ON-POLICY SELF-DISTILLATION WITH SAMPLED DEMONSTRATIONS REDUCES OUTPUT DIVERSITY

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation achieves strong pass@1 accuracy by using a single model as both teacher and student, with the teacher conditioned on a correct demonstration to provide dense token-level feedback. We show that this comes at a hidden cost: rollout diversity decreases and pass@k curves flatten (i.e., generating more rollouts fails to improve accuracy). We trace this to compounding biases in the design of self-distillation with sampled demonstrations. The teacher scores each student rollout while conditioned on a sampled correct rollout, channeling its feedback through the model's own biases. We theoretically analyze the optimal self-distillation policy and show that it tilts the base distribution by a pointwise conditional mutual information score between the student's rollout and the correct rollout used as context. Unlike the ideal optimal on-policy reinforcement learning (RL), which preserves probability ratios among equally correct rollouts, self-distillation can amplify existing probability gaps, concentrating mass on already-dominant modes. On a controlled graph path-finding task and science question-answering benchmarks, self-distilled models match or exceed RL on average performance but exhibit substantially lower *functional* and *semantic* diversity, failing on out-of-distribution settings that require diverse strategies.

1 INTRODUCTION

Current LLM post-training approaches to instill capabilities in models have different tradeoffs, with supervised fine-tuning (SFT) learning initial behaviors and on-policy RL methods refining and exploring new approaches (Zhang et al., 2025). In between, on-policy distillation (Agarwal et al., 2024; Lu & Thinking Machines, 2025) uses a stronger teacher to guide a student using student-generated data. Self-distillation goes further by eliminating the external teacher entirely: the same model, conditioned on privileged information, such as a correct solution or environmental feedback, provides dense token-level feedback on the student's own generations. Recent methods, including SDPO (Hübotter et al., 2026), SDFT (Shenfeld et al., 2026), OPSD (Zhao et al., 2026; Penaloza et al., 2026), and OPCD (Ye et al., 2026), instantiate this approach, achieving strong performance across several tasks such as scientific question-answering, continual learning tasks, and agentic tasks.

In this paper, we investigate a specific case, Self-Distillation with Sampled Demonstrations (SDSD) where student rollouts are guided by a teacher with demonstrations in its context. The demonstrations could come from correct student rollouts (exactly the setup of Hübotter et al. (2026) with student demonstrations) or from external models (Zhao et al., 2026; Penaloza et al., 2026). Here, we find that good accuracy might come at a hidden cost. SDSD models with sampled demonstrations exhibit pass@k curves with small or nearly flat slopes (Figures 3, 4): generating more rollouts fails to solve new problems. By contrast, models trained with on-policy RL (e.g., GRPO) show steep pass@k improvement, where each additional sample meaningfully increases problem coverage. SDSD could thus trade rollout diversity for average accuracy.

We argue that SDSD introduces compounding sources of bias that can progressively reduce rollout diversity. Intuitively, a rollout is *aligned* with a demonstration when the two share more structural or stylistic features, causing the teacher conditioned on that demonstration to assign it higher probability and therefore reinforce it more strongly during training. This creates a bias toward solutions that resemble the sampled demonstrations. In particular, a teacher may struggle to effectively guide a correct but less typical rollout when conditioned on a more standard or canonical demonstration,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

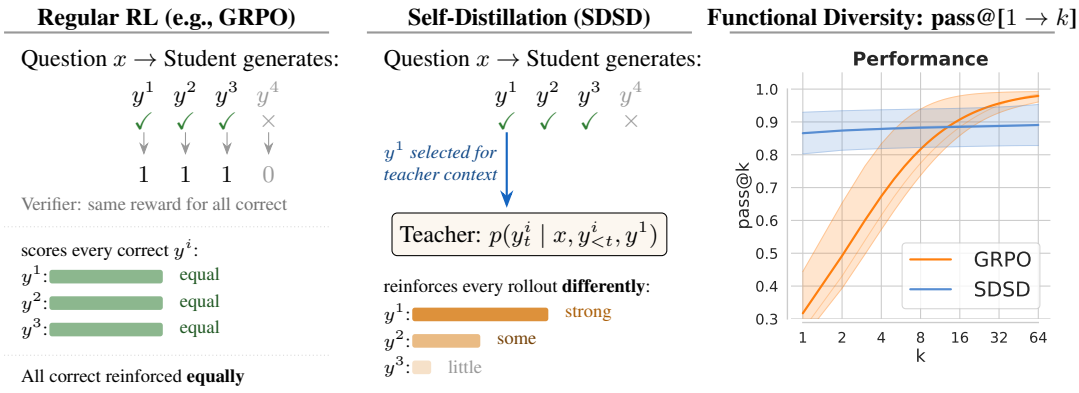


Figure 1: RL and self-distillation treat correct rollouts differently, with consequences for rollout diversity. **Left:** A binary verifier gives equal reward to all correct rollouts, so RL reinforces them uniformly. **Middle:** Self-distillation conditions the teacher on a sampled correct rollout, typically the most probable one, so the teacher's feedback is strongest for similar rollouts and weakest for those taking a different approach. **Right:** On a Graph Path finding task, this gap manifests as flatter pass@k curves for Self-Distillation with Sampled Demonstrations (SDSD): generating more rollouts fails to solve new problems, unlike GRPO where each additional sample meaningfully increases coverage.

simply because the two trajectories share fewer common patterns. As a result, distinctive yet valid solution strategies receive weaker learning signals. Over repeated training updates, this preference can compound: rollouts that are more aligned with previously sampled demonstrations become increasingly reinforced, while less aligned, but still correct, solutions are gradually suppressed (See Fig. 1). We hypothesize that this feedback loop contributes to the reduced rollout diversity and flattened pass@k scaling observed in SDSD models.

We formalize this effect by deriving the optimal self-distillation policy (Proposition 2). The resulting policy is a tilted version of the base distribution, where the tilt is determined by the expected *pointwise conditional mutual information* (PCMI) between a student rollout and the sampled demonstration. PCMI measures how much conditioning on a demonstration increases the model's preference for a particular rollout. Unlike a binary reward, which treats all correct rollouts equally, PCMI distinguishes among equally valid solutions and assigns greater weight to rollouts that are already more compatible with the demonstrations and the base policy. Consequently, self-distillation with sampled demonstrations can amplify existing probability imbalances: likely rollouts become increasingly likely, while less common but correct solutions are progressively suppressed, reducing rollout diversity.

To diagnose the reduced diversity, we use two notions of diversity beyond the commonly-used token-level entropy. *Functional diversity* is the rate at which additional samples solve new problems, reflected in the slope of pass@k curves. *Semantic diversity* measures whether rollouts differ in their high-level strategy (e.g., different paths through a graph, different proof approaches in math) rather than just surface-level wording. We show that token-level entropy fails to capture either notion (§A.3).

Our contributions are:

- We prove that the optimal policy of self-distillation with sampled demonstrations tilts the base distribution by expected PCMI rather than reward, and that this can amplify probability gaps among equally correct rollouts, a property absent from standard RL (Prop. 2, Remark 1).
- We introduce a graph path-finding task, in which *semantic* diversity, the number of distinct concept categories explored, is precisely measurable and directly predicts out-of-distribution generalization (§4.1, Fig. 3).
- On graph path-finding, science QA, coding tasks (Feng et al., 2024), we show that self-distillation with sampled demonstrations achieves competitive pass@1 but substantially lower *functional* and *semantic* diversity than RL, failing on out-of-distribution tasks that require diverse strategies (Fig. 3, 4, Tab.5b).

2 SELF-DISTILLATION WITH SAMPLED DEMONSTRATIONS

We review the self-distillation with sampled demonstration framework of Hübotter et al. (2026), which uses a correctly verified rollout as privileged context for the teacher. For each question x, the student policy generates a group of N rollouts $y_n \sim \pi_\theta(\cdot \mid x)$, $\mathcal{Y}(x) = \{y_1, \dots, y_N\}$.

Let $\mathrm{C}(x) \subseteq \mathcal{Y}(x)$ denote the subset verified as correct. For each rollout $y \in \mathcal{Y}(x)$, a correct rollout $y^{\mathrm{corr}} \in \mathrm{C}(x)$ is sampled uniformly from this subset and provided to the teacher as context.

The teacher is an exponential moving average (EMA) of the student with parameters $\bar{\theta}$, conditioned on the question x and the correct rollout y^{corr}: $\pi_{\bar{\theta}}(\cdot \mid x, y^{\mathrm{corr}}, y_{<t})$. The training objective minimizes the token-level KL divergence between the student and this context-conditioned teacher:

$$\mathcal{L}_{\mathrm{SD}}(\theta; x) = \frac{1}{N} \sum_{y \in \mathcal{Y}(x)} \mathbb{E}_{y^{\mathrm{corr}} \sim \mathrm{C}(x)} \left[\sum_{t=1}^{|y|} \mathrm{KL}(\pi_\theta(\cdot \mid x, y_{<t}) \,\|\, \mathrm{sg}[\pi_{\bar{\theta}}(\cdot \mid x, y^{\mathrm{corr}}, y_{<t})]) \right],$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient. The corresponding token-level gradient (Hübotter et al., 2026) is:

$$\nabla_\theta \mathcal{L}_{\mathrm{SD}}(\theta; x) = \frac{1}{N} \sum_{y \in \mathcal{Y}(x)} \mathbb{E}_{y^{\mathrm{corr}}} \left[\sum_{t=1}^{|y|} \mathbb{E}_{\hat{y}_t \sim \pi_\theta} \left[\log \frac{\pi_\theta(\hat{y}_t \mid x, y_{<t})}{\pi_{\bar{\theta}}(\hat{y}_t \mid x, y^{\mathrm{corr}}, y_{<t})} \, \nabla_\theta \log \pi_\theta(\hat{y}_t \mid x, y_{<t}) \right] \right].$$

Self-distillation thus replaces a single scalar reward for the full sequence with a dense per-token correction signal derived from a context-conditioned version of the model itself.

On-policy RL is mode-seeking, SDSD is even more. Any on-policy method exhibits mode-seeking behavior concentrating its rollouts into a smaller subset (Tajwar et al., 2024; Chen et al., 2025a). SDSD as a reverse KL objective (Bishop, 2006) has the same behavior but even more pronounced due to *additional compounding biases*. In our setting, self-distillation involves two crucial samplings: that of the student rollout and that of the demonstration (either from the student's correct rollouts or from an external source). The *alignment between these two* introduces a bias towards common responses. Consider how two student rollouts with equal reward, one common and one highly novel, are treated. It is more likely to sample a demonstration that resembles the common rollout. Then, the teacher has a bias to give more probability to rollouts that are similar to its context, meaning similar to the demonstration. Together this will lead to the common rollout being upweighted more than the unique one. This leads to a rich-get-richer (likely-get-likelier) behavior stronger than in standard RL, arising from both double sampling and its alignment and the preferences of the teacher.

Moreover, while all on-policy methods exhibit mode-seeking behavior due to *optimization*, self-distillation uniquely introduces incentives for loss of diversity even at the level of the *optimal policy.*

3 OPTIMAL POLICY OF SELF-DISTILLATION

We derive the optimal self-distillation policy and characterize how it differs from standard RL. For ease of presentation, we use a sequence level objective first, and refer to the Appendix B for full derivations and token-level presentation section B.3. Let x denote the input prompt, y an output sequence, and $\pi_0(y \mid x)$ the base policy. We optimize a student policy $\pi(y \mid x)$. All distributions are assumed strictly positive on their support.

Proposition 1 (Optimal policy for standard KL-regularized RL)**.** *The standard RL objective is:*

$$\max_{\pi} \; \mathbb{E}_{y \sim \pi(\cdot \mid x)} \left[R(y \mid x) \right] - \beta_{\mathrm{RL}} \, D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x)) \,. \tag{1}$$

It is well known (Korbak et al., 2022; Rafailov et al., 2023; GX-Chen et al., 2026) that the optimal policy of this objective is the following tilted distribution:

$$\pi^*_{\mathrm{RL}}(y \mid x) \propto \pi_0(y \mid x) \exp\left(\frac{1}{\beta_{\mathrm{RL}}} R(y \mid x) \right). \tag{2}$$

The optimal RL policy consists of the base policy modulated by the reward. Thus, two rollouts with the same probability under the base and the same reward will be *as likely* under the optimal policy.

We now derive the analogous result for self-distillation. We take the teacher to be the fixed base policy conditioned on a correct demonstration, and optimize the student to minimize the reverse KL with the teacher, regularized by a KL penalty to the base model.

In practice, explicit KL regularization to the base policy is not always used in RL or self-distillation. However, since training starts from the base policy and models are rarely trained to convergence, the policy typically remains close in KL to its starting point, making this a reasonable modeling choice.

Let $p_{\mathrm{corr}}(\cdot \mid x)$ denote a reference distribution over correct demonstrations for input x, from which y^{corr} is sampled. In practice, this can be the empirical distribution over $\mathrm{C}(x)$, the student's own correct rollouts (§2), or a distribution over external demonstrations (§A.2).

Let us define the following objective that takes the expectation over demonstrations.

Proposition 2 (Optimal policy for SDSD-KL)**.** *Let the self-distillation + KL objective:*

$$\min_{\pi} \; \mathbb{E}_{y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x)} \Big[D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x, y^{\mathrm{corr}})) \Big] + \beta \, D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x)) \,. \tag{3}$$

The optimal policy of equation 3 is

$$\pi^*_{\text{SD-KL}}(y \mid x) \propto \pi_0(y \mid x) \exp\left(\frac{1}{1+\beta} \, \mathbb{E}_{y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x)} \big[i(y; y^{\mathrm{corr}} \mid x) \big] \right), \tag{4}$$

where the pointwise conditional mutual information (PCMI) is defined as:

$$i(y; y^{\mathrm{corr}} \mid x) := \log \frac{\pi_0(y \mid x, y^{\mathrm{corr}})}{\pi_0(y \mid x)}. \tag{5}$$

The RL optimal policy equation 2 tilts the base policy by the reward. The self-distillation optimum instead tilts by the expected PCMI equation 5: a log-ratio measuring how much more likely the teacher finds y after conditioning on a demonstration. When y^{corr} is relevant and supports y, the PCMI is positive; when y^{corr} is contradictory to y, it is negative. The base policy is thus tilted not by task reward, but by the teacher's assessment of how well each rollout aligns with correct demonstrations.

Remark 1 (Ratio for two correct sequences under SDSD-KL)**.** *Let y_1 and y_2 be two correct sequences for the same input x, and suppose $\pi_0(y_1 \mid x) = k \, \pi_0(y_2 \mid x)$ for some $k \geq 1$. Then*

$$\frac{\pi^*_{\text{SD-KL}}(y_1 \mid x)}{\pi^*_{\text{SD-KL}}(y_2 \mid x)} = k \, \exp\left(\frac{1}{1+\beta} \mathbb{E}_{y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x)} \big[i(y_1; y^{\mathrm{corr}} \mid x) - i(y_2; y^{\mathrm{corr}} \mid x) \big] \right). \tag{6}$$

When does sharpening occur? SDSD-KL preserves the base-policy ratio k only when the two sequences have the same expected PCMI. If the demonstrations y^{corr} support on average y_1 more than y_2, the ratio between the two rollouts becomes even larger under self-distillation, leading to sharpening where likely rollouts become even more likely. This contrasts with the RL optimal policy equation 2, which maintains the initial ratio of k when both rollouts are equally correct, since the reward tilt cancels out. This shows sharpening occurs when the expected PCMI is higher for the already-probable rollout. Such rollouts are more likely to be aligned to the demonstrations, and the teacher shares the same biases as the student, thus this becomes highly likely. The same derivations and implications carry over to the token-level objective, resulting in a bias for sharpening the distribution of the next-token, leading to loss of diversity of the whole rollout (see Appendix B.3).

All on-policy learning methods, like GRPO or self-distillation have mode-seeking behavior due to *optimization*, but the previous remark shows that self-distillation has an *optimal* policy that can be sharper than the initial one.

3.1 Illustrative Example: Mode Collapse Under Self-Distillation

We verify the sharpening predicted by our theory in a minimal controlled environment. The action space is $D{=}100$ discrete actions split into four quarters: the first and third quarters are rewarded, while the other two are not. An ideally diverse policy would place mass on both rewarded modes.

We parameterize the student policy π_θ over a four-bump base distribution. The teacher for SDSD is constructed as $\pi_{\bar{\theta}}(y \mid y^{\mathrm{corr}}) \propto \pi_\theta(y) \cdot K(y, y^{\mathrm{corr}})$, where K is a Gaussian kernel centered on

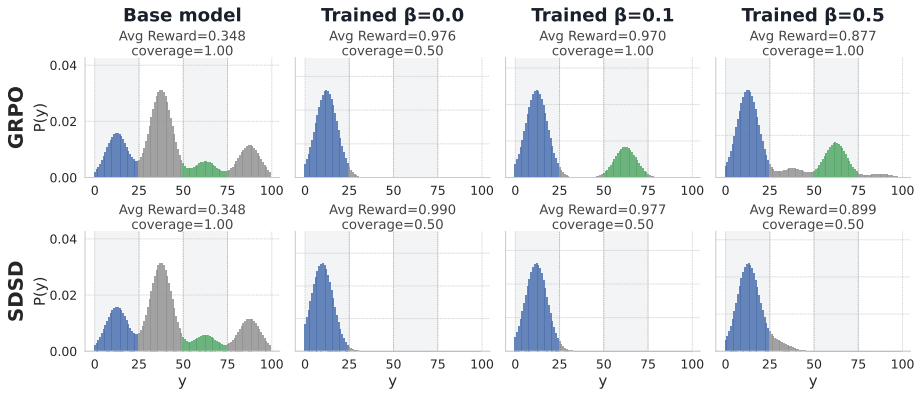


Figure 2: Illustrative example: SDSD collapses to a single high-reward mode, regardless of KL regularization, while GRPO with KL regularization > 0 covers both modes. The environment has two equally-valued reward regions, so an ideal policy would maintain coverage of both.

a correct sample y^{corr} drawn from the student's own correct outputs. This locally upweights the student's mass near each observed correct sample, exactly the mechanism that produces PCMI sharpening in our theory.

Figure 2 shows the result. GRPO recovers both rewarded modes for any $\beta > 0$. SDSD collapses to whichever rewarded region the base policy slightly favors and stays there for every β: the more probable region produces more correct samples, those samples become teacher contexts, and the teacher's kernel-shaped feedback reinforces nearby points, a self-reinforcing loop that no level of KL regularization to the base undoes.

4 Experiments

4.1 Concept-Graph Setting: Loss of Semantic Diversity and OOD Performance

We design a controlled setting, challenging for LLMs, with a precise definition of *semantic diversity* and a direct link between diversity and downstream performance.

Controlled experimental setting. We introduce a graph path-finding task, where we generate multiple graphs and create a query for each one. For each query, we prompt an LLM with a representation of the graph in context and ask it to generate a path between two points. See section B.4 for an example of such query. A graph node represents an instance of a named concept (e.g., specific birds: heron, pigeon; fruits: orange, cherry) as seen in Fig. 5a. The graph has a star structure: a central start node connects to multiple *concept chains*, each consisting of nodes from the same concept (e.g., all birds or all fruits), each ending at a shape node (e.g., diamond, square). Two of the endpoints have the same name and represent the target; the remaining two are distractors. Multiple valid paths to the targets exist, each passing through a different concept chain. Training graphs are biased: some paths to the target are short (11 nodes, easier), while others are long (15 nodes, harder), creating an incentive for models to exploit easy routes and ignore harder but equally valid alternatives.

To check the robustness of the learned models, we evaluate on an *in-distribution* test set and two out-of-distribution datasets: a *larger-graphs* dataset, in which all paths to end nodes have a fixed length of 20, and a *harder-graphs* dataset, in which all paths have fixed length of 11, but one edge is removed from chains leading to one of the two target nodes, giving fewer valid solutions.

Baselines. We train Qwen3-1.7B (Yang et al., 2025) models using variants of GRPO and SDSD, on a dataset of 16k graphs for training and 128 for testing. We train for 1000 steps, with a mini-batch size of 16 queries, 4 rollouts per query, and a maximum generation length of 8,192 tokens. The experiments are trained on a single GPU using the library of Kazemnejad et al. (2025).

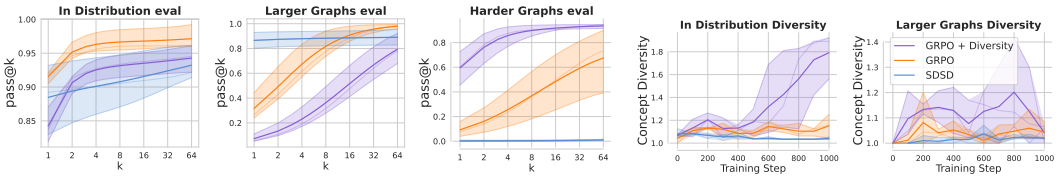


Figure 3: In-distribution and Larger Graphs evaluations show SDSD has good pass@1 performance but, toward pass@k, the curve has a small slope highlighting low *functional diversity*. The third setup requires the model to have learned diverse rollouts during training, and is completely unsolved by self-distillation. Additionally, the last two figures show that the explicitly defined *semantic diversity* of SDSD is the lowest. All runs train Qwen3-1.7B; mean and min/max runs with 3 seeds are shown.

GRPO. Standard GRPO (Shao et al., 2024) serves as the primary baseline. At each iteration, the policy model generates N=4 rollouts per query. Within each group of N rollouts, a scalar reward is used to compute a scalar advantage for the whole sequence.

GRPO+diversity. GRPO+diversity adds a diversity reward to the score reward to encourage the model to explore different concept chains across its N rollouts. For each rollout, its diversity score is the fraction of the other $N-1$ rollouts in the same group that used a disjoint set of concepts. A rollout using different concepts than all the other rollouts of the same query, will have a diversity score of 1. This relates to Li et al. (2025), who multiply the score reward by a clustering-based diversity score.

SDSD. The implementation follows SDPO (Hübotter et al., 2026), where for each student rollout, the teacher (EMA of student) is conditioned on another correct student rollout for the same query.

All models add to the main loss a KL regularization to the reference model.

The results in Fig. 3 show that SDSD achieves good in-distribution pass@1 and the best pass@1 on the Larger Graph setting. However, its pass@k performance increases very slowly, or not at all for the harder settings. This flat pass@k curve indicates low *functional diversity*: successive samples rarely solve new queries. The Harder Graph setting can only be solved by models that learned diverse rollouts during training. SDSD's failure there confirms that it relied exclusively on easy routes.

Semantic diversity. We seek rollouts with *semantic diversity*, capturing meaningful variations in their trajectories, such as different high-level strategies or approaches. In mathematical reasoning, for instance, one might want geometric vs. algebraic approaches, or different theorems. In the graph setting, we define the *semantic diversity* of a set of rollouts as the number of unique concepts present across all of them. This measures whether a model explores fundamentally different strategies (e.g., following animal chains vs. flower chains) rather than mere surface-level token variation. For each query in the in-distribution and larger graphs testsets, we sample 64 rollouts and compute the average number of unique concepts of the nodes present in them, and show these diversity scores across training in Fig. 3. We find that GRPO has relatively low semantic diversity, but adding a diversity reward significantly improves it. On the other hand, SDSD has the lowest semantic diversity scores, which are correlated with the low functional diversity (slope of pass@k) and low scores in the Harder Graphs dataset that requires diversity.

4.2 Science QA: Functional Diversity in Practice

We highlight the interplay of accuracy and diversity of SDSD and GRPO in science QA settings.

Setup. We evaluate on four verifiable reasoning tasks spanning scientific knowledge drawn from SciKnowEval (Feng et al., 2024), a benchmark of multiple-choice science questions. Across all tasks, we train on the training split and evaluate on the held-out test split by generating N=16 rollouts per question and reporting mean accuracy (pass@1) as well as pass@k.

All experiments follow the implementation and configuration of Hübotter et al. (2026), and use Qwen3-8B (Yang et al., 2025) and Olmo-3-7B-Instruct (OLMo et al., 2025) as the base models,

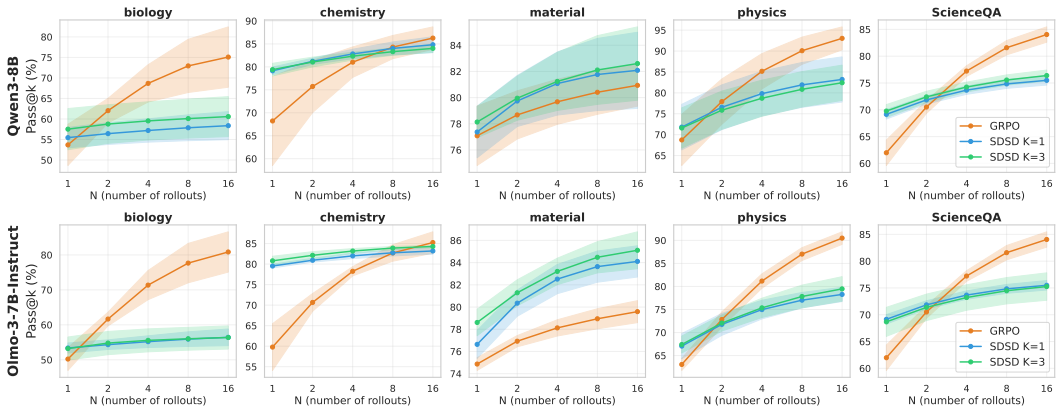


Figure 4: Pass@k curves for Science QA (4 tasks and average across tasks). SDSD achieves better pass@1 but its curves flatten quickly, indicating low functional diversity. Mean ± stderr over 3 seeds.

trained with AdamW (Loshchilov & Hutter, 2019) for up to 30 epochs bounded by a 5h training time budget. At each training step, the policy generates N=8 rollouts per question for a batch of 32 questions, sampled with temperature 1.0. Each configuration is run with 3 seeds on 4 Nvidia H200 GPUs. We compare the following models:

GRPO. Standard GRPO generating N=8 rollouts per question for a batch of 32 questions. The batch is optimized in mini-batches of 8 questions.

SDSD ($K = 1$). SDPO-style self-distillation (Hübotter et al., 2026) where the teacher is an EMA of the student that is conditioned on one correct student rollout.

SDSD ($K = 3$). We introduce a baseline using an ensemble of $K = 3$ teachers. Everything is the same as above, but we collect three distinct correct demonstrations (if available) and condition the same EMA teacher on each one, then average their distillation losses per student rollout. This requires K forward passes through the teacher, though it only adds 5.8% wall-clock time per training step since generating the student rollouts dominates computation.

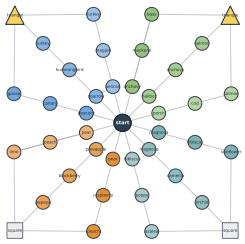
Fig. 4 shows SDSD variants have flatter pass@k curves than GRPO, indicating lower functional diversity. Table 1 reports mean accuracy across 3 seeds, using the best checkpoint per method by average accuracy. SDSD variants attain higher pass@1 but lower pass@k throughout training, consistent with reduced diversity.

Table 1: Pass@1 and Pass@16 at the best checkpoint per dataset, for Qwen3-8B and Olmo-3-7B-Instruct. Overall both SDSD variants have better Pass@1 but worse Pass@16, indicating low diversity between rollouts. Mean ± std across 3 seeds. **Bold** = best per column.

Method	Biology		Chemistry		Material		Physics		Average	
	Pass@1	Pass@16	Pass@1	Pass@16	Pass@1	Pass@16	Pass@1	Pass@16	Pass@1	Pass@16
Qwen3-8B										
GRPO	$57.2_{\pm 3.9}$	$\mathbf{69.3_{\pm 2.9}}$	$76.6_{\pm 3.6}$	$\mathbf{87.3_{\pm 1.7}}$	$\mathbf{79.3_{\pm 0.4}}$	$\mathbf{82.3_{\pm 0.7}}$	$74.4_{\pm 2.9}$	$\mathbf{95.6_{\pm 0.5}}$	$71.9_{\pm 2.0}$	$\mathbf{83.6_{\pm 1.1}}$
SDSD K=1	$57.5_{\pm 1.2}$	$60.3_{\pm 3.2}$	$\mathbf{78.8_{\pm 0.9}}$	$86.5_{\pm 0.1}$	$78.1_{\pm 2.3}$	$80.1_{\pm 2.6}$	$\mathbf{76.6_{\pm 2.6}}$	$88.1_{\pm 2.7}$	$72.7_{\pm 1.1}$	$78.7_{\pm 1.2}$
SDSD K=3	$\mathbf{61.8_{\pm 1.5}}$	$64.8_{\pm 1.0}$	$78.5_{\pm 0.4}$	$83.9_{\pm 1.1}$	$77.7_{\pm 1.0}$	$80.1_{\pm 0.6}$	$75.8_{\pm 2.8}$	$85.4_{\pm 3.3}$	$\mathbf{73.4_{\pm 0.8}}$	$78.5_{\pm 0.6}$
Olmo-3-7B-Instruct										
GRPO	$50.2_{\pm 3.2}$	$\mathbf{80.9_{\pm 5.8}}$	$59.8_{\pm 5.7}$	$\mathbf{85.2_{\pm 2.6}}$	$74.9_{\pm 0.6}$	$79.6_{\pm 1.0}$	$63.1_{\pm 1.3}$	$\mathbf{90.5_{\pm 1.4}}$	$62.0_{\pm 2.4}$	$\mathbf{84.0_{\pm 1.4}}$
SDSD K=1	$\mathbf{53.4_{\pm 1.4}}$	$56.4_{\pm 2.4}$	$79.5_{\pm 0.5}$	$83.2_{\pm 0.8}$	$76.6_{\pm 1.3}$	$84.1_{\pm 1.4}$	$67.1_{\pm 2.7}$	$78.3_{\pm 1.5}$	$\mathbf{69.1_{\pm 0.9}}$	$75.5_{\pm 0.9}$
SDSD K=3	$53.2_{\pm 3.3}$	$56.4_{\pm 3.3}$	$\mathbf{80.8_{\pm 1.1}}$	$84.3_{\pm 0.3}$	$\mathbf{78.6_{\pm 1.2}}$	$\mathbf{85.1_{\pm 1.7}}$	$\mathbf{67.4_{\pm 1.8}}$	$79.5_{\pm 2.7}$	$68.7_{\pm 2.7}$	$75.3_{\pm 2.6}$

4.3 Coding with Rich Environment Feedback

We next investigate whether the reduction in functional diversity extends to coding environments. We follow the LiveCodeBench setup of Hübotter et al. (2026) and restrict evaluation to the LCBv6 subset. We train Qwen3-8B models using GRPO and several variants of self-distillation.



(a) Sample Concept Graph.

Method	Pass@k 1	2	4	8	16	$\Delta_{1\to16}$
GRPO	40.5	42.9	44.5	45.7	46.7	**+6.2**
SDSD	44.7	45.3	45.6	45.8	45.9	+1.2
SDSD + feedback	47.6	48.2	48.7	48.9	49.0	+1.4
SDSD else-feedback	48.4	48.8	48.9	49.0	49.1	+0.7

(b) LiveCodeBench v6 results at the final training step. Compared to GRPO, SDSD has low functional diversity, where additional samples bring less gains. Average over 3 seeds.

Figure 5: **(a)** A Concept Graph instance with chain length 3: four concept chains radiate from *start*; the two yellow triangle endpoints are valid targets, the two gray squares are distractors. **(b)** LiveCodeBench v6 pass@k at the final training step, averaged over three seeds. SDSD has good Pass@1 while having low Pass@16 showing decreased functional diversity.

Rich environment feedback. Coding environments provide rich textual feedback such as failing test cases, the produced and expected outputs, or runtime exceptions. Following Hübotter et al. (2026), this feedback is additionally provided as privileged context to the teacher.

We consider the following variants. **SDSD** is our standard setting, where the EMA teacher is conditioned on a correct demonstration sampled from the student's other rollouts. **SDSD+feedback** provides both a sampled correct demonstration and the environment feedback associated with the current student rollout. **SDSD else-feedback** follows the SDPO setup: when a correct student solution is available, it is used as privileged context; otherwise, the teacher receives the environment feedback.

Table 5b shows that SDSD has good Pass@1 while having lower Pass@16 than GRPO. Thus, this coding setup also shows a decreased functional diversity of SDSD. Using additional feedback increases the coverage, as shown by Pass@16 but doesn't solve the diversity problem.

4.4 Diversity is not achieved by diverse demos nor strong teachers alone

What interventions would make distillation have high diversity? Would diverse demonstrations be enough, or would a better teacher suffice? We show that individually both are insufficient and show that solving the diversity problem requires an interplay between them.

We first train models on a fixed pool of correct demonstrations, obtained from a GRPO+diversity model, that are highly diverse. An SDSD model trained with the standard EMA teacher conditioned on them still has low diversity and low performance on the Harder Graphs task that requires diversity. E.g. we see in table 2) that Concept Diversity increases from 1.034 of standard SDSD to only 1.119. Using the initial model as a teacher slightly increases the diversity metrics (e.g. Concept Diversity reaches 1.138). The key is to use diverse demonstrations and use the GRPO+diversity model as a fixed teacher. Training with these achieves good performance on all metrics, including the ones requiring diverse outputs (e.g., 1.474 on Concept Diversity). Nevertheless, when using a strong teacher with student-sampled demos, the diversity metrics collapse (e.g., 1.031 on Concept Diversity).

Thus, an interplay between diverse demonstrations and a strong teacher leads to training diverse models, but individually each one is not enough. While these modifications lead us away from the self-distillation framework, they show us axes for further improvements in diversity.

Additional results. In the Appendix, we also show results on math A.1, demo diversity A.2, argue that token entropy is not sufficient for diversity A.3, show temperature doesn't control semantic diversity A.5, ConceptGraph results with bigger models section A.6, and detailed derivations B.

5 Related Work

On-policy self-distillation. Distillation transfers knowledge from a teacher model to a student (Hinton et al., 2015; Bucilă et al., 2006), and on-policy distillation gives teacher guidance on student-

Method	In-Distribution Pass@1 / @8	Larger Graphs Pass@1 / @8	Harder Graphs Pass@1 / @8	Concept Diversity
GRPO + diversity	92% / **100%**	5% / 29%	76% / **98%**	**1.797**
SDSD	90% / 91%	87% / **89%**	3% / 3%	1.034
SDSD + diverse demos, EMA teacher	78% / 92%	42% / 79%	6% / 21%	1.119
SDSD + diverse demos, initial teacher	76% / 91%	45% / 80%	15% / 34%	1.138
SDSD + diverse demos, strong teacher	97% / **100%**	47% / **90%**	63% / **84%**	**1.474**
SDSD + strong teacher	99% / **100%**	74% / 84%	4% / 7%	1.031

Table 2: SDSD has lower diversity. Separately using a fixed pool of demonstrations, or a strong teacher, does not solve the diversity problem. Using both, the diversity is significantly improved, showing axes for further improvements.

generated data (Agarwal et al., 2024). Moreover, self-distillation methods like SDPO (Hübotter et al., 2026), OPSD (Zhao et al., 2026; Penaloza et al., 2026), SDFT (Shenfeld et al., 2026), RLSD (Yang et al., 2026), and OPCD (Ye et al., 2026) use the same model as the teacher to give guidance by conditioning it on privileged information, achieving strong results. All these approaches use token-wise supervision to give dense feedback to the student, with Zhao et al. (2026) noting that significant gains come from matching the teacher and student over the whole vocabulary. This dense feedback makes self-distillation achieve high performance in small number of steps (Zhao et al., 2026; Yang et al., 2026) before plateauing or decreasing. SDPO (Hübotter et al., 2026) uses two settings: one where the demonstrations are collected from correct student responses and one where coding runtime feedback is used as privileged information. For the second approach, the alignment between feedback and student rollout should be implicitly higher, since the feedback is generated based on the rollout, thus the teacher should have an easier time understanding their relation and conversely provide good guiding signal. RLSD (Yang et al., 2026) points to an irreducible gap in the objective of self-distillation, which results in privileged information leakage when the optimization is done, as usual, in mini-batches. They propose to fix this by using the teacher guidance to change the magnitude of the RL gradient, but keep the direction given by the verifiable reward.

Entropy collapse and mode-seeking. Entropy collapse under on-policy training is widely documented (GX-Chen et al., 2026; Yue et al., 2025; Wu et al., 2025) with prior work attributing it primarily to mode-seeking of the on-policy training. Our analysis identifies an additional mechanism specific to self-distillation: unequal alignment between the sampled rollouts and the sampled demonstrations, as defined by PCMI. GX-Chen et al. (2026) analyze the optimal RL policy and suggest transformed reward targets to improve diversity, while Yu et al. (2025) propose an entropy regularized multi-turn preference optimization to improve the diversity of resulting optimal policy. Nagarajan et al. (2025) analyze the diversity and creativity of LLMs using controlled graph understanding tasks.

Prior work on RLHF shows that RL-trained models are less diverse than SFT models (Kirk et al., 2024). This is addressed by Pass@k-aware training objectives (Chen et al., 2025b), and best-of-N fine-tuning (Chow et al., 2025) that directly optimize for output coverage. Li et al. (2025) uses LLM embeddings to partition the rollouts and use them to compute a diversity score. Multiplying the reward with this diversity score improves the diversity of math reasoning and creative writing.

6 Conclusion

We analyzed on-policy self-distillation with sampled demonstrations (SDSD) through the lens of rollout diversity. While SDSD models achieve strong average accuracy, their pass@k curves are often flat, indicating collapsed functional diversity. Theoretically, the optimal self-distillation policy tilts the base distribution by a pointwise conditional mutual information (PCMI) score between the student rollout and the demonstration. Unlike standard RL objectives, which preserve probability ratios among equally correct rollouts, this mechanism amplifies pre-existing imbalances and reinforces solutions that already align with the demonstrations and base policy. Empirically, we observed this diversity collapse on controlled graph path-finding, scientific QA, and coding tasks. Our results show that average accuracy alone is insufficient for evaluating self-distillation methods, and functional and semantic diversity are at risk of collapse and should be explicitly monitored.

AI USE STATEMENT

In this work, we used generative AI tools for helping with coding implementations, proving the mathematical results, and for manuscript polishing. AI models were not used for establishing and conceptualization of the research direction, designing of the methodology, experiments, or theory.

In more detail, in this work, we used generative AI tools to: formulate mathematical claims, assist in the writing of proofs, implement methods, generate the synthetic graph data set. We didn't use them to help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, or interpret results. We didn't need to clean and reformat dataset, or assist with translation.

We assume full responsibility for everything stated in the paper.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

Christopher M Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541, 2006.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*, 2025a.

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. *arXiv preprint arXiv:2508.10751*, 2025b.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Aviral Kumar, Rishabh Agarwal, Sridhar Thiagarajan, Craig Boutilier, and Aleksandra Faust. Inference-aware fine-tuning for best-of-n sampling in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=77gQUdQhE7.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Anthony GX-Chen, Jatin Prakash, Jeff Guo, Rob Fergus, and Rajesh Ranganath. KL-regularized reinforcement learning for generative modelling is designed to mode collapse. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=flBRtdIihA.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Amirhossein Kazemnejad, Milad Aghajohari, Alessandro Sordoni, Aaron Courville, and Siva Reddy. Nano aha! moment: Single file "rl for llm" library. https://github.com/McGill-NLP/nano-aha-moment, 2025. GitHub repository.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PXD3FAVHJT.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. Rl with kl penalties is better viewed as bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1083–1091, 2022.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Kevin Lu and Lab Thinking Machines. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Vaishnavh Nagarajan, Chen Henry Wu, Charles Ding, and Aditi Raghunathan. Roll the dice & look before you leap: Going beyond the creative limits of next-token prediction. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Hi0SyHMmkd.

Team OLMo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Jiahao Yu, Zelei Cheng, Xian Wu, and Xinyu Xing. Building coding agents via entropy-enhanced multi-turn preference optimization. *arXiv preprint arXiv:2509.12434*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4OsgYD7em5.

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models. *arXiv preprint arXiv:2512.07783*, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

A Additional Results

A.1 Math Reasoning: Memorization Under Overtraining

Goal. Here we show that self-distillation approaches have more biases for memorizing the answers without internalizing the reasoning process.

We use DeepScaleR [1], a collection of competition-level mathematics problems. We train on the full dataset, minus a split of 100 questions used as a validation set, and train Qwen3-1.7B models operating in non-thinking mode. All experiments run on a single node with 8 GPUs. As a baseline, we use GRPO with outcome-based advantages and KL regularization against the reference policy. We compare with SDSD, where one correct rollout among the $N{=}8$ rollouts is used as context.

First, similar to the findings of previous work (Yang et al., 2026), we see that SDSD completely collapses to low performance and low diversity. To investigate this, we propose a simple experiment where we train both GRPO and SDSD on a set of 100 questions for 500 epochs to test the memorization capabilities of both methods. In Fig. 6, we observe a different behavior in both methods. GRPO increases both its pass@1 and pass@8 performance across the training steps. Since pass@8 is higher than pass@1, it shows that GRPO still maintains *functional diversity* even though it has seen the same questions an extreme number of times. Manually inspecting its rollouts, we observe that it produces full and different solutions across rollouts. On the other hand, SDSD maintains the pass@1 of the base model but its pass@k does not increase. This shows that SDSD loses its *functional diversity* as multiple successive rollouts do not produce more correct answers.This shows that SDSD is significantly more prone to memorizing the answers and collapsing its rollouts.

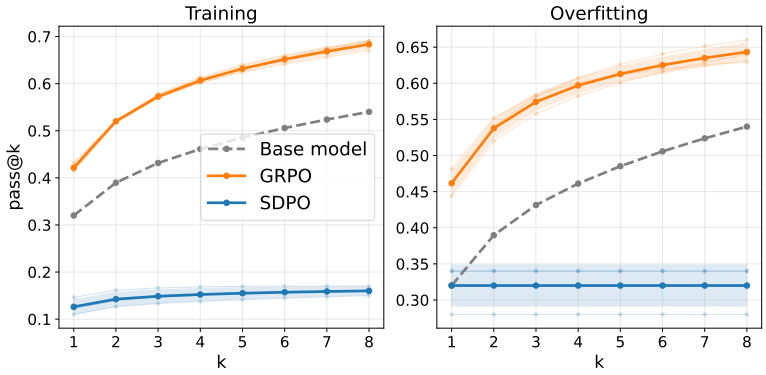


Figure 6: DeepScaleR math reasoning: First, training on the full DeepScaleR dataset SDSD collapses to low performance and low rollout diversity. Second, in extreme overfitting on a small validation set, GRPO still improves its performance while maintaining diversity. SDSD maintains average performance but collapses functional diversity, as seen by the flat pass@k curve.

A.2 Diverse External Demonstrations Still Lead to Diversity Collapse

Previously, we evaluated SDSD on demonstrations coming from the student's own correct rollouts. Similar to approaches like OPSD (Zhao et al., 2026), we now investigate the case when the demonstrations come from external models. We will see that self-distillation with external demonstrations still suffers from diversity collapse, regardless of the diversity level of the demonstrations.

We analyze this in the Concept Graph setup. We create multiple demonstration datasets, with different levels of diversity, where each query has multiple correct solutions. We obtain the demonstrations from different checkpoints of Qwen3-1.7B GRPO+diversity regularizer models. For each dataset, we train SDSD models as in section 4.1 but this time we condition the EMA teacher on these external demonstrations instead of self-generated demonstrations. This way, we can control the diversity of the teacher demonstrations and see their influence on the diversity of the learned student.

[1] https://huggingface.co/datasets/agentica-org/DeepScaleR-Preview-Dataset

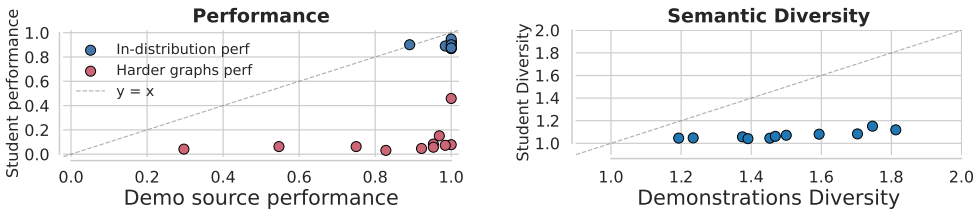


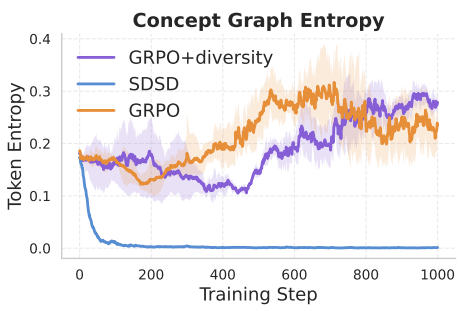
Figure 7: We train SDSD models using external demonstrations that are both correct and diverse. We use multiple datasets of demonstrations with increasing levels of diversity. Each demonstration dataset leads to SDSD models with good in-distribution performance (left plot), but low semantic diversity (right plot), regardless of the level of diversity in the demonstration. This shows that, even with external demonstrations, we still have a problem of diversity collapse, regardless of the level of diversity of the demonstrations.

All resulting SDSD models have high in-distribution performance (pass@8), similar to the models that generated the demonstrations (Fig. 7). Nevertheless, on the Harder Graph setup (that requires diverse exploration during training) pass@8 performance of the students remains low. Moreover, we compare the semantic diversity of the demonstrations and the diversity of the resulting SDSD models for in-distribution questions. We observe that the student models have low semantic diversity, regardless of the level of diversity of the demonstrations. This shows that using diverse demonstrations in the teacher does not fix the diversity problem.

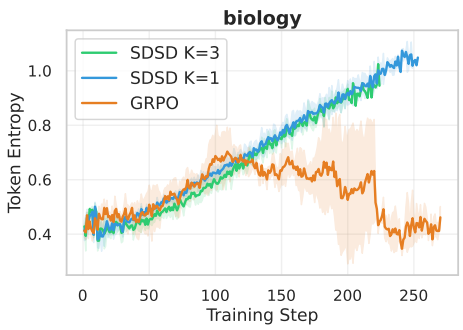
A.3 Token Entropy Is Not a Sufficient Metric of Diversity

In Fig. 8a on Concept Graphs, SDSD has clearly lower average token-level entropy than the GRPO baselines. This correlates well with SDSD's lack of semantic diversity. On the other hand, token-level entropy cannot distinguish between GRPO and GRPO+diversity, even though GRPO+diversity is clearly more semantically diverse and has better performance on the Harder Graphs task that requires diverse rollouts during training.

Conversely, in the QA setting (Fig. 8b), SDSD has *higher* token-level entropy than GRPO despite lower functional diversity and pass@16. Again, token-level entropy does not correlate well with a meaningful notion of diversity or performance. This suggests that we need more nuanced notions of diversity than entropy at the token level.



(a) Token Entropy of Concept Graph models



(b) Token Entropy of QA models.

Figure 8: **(a, b)** Token-level entropy does not tell the whole story. For ConceptGraph, token-level entropy cannot explain the higher semantic diversity (Fig. 3) of GRPO+diversity compared to GRPO. In the QA setting, average token-level entropy does not correlate with functional diversity or low pass@k, since GRPO has the lowest token-level entropy but highest functional diversity and pass@16.

Method	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16
GRPO	62.0	67.4	71.2	**73.7**	**75.4**
SDSD	61.9	65.5	68.1	70.0	71.3
SDSD + feedback	64.7	68.2	70.8	72.7	73.9
SDSD else-feedback	**65.3**	**68.7**	**71.3**	73.2	74.5

Table 3: LiveCodeBench v6 test-pass rate pass@k (fraction of test passed using k rollouts) at the final training step. Results averaged over three seeds.

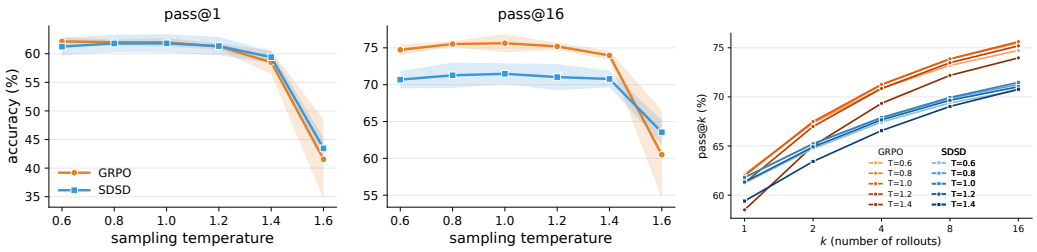


Figure 9: a) b) LiveCodeBench test-pass rate Pass@1 / 16(%) as a function of evaluation sampling temperature. In the range of temperatures with reasonable results (good pass@1, up to temperature 1.4), the SDSD has lower Pass@16 while having similar pass@1 to GRPO, meaning SDSD has low functional diversity. c) Pass@k of SDSD and GRPO with different temperature values: we observe that, regardless of temperature, SDSD has the flatter curve, meaning lower functional diversity.

A.4 Additional coding results

In table 3 we also report the performance of LiveCodeBench models we discussed in section 4.3, measured by the test-pass rate, the fraction of its unit tests passed for all problems, alongside the fractions of problems with all unit tests passed that we previously reported.

The same conclusions holdshold; SDSD has good pass@1 but is overpassed by GRPO in terms of pass@16, showing that SDSD has low functional diversity. While using feedback increases the coverage (pass@16) but doesn't solve the diversity problem.

A.5 Robustness to Sampling Temperature and Checkpoint Selection

A flatter Pass@k curve could in principle result from a poorly chosen sampling temperature rather than a change in the learned policy. We therefore re-evaluate the trained models over a broad range of sampling temperatures.

LiveCodeBench. Figure 9 reports test-pass rate Pass@1/Pass@16 at each temperature. Across the practically useful temperature range, GRPO consistently obtains the largest Pass@16. Increasing the sampling temperature therefore does not remove the functional-diversity gap.

Concept Graph. Temperature also cannot recover the explicitly measured semantic diversity of SDSD. Across temperatures from 0.6 to 3.0, its Concept Diversity remains approximately 1.03–1.04.

Importantly, the same conclusion holds for the Harder Graph evaluation, which requires diverse solution strategies: SDSD remains close to zero across the temperature sweep. Thus, higher sampling

Temperature	.6	.8	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
GRPO	1.12	1.13	1.12	1.13	1.11	1.07	1.15	1.01	–	–	–	–	–
GRPO+diversity	1.74	1.77	1.78	1.78	1.73	1.15	1.00	–	–	–	–	–	–
SDSD	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.04	1.04	1.04	1.04	1.03	1.03

Table 4: In-distribution Concept Diversity at $k = 8$ under different evaluation temperatures. The semantic Diversity of SDSD remains low at all temperatures.

temperature may increase surface-level stochasticity, but it does not recover the missing learned solution modes.

Finally, checkpoint selection does not explain the result. On Concept Graph, SDSD has the lowest directly measured semantic diversity throughout training, rather than only at the final checkpoint. The LiveCodeBench experiments similarly show a persistent Pass@1–Pass@16 gap across training steps. Together, these results indicate that the diversity reduction reflects the learned policy rather than a particular decoding temperature or checkpoint.

A.6 CONCEPT-GRAPH

Similar to Fig. 3, we show performance and diversity results using different Qwen3-4B, Qwen3-8B, Olmo-3-7B-Instruct models in Fig. 10 and Fig. 11.

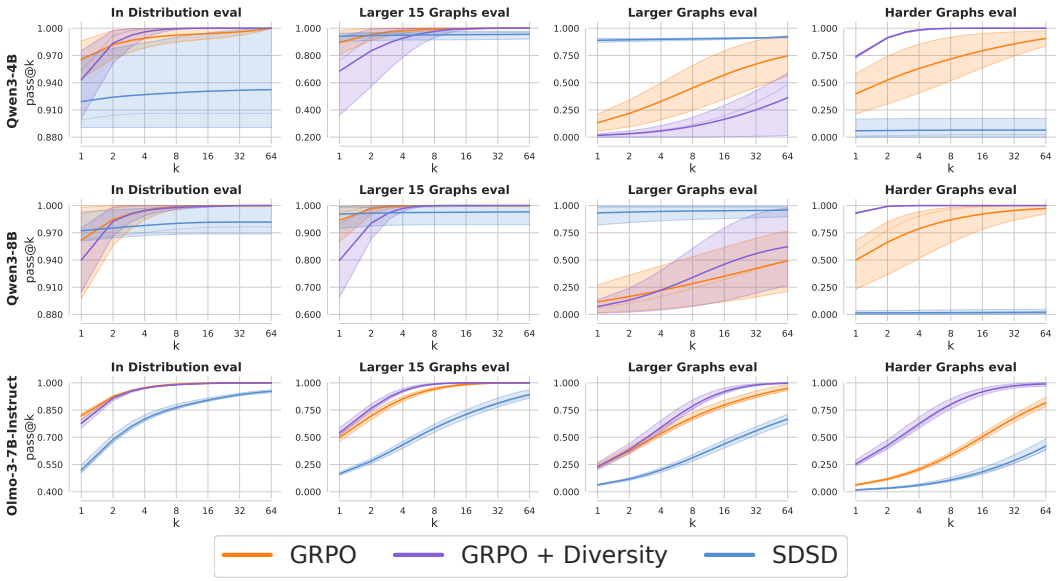


Figure 10: Compare GRPO, GRPO + Diversity, SDSD in terms of performance (pass@k curves) for models of different sizes. We observe that SDSD usually has flatter pass@k curves for In-Distribution and Larger Graphs (15 nodes or 20 nodes), showing low functional diversity. The Harder Graphs task that requires learning diverse trajectories during training remains mainly unsolved by SDSD, showing again a lack of output diversity. Mean and min/max over 3 seeds are shown.

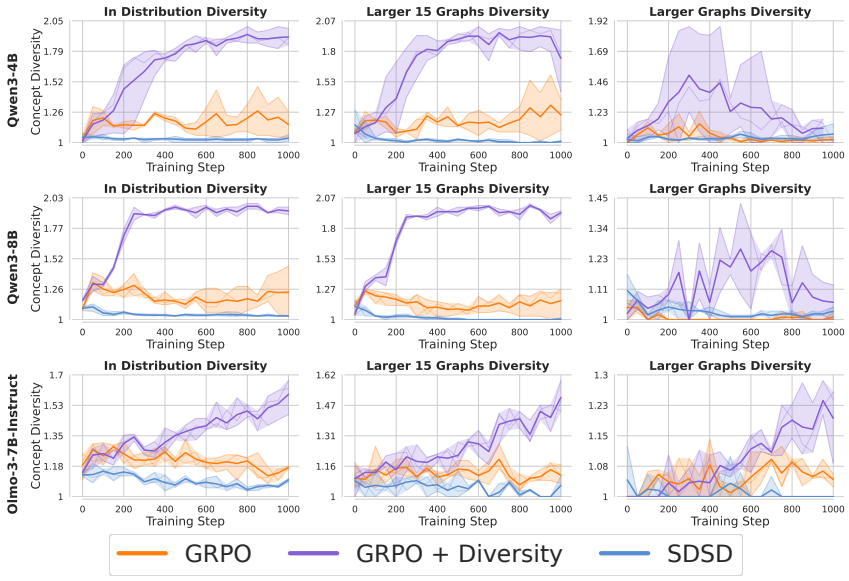


Figure 11: Compare GRPO, GRPO + Diversity, SDSD in terms of semantic diversity for models of different sizes. We observe that SDSD has the lowest concept diversity (as defined in the main text) across training. This explicit measure of diversity gives additional evidence of the loss of diversity in SDSD models.

A.7 SCIENCE QA

We show the average performance evolution across training steps in Fig. 12.

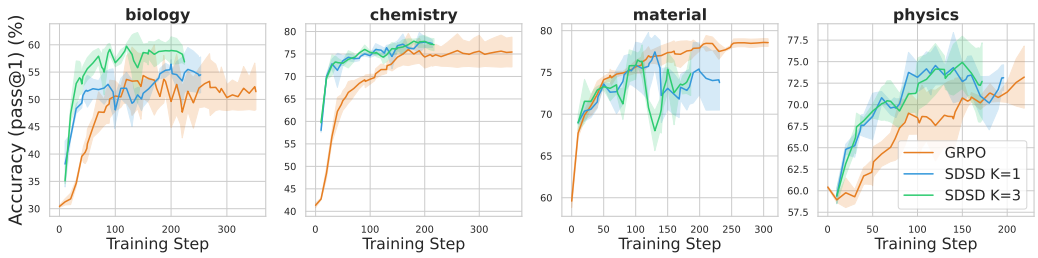


Figure 12: Pass@1 scores of SDSD and GRPO across 5h of training time.

B DETAILED DERIVATIONS AND ADDITIONAL RESULTS

B.1 STANDARD KL-REGULARIZED RL

We first consider the standard KL-regularized reinforcement learning objective

$$\max_{\pi} \; \mathbb{E}_{y\sim\pi(\cdot\,|\,x)}[R(y\,|\,x)] - \beta_{\mathrm{RL}}\, D_{\mathrm{KL}}(\pi(\cdot\,|\,x)\,\|\,\pi_0(\cdot\,|\,x))\,. \tag{7}$$

Proposition 3 (Optimal policy for standard KL-regularized RL)**.** *The optimizer of equation 7 is*

$$\pi^*_{\mathrm{RL}}(y\,|\,x) \propto \pi_0(y\,|\,x)\exp\!\left(\frac{1}{\beta_{\mathrm{RL}}}R(y\,|\,x)\right). \tag{8}$$

Proof. Fix x and suppress it in the notation. The objective is

$$\max_{\pi}\sum_y \pi(y)R(y) - \beta_{\mathrm{RL}}\sum_y \pi(y)\log\frac{\pi(y)}{\pi_0(y)} \quad \text{subject to} \quad \sum_y \pi(y) = 1.$$

Its Lagrangian is

$$\mathcal{L}(\pi,\lambda) = \sum_y \pi(y)R(y) - \beta_{\mathrm{RL}}\sum_y \pi(y)\log\frac{\pi(y)}{\pi_0(y)} + \lambda\left(\sum_y \pi(y) - 1\right).$$

Differentiating with respect to $\pi(y)$ gives

$$R(y) - \beta_{\mathrm{RL}}\left(\log\frac{\pi(y)}{\pi_0(y)} + 1\right) + \lambda = 0.$$

Rearranging,

$$\log\pi(y) = \log\pi_0(y) + \frac{1}{\beta_{\mathrm{RL}}}R(y) + c,$$

where c is a constant independent of y. Exponentiating and normalizing yields

$$\pi^*_{\mathrm{RL}}(y) \propto \pi_0(y)\exp\!\left(\frac{1}{\beta_{\mathrm{RL}}}R(y)\right),$$

which, after reintroducing the x into the notation, proves equation 8. □

Remark 2. *Equation equation 8 shows that KL-regularized RL preserves the base policy while exponentially tilting it by the reward.*

B.2 SDSD-KL: Distillation from a Correct Demonstration

Consider as teacher the base policy conditioned on a fixed correct demonstration. Let y^{corr} denote a correct reference demonstration for the same input x. We define the teacher by

$$\pi_0(y \mid x, y^{\mathrm{corr}}). \tag{9}$$

The corresponding pointwise conditional mutual information (PCMI) is

$$i(y; y^{\mathrm{corr}} \mid x) := \log \frac{\pi_0(y \mid x, y^{\mathrm{corr}})}{\pi_0(y \mid x)}. \tag{10}$$

This quantity measures how much conditioning on the fixed correct demonstration changes the base policy's log-probability of the candidate sequence y. Thus, it can be interpreted as how much support y^{corr} brings for y.

But the demonstration is not fixed, it is sampled from the correct solutions. Let

$$y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x) \tag{11}$$

be a correct demonstration drawn from a reference distribution over correct solutions. For each realized y^{corr}, we define the teacher

$$q_{y^{\mathrm{corr}}}(y \mid x) := \pi_0(y \mid x, y^{\mathrm{corr}}). \tag{12}$$

The SDSD-KL objective averages the distillation loss over demonstrations:

$$\min_{\pi} \; \mathbb{E}_{y^{\mathrm{corr}}} \Big[D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x, y^{\mathrm{corr}})) \Big] + \beta \, D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x)) \,. \tag{13}$$

Proposition 4 (Optimal policy for SDSD-KL)**.** *The optimizer of equation 13 is*

$$\pi^*_{\text{SD-KL}}(y \mid x) \propto \pi_0(y \mid x) \exp\left(\frac{1}{1+\beta} \, \mathbb{E}_{y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x)} \big[i(y; y^{\mathrm{corr}} \mid x) \big] \right). \tag{14}$$

Proof. Fix x and suppress it in the notation. Expanding equation 13 gives

$$\min_{\pi} \mathbb{E}_{y^{\mathrm{corr}}} \left[\sum_y \pi(y) \log \frac{\pi(y)}{\pi_0(y \mid y^{\mathrm{corr}})} \right] + \beta \sum_y \pi(y) \log \frac{\pi(y)}{\pi_0(y)} \quad \text{subject to} \quad \sum_y \pi(y) = 1.$$

Interchanging expectation and summation,

$$\min_{\pi} (1+\beta) \sum_y \pi(y) \log \pi(y) - \sum_y \pi(y) \mathbb{E}_{y^{\mathrm{corr}}} [\log \pi_0(y \mid y^{\mathrm{corr}})] - \beta \sum_y \pi(y) \log \pi_0(y).$$

The Lagrangian is therefore

$$\mathcal{L}(\pi, \lambda) = (1+\beta) \sum_y \pi(y) \log \pi(y) - \sum_y \pi(y) \mathbb{E}_{y^{\mathrm{corr}}} [\log \pi_0(y \mid y^{\mathrm{corr}})] - \beta \sum_y \pi(y) \log \pi_0(y) + \lambda \left(\sum_y \pi(y) - 1 \right).$$

Differentiating with respect to $\pi(y)$ gives

$$(1+\beta)(\log \pi(y) + 1) - \mathbb{E}_{y^{\mathrm{corr}}} [\log \pi_0(y \mid y^{\mathrm{corr}})] - \beta \log \pi_0(y) + \lambda = 0.$$

Hence,

$$\log \pi(y) = \frac{1}{1+\beta} \mathbb{E}_{y^{\mathrm{corr}}} [\log \pi_0(y \mid y^{\mathrm{corr}})] + \frac{\beta}{1+\beta} \log \pi_0(y) + c.$$

Using equation 10,

$$\log \pi_0(y \mid y^{\mathrm{corr}}) = \log \pi_0(y) + i(y; y^{\mathrm{corr}}).$$

Therefore, since taking the expectation over y^{corr} does not influence the first term:

$$\mathbb{E}_{y^{\mathrm{corr}}} [\log \pi_0(y \mid y^{\mathrm{corr}})] = \log \pi_0(y) + \mathbb{E}_{y^{\mathrm{corr}}} \big[i(y; y^{\mathrm{corr}}) \big].$$

Substituting back,

$$\log \pi(y) = \log \pi_0(y) + \frac{1}{1+\beta}\mathbb{E}_{y^{\text{corr}}}\big[i(y; y^{\text{corr}})\big] + c.$$

Exponentiating and normalizing yields:

$$\pi(y) \propto \pi_0(y) \exp\left(\frac{1}{1+\beta}\,\mathbb{E}_{y^{\text{corr}}}\big[i(y; y^{\text{corr}})\big]\right). \tag{15}$$

Restoring x to the notation gives equation 14. □

Remark 3 (Interpretation of SDSD-KL)**.** *SDSD-KL has the same formal structure as KL-regularized RL: it exponentially tilts the base policy. The effective reward is now the expected PCMI,*

$$\mathbb{E}_{y^{\text{corr}}\sim p_{\text{corr}}(\cdot\,|\,x)}\big[i(y; y^{\text{corr}} \mid x)\big],$$

which measures how strongly, on average over correct demonstrations, the base policy shifts toward the candidate sequence y.

Remark 4 (Ratio for two correct sequences under SDSD-KL)**.** *Let y_1 and y_2 be two correct sequences for the same input x, and suppose*

$$\pi_0(y_1 \,|\, x) = k\, \pi_0(y_2 \,|\, x)$$

for some $k > 0$. Then, by equation 14, their ratio under the optimal SDSD-KL policy is

$$\frac{\pi^*_{\text{SD-KL}}(y_1 \,|\, x)}{\pi^*_{\text{SD-KL}}(y_2 \,|\, x)} = k\, \exp\left(\frac{1}{1+\beta}\mathbb{E}_{y^{\text{corr}}\sim p_{\text{corr}}(\cdot\,|\,x)}\big[i(y_1; y^{\text{corr}} \mid x) - i(y_2; y^{\text{corr}} \mid x)\big]\right). \tag{16}$$

Equivalently,

$$\frac{\pi^*_{\text{SD-KL}}(y_1 \,|\, x)}{\pi^*_{\text{SD-KL}}(y_2 \,|\, x)} = k\, \exp\left(\frac{1}{1+\beta}\mathbb{E}_{y^{\text{corr}}\sim p_{\text{corr}}(\cdot\,|\,x)}\left[\log\left(\frac{\exp(i(y_1; y^{\text{corr}} \mid x))}{\exp(i(y_2; y^{\text{corr}} \mid x))}\right)\right]\right).$$

Thus SDSD-KL preserves the base-policy ratio k only when the two sequences have the same expected PCMI. Otherwise, SDSD-KL further reweights them according to the gap in how strongly correct demonstrations support y_1 versus y_2.

B.3 Optimal Policy for Token-Level SDSD-KL

Previously, we discussed the sequence level objective and its implications. This was done for ease of presentation, but in practice, we use a token-level objective. We will see that exactly the same derivations carry in the token-level case, and we will discuss the implications.

Consider the optimization problem at a single generation step t. The policy generates the next token y_t from the vocabulary, conditioned on the input x and the generated prefix $y_{<t}$. The token-level SDSD-KL objective averages the distillation loss over demonstrations at the next-token distribution:

$$\min_{\pi} \mathbb{E}_{y^{corr}\sim p_{corr}(\cdot|x)}\left[\text{KL}(\pi(\cdot|x, y_{<t})||\pi_0(\cdot|x, y_{<t}, y^{corr}))\right] + \beta\text{KL}(\pi(\cdot|x, y_{<t})||\pi_0(\cdot|x, y_{<t})) \tag{17}$$

Proposition 5 (Optimal policy for token-level SDSD-KL)**.** *The optimizer of the token-level objective is:*

$$\pi^*_{SD-KL}(y_t|x, y_{<t}) \propto \pi_0(y_t|x, y_{<t}) \exp\left(\frac{1}{1+\beta}\mathbb{E}_{y^{corr}\sim p_{corr}(\cdot|x)}[i(y_t; y^{corr}|x, y_{<t})]\right) \tag{18}$$

where the token-level pointwise conditional mutual information (PCMI) is defined as:

$$i(y_t; y^{corr}|x, y_{<t}) := \log \frac{\pi_0(y_t|x, y_{<t}, y^{corr})}{\pi_0(y_t|x, y_{<t})} \tag{19}$$

The proof follows exactly the same steps as before.

Implications of the token-level objective. In the case of sequence-level objective, we have seen that there is a bias for common rollouts. That bias comes from the alignment between rollouts and demonstrations and the preference of the teacher. A similar phenomenon is happening at the token-level. Some tokens are more aligned to the context determined by the demonstration and the previous tokens of the student rollout. Again, next-tokens that will move the current rollout into a novel or uncommon direction might be less aligned to the context. Moreover, given the same context, some next-tokens are preferred over others by the teacher.

On top of this, differently from the sequence level objective, here the teacher is myopic to the full student rollout. It guides the next token without taking into account the relation to future tokens. Thus, some commonalities between student rollout and demonstrations might only be seen at a sequence level and could be harder to establish at intermediate token positions. Thus, the alignment at intermediate token positions might be low, causing learning to be even more biased. This leads to a bias for common next-tokens, which turns into common entire distributions as we discussed before.

B.4 Concept Graph Task: Additional details

In the Concept Graph task, each query is a problem of finding a path given to the LLM. Each query has a different graph with randomly sampled structure and node names. Here is an example:

```
You are given the following graph structure.
Nodes:
  0   start           [hub]
  1   triangle        [shape]
  2   pigeon          [concept=birds]
  3   parrot          [concept=birds]
  4   sparrow         [concept=birds]
  5   crow            [concept=birds]
  6   heron           [concept=birds]
  7   eagle           [concept=birds]
  8   square          [shape]
  9   tuna            [concept=fish]
  ...
Edges:
  0 (start) -- 2 (pigeon)
  0 (start) -- 5 (crow)
  0 (start) -- 9 (tuna)
  0 (start) -- 21 (plum)
  ...
  2 (pigeon) -- 3 (parrot)
  3 (parrot) -- 4 (sparrow)
  1 (triangle) -- 4 (sparrow)
  ...
  21 (plum) -- 22 (mango)
  22 (mango) -- 27 (triangle)
  ...
Your task is to generate a path from the start node to the target node
    named triangle. For the output, print the names of the nodes in the
    path, use the following format: A path from start to triangle is \boxed
    {start, node_1_name, ..., triangle} for example \boxed{start, Jacobi,
    Hamilton, ..., star}.
```

A correct response following the *birds* concept chain would be:

```
A path from start to triangle is \boxed{start, pigeon, parrot, sparrow, triangle}
```

An equally valid alternative following a different concept (*fruits*) would be:

```
\boxed{start, plum, mango, triangle}
```

Both receive the same maximum reward, but a model that *only* produces bird-paths across different graphs exhibits lower solution diversity than one that explores both bird and fruit paths.