otherwise.

E.3 Evaluation protocol

At inference, we fix $k = \min(N, 5)$ distinct latent skills and generate one completion per skill at a fixed temperature. We then report:

- `pass@1`: the fraction of skills (out of k) that yield a correct completion, i.e. the probability that a single uniformly sampled skill would succeed.
- `pass@k`: the probability that at least one of the k skills yields a correct completion.

This definition differs from conventional `pass@1` (best-of-k) to more closely capture the multi-skill sampling process we target.

E.4 Model and optimization

The policy is a 2-layer causal Transformer (hidden size 128, 4 attention heads, pre-layer normalization, GELU activations). Inputs are embedded with learned token and positional embeddings. The output vocabulary consists of 15 symbols: `0-9`, `+`, `-`, `*`, `/`, `%`. Training uses GRPO updates without a KL penalty, comparing runs with and without the MISL reward. Teacher-forced cross-entropy warmup is applied for 100 steps before switching to RL. Unless otherwise noted: N=5, temperature 0.9, batch size 32 groups, and C=5 completions per update.

E.5 Training Outcomes

Table 1 reports `pass@1`, `pass@5`, and marginal token entropy H_m at the end of the supervised warmup (step 0), mid-training (step 1000), and the final step (step 2000). All runs use $N = 5$ skills and 2000 training steps.

E.6 Sensitivity to α_1 and `cap`

Increasing the clipping parameter `cap` sustains higher entropy but also introduces more variance across runs. Raising α_1 strengthens specialization and increases `pass@5`, though at the expense of `pass@1`. A moderate setting of α_1=0.5 and `cap`= 1.0 provided the most consistent balance in our experiments.

Condition	After Warmup (Step 0)			Step 1000			Step 2000		
	p@1	p@5	H_m	p@1	p@5	H_m	p@1	p@5	H_m
Control – ($\alpha_1 = 0$)	0.313	0.665	0.723	**0.668**	0.673	0.016	**0.793**	0.793	0.013
($\alpha_1 = 0.5$, cap=1.0)	0.313	0.665	0.723	0.353	**0.843**	0.755	0.390	**0.897**	0.768
($\alpha_1 = 0.5$, cap=1.5)	0.313	0.665	0.723	0.338	0.833	0.764	0.399	0.830	**0.852**
($\alpha_1 = 1.0$, cap=1.0)	0.313	0.665	0.723	0.281	0.813	**0.813**	0.373	**0.897**	0.844

Table 1: Arithmetic environment training outcomes. Accuracy is reported as `pass@1` and `pass@5`; H_m is marginal token entropy. MISL prevents entropy collapse and sustains diverse skill-conditioned strategies.

E.7 Additional Distribution Data

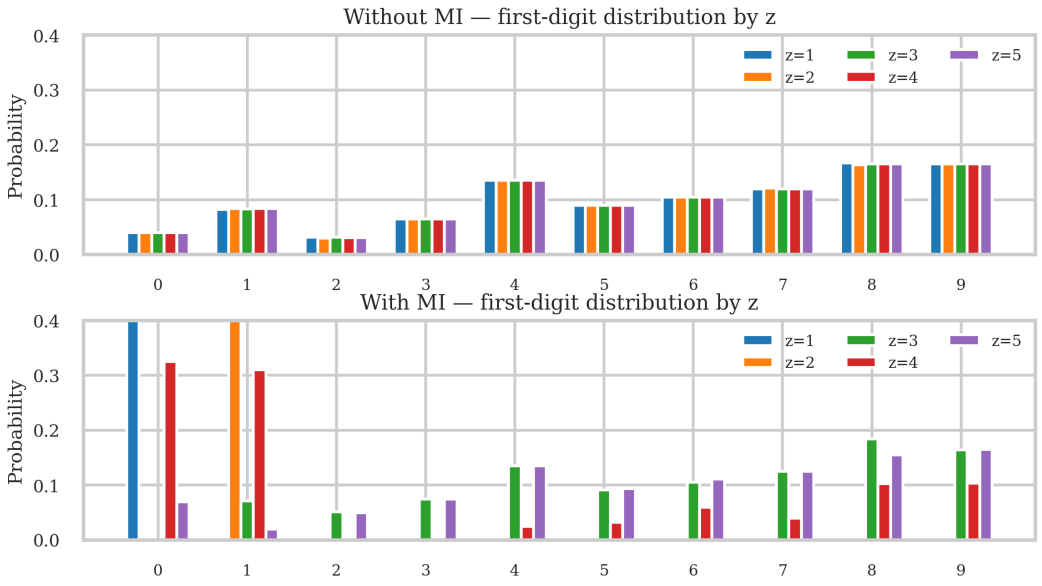

Figure 6: Learned distribution over first response digit with $\alpha_1 = 0.5$ and cap $= 1.0$

F Ablated Arithmetic Environment

Here we ablate *model capacity* by varying the number of warmup steps and by adding a KL penalty. Warmup 50 corresponds to a weaker base model, while warmup 100 produces a stronger base model. This manipulation allows us to study how UpSkill interacts with correctness-oriented pretraining and how much headroom remains for diversity improvements. We also include a KL penalty with coefficients `kl_coef` $\in \{0.05, 0.10\}$, completions per group $C \in \{5, 10\}$, and MIs weights $\alpha_1 \in \{0.0, 0.1, 0.3, 0.5\}$ (token MI only).

Warmup Steps	**pass@1**	**pass@k**
50	0.235	0.540
100	0.313	0.665

Table 2: Performance of the base model after warmup only (no RL). Warmup 50 yields a weaker base capacity, while warmup 100 yields a stronger base capacity.

Tables 3 and 4 report `pass@1` and `pass@5` after RL across settings. Columns `aX.Y` denote $\alpha_1 = X.Y$.

kl_coef	warmup	C	$\mathbf{pass@1_{a0.0}}$	$\mathbf{pass@1_{a0.1}}$	$\mathbf{pass@1_{a0.3}}$	$\mathbf{pass@1_{a0.5}}$
0.050	50	5	0.779	0.780	0.561	0.457
0.050	50	10	0.845	0.817	0.655	0.439
0.100	50	5	0.813	0.801	0.557	0.366
0.100	50	10	0.833	0.860	0.629	0.437
0.050	100	5	0.908	0.897	0.749	0.521
0.050	100	10	0.914	0.897	0.767	0.521
0.100	100	5	0.911	0.895	0.788	0.568
0.100	100	10	0.917	0.901	0.844	0.614

Table 3: `pass@1` after RL with KL penalty in the arithmetic environment.

kl_coef	warmup	C	$\mathbf{pass@5_{a0.0}}$	$\mathbf{pass@5_{a0.1}}$	$\mathbf{pass@5_{a0.3}}$	$\mathbf{pass@5_{a0.5}}$
0.050	50	5	0.793	0.807	0.840	**0.870**
0.050	50	10	**0.867**	0.833	**0.867**	0.817
0.100	50	5	0.840	0.843	0.840	**0.847**
0.100	50	10	0.857	**0.893**	0.850	0.820
0.050	100	5	**0.940**	0.917	0.863	0.883
0.050	100	10	**0.927**	0.903	0.903	0.887
0.100	100	5	**0.927**	0.907	0.910	0.917
0.100	100	10	**0.944**	0.931	0.940	0.923

Table 4: `pass@5` after RL with KL penalty in the arithmetic environment.

Overall, a modest KL penalty (0.05–0.10) prevents entropy collapse and supports higher multi-attempt accuracy, especially when the warmup baseline is stronger (100 vs. 50). With warmup 50, larger α_1 increases `pass@5` substantially (e.g., $0.793 \rightarrow 0.870$), though often at the expense of `pass@1`. With warmup 100, the base capacity is already high, and further MISL gains are more limited, reflecting our theoretical expectation that improvements in `pass@k` depend on the available headroom for diversity.

G GSM8K Experimental Details

G.1 Setup

We evaluate our method on GSM8K [8], a grade-school arithmetic dataset with 2,000 training and 500 held-out evaluation problems. Prompts are provided in a zero-shot format with a maximum sequence length of 1024 tokens. For inference, we fix k distinct latent skill identifiers and sample one completion per skill at fixed temperature.

G.2 Models

We attach LoRA adapters (about 80M parameters) to three open-weight backbones: Llama 3.1–8B, Qwen 2.5–7B, and R1-Distilled–Qwen2.5–Math–1.5B. Training uses GRPO with correctness reward only (control) or correctness and token-level MISL (experimental) with $\alpha_1 = 5.0$ and a learning rate of 5×10^{-6}. Each experiment is run for 2000 steps on a single H100 GPU with 80 GB of memory. For Llama, we adjusted to a learning rate of 1×10^{-6}, and α_1 to 1. Results with and without these changes are included in Appendix H.

G.3 Evaluation Details

We train and evaluate the model with the prompting format of: "`Strategy [z] | Question`" when training with the token MI reward and with only the question otherwise (base model and without MI models). We ablate the prompting format for evaluation in Appendix I. During training and evaluation, we fix the system prompt as "You are a helpful math assistant that solves problems step by step."

In inference, we fix N distinct latent skills and generate one completion per skill, with $N = 5$ except in ablations. To determine correctness, we use a robust extraction function that searches for values within common tags (such as "`<answer> </answer>`"), defaulting to the final word containing digits in the model's output, removes characters not in `0123456789.-`, and compares the resulting value against the reference answer. We then report:

- `pass@1`: the fraction of skills (out of N) that yield a correct completion; i.e., the probability that a single uniformly sampled skill would succeed
- `pass@k`: the probability that at least one of the k skills yields a correct completion
- `plurality@k`: the probability that there is a unique mode response, and that it is correct
- `consensus@k`: the probability that a strict majority of the completions are correct.

H Training Hyperparameter Choice

For the Llama model, we noticed the learning rate of 5×10^{-6} and $\alpha_1 = 5$ led to training instability, and a model that, at intermediate points in training, substantially sacrificed correctness to maximize the token-level mutual information reward. Thus, we changed the learning rate for the Llama model to 1×10^{-6} and the reward coefficient to $\alpha_1 = 1$. The effects of the change on Llama is shown in Figure 7.

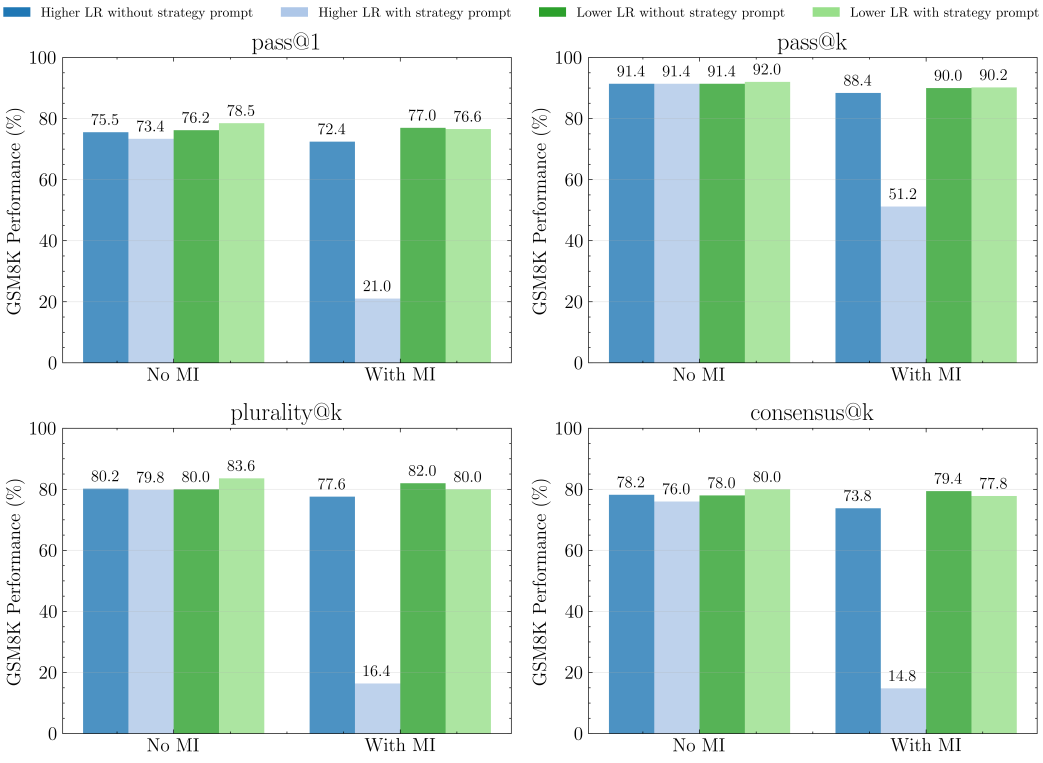

Figure 7: Llama results for learning rate of 1×10^{-6} and $\alpha_1 = 1$ (green bars) compared to 5×10^{-6} and $\alpha_1 = 5$ (blue bars). We simultaneously ablate the result of the strategy prompting, as we do in Figure 8. In each pair of 4 pairs, the leftmost represents the 5×10^{-6} without the strategy prompt, the second furthest to the left represents 5×10^{-6} with the strategy prompt, followed by the models trained with learning rate 1×10^{-6} and evaluated without and with the strategy prompt, respectively.

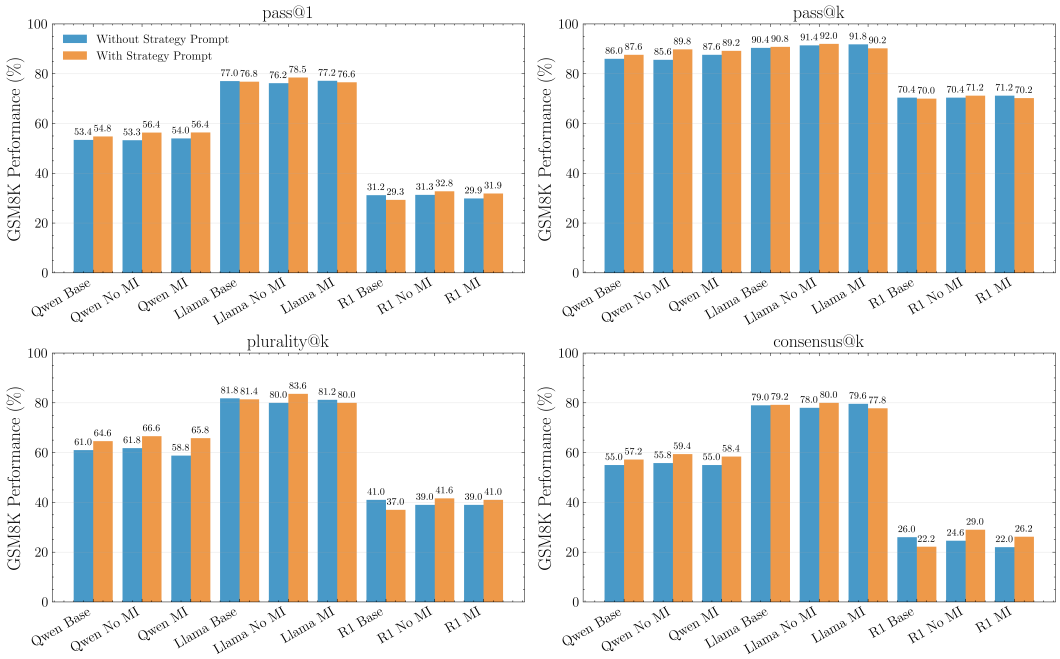

Figure 8: Comparison of open-source models under different prompts during evaluation. Models were trained With MI or Without MI.

I Ablation on Prompting for Evaluations

We ablate the impact of prompting in the format of: "`Strategy [z] | Question`" at the evaluation stage in Figure 8. We test prompting with "`Strategy [z] | Question`" compared to "`Question`" at the evaluation stage to determine if our difference in prompting leads to an advantage. We generally find that some models, including base models and models trained without strategy prompting (Without MI models), still often prefer the strategy prompting, likely as it introduces diversity into the prompt.

J Ablation Studies

J.1 Scaling the number of strategies.

We investigated the effect of increasing the number of latent skills N beyond the default $N{=}5$. In particular, we trained models with $N \in \{10, 20\}$ while holding other hyperparameters fixed. The gains were mixed: although we continued to see improvements in `consensus@k` relative to baselines without MISL, the magnitude of these gains was reduced compared to the $N{=}5$ case, and we did not see a gain in `pass@k` or `plurality@k`. Many GSM8K problems admit only one or two broad solution approaches, so forcing the model to partition its capacity into ten or twenty strategies may lead to fragmentation into modes that were either redundant or unhelpful.

J.2 Semantic MI is promising but unstable.

We also evaluated the addition of a semantic mutual information reward, using the KSG estimator of $\mathcal{I}(\psi(\tau); z \mid x)$ in the embedding space of τ. In principle, this should encourage the learned strategies to differ not only at the token level but also in their higher-level semantics. In practice, however, training with a nonzero α_2 proved challenging. Optimizing against the KSG estimator often destabilized learning, likely due to estimator variance in high dimensions. KSG requires multiple points per class to be well-behaved. In our experiments (Section 5), we use $N = 5$ strategies and $C = 6$ completions per strategy (30 points per x) when the semantic term is enabled; for larger N we

disable the semantic term to avoid excessive compute and estimator variance. We include this test in our ablation.

On R1-Distilled-Qwen, combining semantic MI with token MI occasionally yielded additional improvements, but these were not consistent and were offset by instability in other runs. On Qwen2.5–7B, semantic MI frequently caused degeneracy, with certain strategies collapsing to incoherent or repetitive outputs.

These results suggest that semantic MI is promising but fragile. The occasional positive signal points to the potential of semantic surrogates for enforcing genuine strategy diversity, but the current estimator is too noisy to be reliable in practice. We conjecture that improved semantic objectives could offer a more stable alternative. For the purposes of this work, we therefore report token-MI only as the main result, with semantic MI left for future exploration.

K Full Example Outputs

The following contains the full output of the summarized items in Figure 2:

Strategy 1: Problem: If Raymond does half as much laundry as Sarah, and Sarah does 4 times as much laundry as David, calculate the difference in the amount of laundry Raymond and David do if Sarah does 400 pounds of laundry.

SOLUTION: To find the solution, let's break it down step by step:

1. Sarah does 400 pounds of laundry.
2. David does 1/4 of the laundry Sarah does because Sarah does 4 times as much laundry as David. So David does 400 / 4 = 100 pounds of laundry.
3. Raymond does half as much laundry as Sarah does, so Raymond does 400 / 2 = 200 pounds of laundry.
4. Now, we need to calculate the difference between the amount of laundry Raymond does and David does.
   The difference is 200 - 100 = 100 pounds

The difference in the amount of laundry Raymond and David do is 100 pounds.

Strategy 3: Problem: If Raymond does half as much laundry as Sarah, and Sarah does 4 times as much laundry as David, calculate the difference in the amount of laundry Raymond and David do if Sarah does 400 pounds of laundry.

SOLUTION: To solve this problem, let's break it down step by step.

Given information:

- Raymond does half as much laundry as Sarah.
- Sarah does 4 times as much laundry as David.
- Sarah does 400 pounds of laundry.

Step 1: Determine the amount of laundry David does. Since Sarah does 4 times as much laundry as David, and Sarah does 400 pounds of laundry, we can set up the following equation to find out how much David does:

Let D be the amount of laundry David does. 4D = 400

To find D, we divide both sides by 4.
D = 400 / 4
D = 100

Step 2: Determine the amount of laundry Raymond does.
Since Raymond does half as much laundry as Sarah, and Sarah does 400 pounds of laundry, we can find out how much Raymond does:

Let R be the amount of laundry Raymond does.
R = 400 / 2
R = 200

Step 3: Calculate the difference between the amount of laundry Raymond and David do.
Difference = Amount of laundry Raymond does - Amount of laundry David does
Difference = 200 - 100
Difference = 100

Therefore, the difference in the amount of laundry Raymond and David do is 100 pounds.

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: We claim that we have identified a method to improve multi-attempt performance with a mutual information reward term and demonstrate in the experiments section that our method successfully works.

   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]

   Throughout the paper, we include comments and mentions in directions we believe can be improved. Additionally, in our conclusion and future work sections, we highlight the current limitations of our theory and experimentation.

   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: [Yes]

Justification: We provide a complete and correct proof to justify our theoretical results in section 5. The proofs are included in section 5 and additional details are completed in the appendix.

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: We describe our algorithm completely in section 4 and provide additional practical implementation details in sections 4 and 6. Our code repository will also be open sourced.

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: We leverage open-source models and datasets, and we aim to open-source our code repository, along with the scripts required to rerun our experiments and load both the data and models.

   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: Throughout section 4, we explain all the necessary details to understand the results. Although there are minor choices that are not included in the main paper, these are not necessary to understand the results and will be included in the forthcoming code release.

   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: For our main result in Figure 1 of Section 7, we analyze the statistical significance of our result, showing which improvements satisfy $p < 0.05$ and providing details on the statistical test employed. However, unfortunately, due to the computational cost of model training, we were unable to run multiple training runs per model, so the error within training runs remains not properly understood.

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 specify the computer resources required in section 6.

   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: This paper does not involve any human subjects or sensitive information in datasets. As explained below, it poses minimal societal impact.

   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: [NA]

    Justification: The paper only discusses problems with verifiable rewards, which means that there can only be a right or wrong answer, and furthermore there are straightforward methods to check whether the resulting output is correct or not.

    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: [NA]

    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: In section 6, we detail the licenses for each model and resources, and we properly credit the model and dataset providers. Moreover, we choose models and datasets with well-understood licenses and that are commonly-used in RLVR literature with an aim to ensure further research is accessible. All licenses are properly credited and respected.

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: [NA]

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: [NA]

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: [NA]

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: We have detailed the LLM usage in Appendix A.

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.