On-Policy Self-Distillation with Sampled Demonstrations Reduces Output Diversity

Anonymous Authors[1]

Abstract

On-policy self-distillation has recently established as an important method to improve the reasoning capabilities of LLMs. It strength comes from using the in-context capabilities of LLMs such that the learning models can be used as a teacher to incorporate knowledge from successful demonstrations or feedback. We show that this could come at a cost, when using sampled demonstrations we observe a decrease in output diversity. This is due to compounding biases, there is an implicit matching problem between sampled demonstration and student rollouts that is modulated through the models 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 confirms this pattern: competitive average performance but substantially lower *functional* and *semantic* diversity than RL models, and failure 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 & Lab, 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 (Figure 3, Figure 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 more *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 unconventional rollout when conditioned on a more standard or canonical demonstration, 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

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

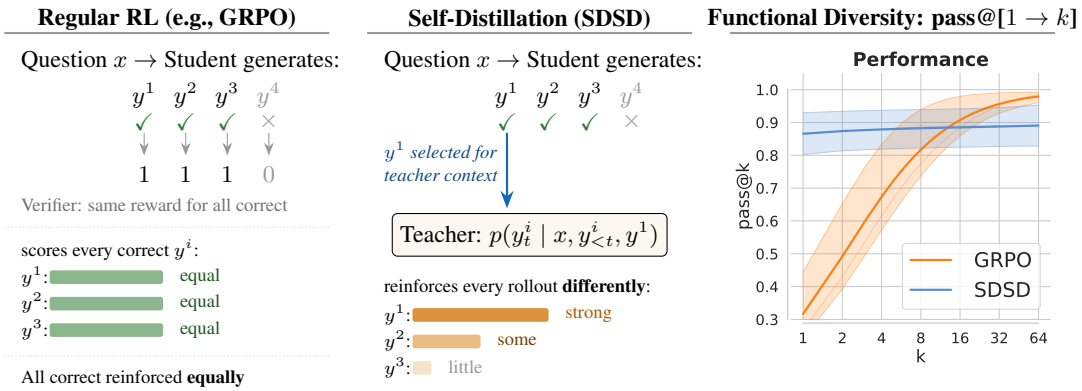


Figure 1. RL and self-distillation treat correct rollouts differently, with consequences for rollout diversity. **Left:** A binary verifier gives equal reward to all correct rollouts, so RL reinforces them uniformly. **Middle:** Self-distillation conditions the teacher on a sampled correct rollout, typically 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 SDSD: generating more rollouts fails to solve new problems, unlike GRPO where each additional sample meaningfully increases coverage.

compound: rollouts that are more aligned with previously sampled demonstrations become increasingly reinforced, while less aligned, but still correct, solutions are gradually suppressed. 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 3.2). 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 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 (§4.4).

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 RL (Proposition 3.2, Remark 3.3).
- We introduce a graph path-finding environment, 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 and science QA tasks (Feng et al., 2024), we show that self-distillation achieve 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).

2. Background: 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:

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

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

$$\nabla_\theta \mathcal{L}_{\mathrm{SD}}(\theta; x) = \frac{1}{N} \sum_{y \in \mathcal{Y}(x)} \mathbb{E}_{y^{\mathrm{corr}}} \Big[\sum_{t=1}^{|y|} \mathbb{E}_{\hat{y}_t \sim \pi_\theta} \Big[\log \frac{\pi_\theta(\hat{y}_t \mid x, y_{<t})}{\pi_{\bar{\theta}}(\hat{y}_t \mid x, y^{\mathrm{corr}}, y_{<t})} \nabla_\theta \log \pi_\theta(\hat{y}_t \mid x, y_{<t}) \Big] \Big]. \tag{2}$$

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. SDSD as a reverse KL objective (Bishop & Nasrabadi, 2006) has the same behavior, and even more. We show that it has *additional compounding biases* making this even more pronounced. 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 gets likelier) behavior stronger than in standard RL, arising from both the double sampling 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 fixed base policy, with the teacher being a conditioned version thereof. We optimize a student policy $\pi(y \mid x)$. All distributions are assumed strictly positive on their support.

Proposition 3.1 (Optimal policy for standard KL-regularized RL)**.** *The standard RL objective is:*

$$\max_\pi \; \mathbb{E}_{y \sim \pi(\cdot \mid x)} \left[R(y \mid x) \right] - \beta_{\mathrm{RL}} \, \mathrm{KL}(\pi(\cdot \mid x) \, \| \, \pi_0(\cdot \mid x)) \, . \tag{3}$$

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

$$\pi^*_{\mathrm{RL}}(y \mid x) \propto \pi_0(y \mid x) \exp\left(\frac{1}{\beta_{\mathrm{RL}}} R(y \mid x) \right). \tag{4}$$

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. The teacher is the base policy conditioned on a correct demonstration, and the objective is the KL divergence between student and teacher, with an additional 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 (§4.3).

Let's define the following objective that takes the expectation over demonstrations.

Proposition 3.2 (Optimal policy for SDSD-KL)**.** *Let the self-distillation + KL objective:*

$$\min_\pi \; \mathbb{E}_{y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x)} \Big[\mathrm{KL}(\pi(\cdot \mid x) \, \| \, \pi_0(y \mid x, y^{\mathrm{corr}})) \Big] + \beta \, \mathrm{KL}(\pi(\cdot \mid x) \, \| \, \pi_0(\cdot \mid x)) \, . \tag{5}$$

The optimal policy of (5) *is*

$$\pi^*{}_{\text{SD-KL}}(y \mid x) \propto \pi_0(y \mid x) \exp\left(\frac{1}{1+\beta} \, \mathbb{E}_{y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x)} \big[i(y; y^{\mathrm{corr}} \mid x) \big] \right), \tag{6}$$

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

$$i(y; y^{\mathrm{corr}} \mid x) := \log \frac{\pi_0(y \mid x, y^{\mathrm{corr}})}{\pi_0(y \mid x)}. \tag{7}$$

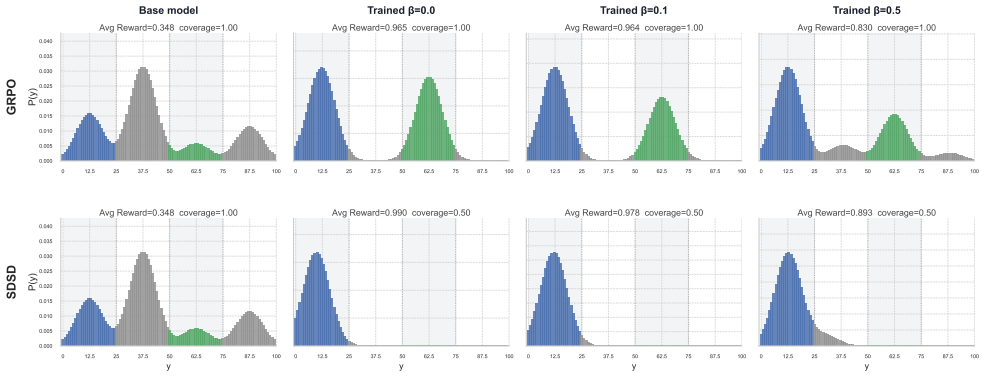


Figure 2. Illustrative example: SDSD collapses to a single high-reward mode while GRPO with KL regularization covers both modes. The environment has two equally-valued reward regions, so an ideal policy would maintain coverage of both.

The RL optimal policy (4) tilts the base policy by the reward. The self-distillation optimum instead tilts by the expected PCMI (7): 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 candidate aligns with correct demonstrations.

Remark 3.3 (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

$$\frac{\pi^*_{\text{SD-KL}}(y_1 \mid x)}{\pi^*_{\text{SD-KL}}(y_2 \mid x)} = k\,\exp\!\left(\frac{1}{1+\beta}\mathbb{E}_{y^{\text{corr}}\sim p_{\text{corr}}(\cdot \mid x)}\Big[i(y_1; y^{\text{corr}} \mid x) - i(y_2; y^{\text{corr}} \mid x)\Big]\right). \tag{8}$$

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 (4), 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 implication are happening for the token-level objective, that has bias for sharpening the distribution of the next-token, leading to loss of diversity of the whole rollout (see Section B.3 from Appendix).

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 more 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 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 $\pi_{\bar{\theta}}(y \mid y^{\text{corr}}) \propto \pi_\theta(y) \cdot K(y, y^{\text{corr}})$, where K is a Gaussian kernel centered on 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.

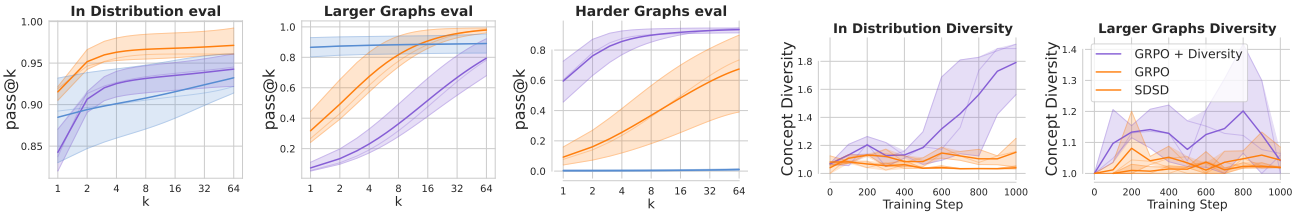


Figure 3. In-distribution and Larger Graphs evaluations show Self-Distillation (SD) has good pass@1 performance but the pass@k curve has a small slope highlighting low *functional diversity*. The third setup requires the model to have learned diverse rollouts during training, and is completely unsolved by SD. Additionally, the last two figures show that the explicitly defined *semantic diversity* of SD is the lowest. All runs train Qwen3-1.7B models, and we show mean and min/max runs across 3 seeds.

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. 6a. 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).

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

GRPO+diversity. The GRPO+diversity variant, similar to Li et al. (2025), adds a diversity reward to the score reward to encourage the model to explore different concept chains across its N rollouts. For each rollout i, its diversity score is the fraction of the other $N-1$ rollouts in the same group that used a disjoint set of concepts, measuring how distinct i's path is from the other rollouts produced for the same query.

SDSD. SDSD implementation following SDPO (Hübotter et al., 2026), where for each student rollout, the teacher 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. flowers 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 notice that GRPO has relatively low semantic diversity, but

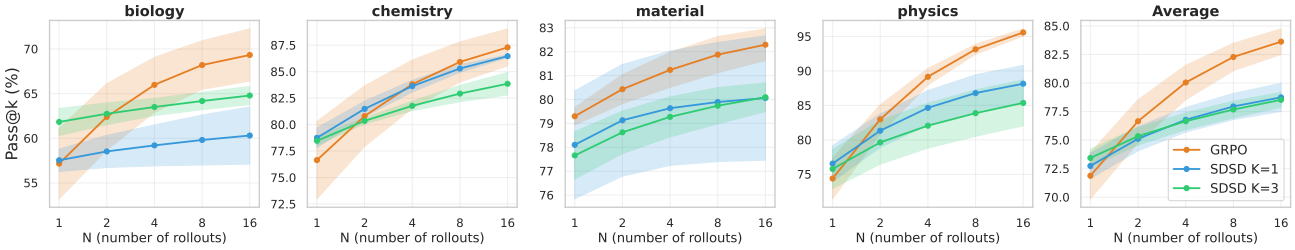


Figure 4. Pass@k curves for Science QA tasks. SDSD achieves competitive pass@1 but its curves flatten quickly, indicating low functional diversity. All methods use Qwen3-8B; mean ± stderr across 3 seeds.

Table 1. Pass@1 and Pass@16 at the best checkpoint per dataset. 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
GRPO off-policy	$57.2_{\pm 3.9}$	$\mathbf{69.3_{\pm 2.9}}$	$76.6_{\pm 3.6}$	$\mathbf{87.3_{\pm 1.7}}$	$\mathbf{79.3_{\pm 0.4}}$	$\mathbf{82.3_{\pm 0.7}}$	$74.4_{\pm 2.9}$	$\mathbf{95.6_{\pm 0.5}}$	$71.9_{\pm 2.0}$	$\mathbf{83.6_{\pm 1.1}}$
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	$\mathbf{61.8_{\pm 1.5}}$	$64.8_{\pm 1.0}$	$78.5_{\pm 0.4}$	$83.9_{\pm 1.1}$	$77.7_{\pm 1.0}$	$80.1_{\pm 0.6}$	$75.8_{\pm 2.8}$	$85.4_{\pm 3.3}$	$\mathbf{73.4_{\pm 0.8}}$	$78.5_{\pm 0.6}$

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) as the base model, 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, making successive updates increasingly off-policy relative to the generating model.

SDSD ($K = 1$). Following SDPO implementation of Hübotter et al. (2026) where the teacher 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) per student rollout and create an independent teacher from each, then average their distillation losses. This requires K forward passes through the teacher, though it adds only 5.8% wall-clock time per training step since generating the student rollouts dominates computation.

Observing the pass@k curves in Fig. 4 we note that, in general, SDSD variants have less steep pass@k curves showing that the *functional diversity* of SDSD variants is lower than GRPO. We also report the checkpoint with the highest average accuracy for each method, averaged across 3 seeds, in Tab. 1 and show the evolution of pass@1 scores across training in Fig. 7. Overall, SDSD variants have higher pass@1 and lower pass@k indicating low diversity.

4.3. 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.

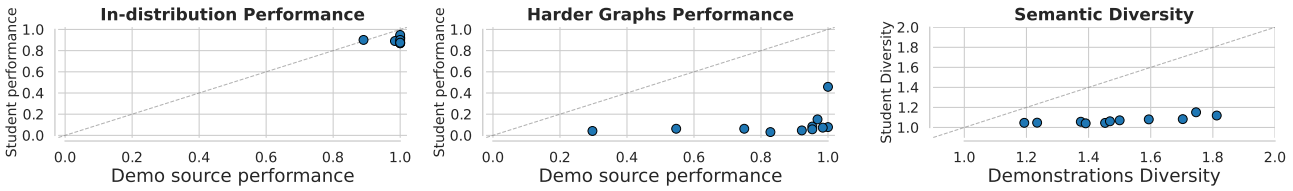
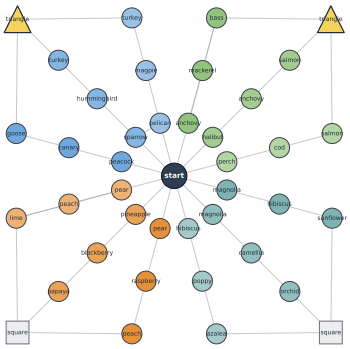
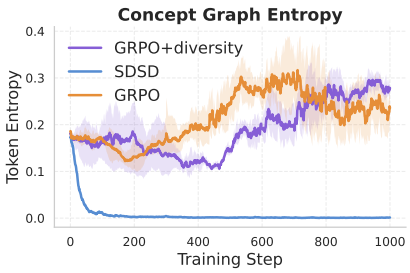


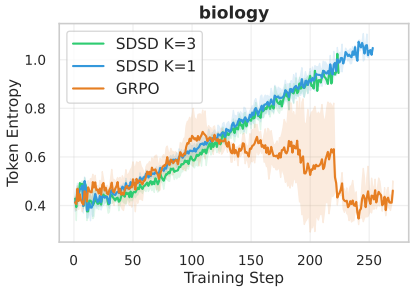
Figure 5. We train SDSD models using external demonstrations that are both correct and diverse. We use multiple datasets of demonstrations with increasing levels of diversity. Each demonstration dataset leads to SDSD models with good in-distribution performance, but low semantic diversity, regardless 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.



(a) Sample Concept Graph.



(b) Token Entropy of Concept Graph models



(c) Token Entropy of QA models.

Figure 6. **(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, c)** 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.

We obtain the demonstrations from different checkpoints of GRPO+diversity regularizer models. For each dataset, we train a SDSD model using teachers conditioned on these external demonstrations. This way we can control the diversity of the teacher demonstrations and see their influence on the diversity of the learned student.

All resulting SDSD models have high in-distribution performance, similar to the models that generated the demonstrations. Nevertheless, on the Harder Graph setup (that requires diverse exploration during training) performance of the students remains low. Moreover, we compare the semantic diversity of the demonstrations and the diversity of the resulting SDSD models in Fig. 5. 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.

4.4. Token Entropy Is Not a Sufficient Metric of Diversity

In Fig. 6b 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. 6c), 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.

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-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 teachers to give guidance, by conditioning it on privileged information, achieving strong results. As privileged information they use their own correct rollouts, demonstrations from stronger models (a setting that we also use), correct answers, or environment feedback on the student rollouts, such as runtime errors. 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 short amounts of steps (Zhao et al., 2026; Yang et al., 2026) before plateauing or decreasing. SDPO (Hübotter et al., 2026) uses two setting: 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 and on-policy training is widely documented (GX-Chen et al., 2025; 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. 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., 2025), and best-of-N fine-tuning (Chow et al., 2024) 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 score reward with this diversity reward improves the diversity of math reasoning and creative writing.

6. Discussion and Limitations

We focus specifically on the variant of self-distillation that uses sampled correct rollouts as demonstrations. We do not analyze settings where the teacher is conditioned on richer privileged signals, such as runtime errors in coding (Hübotter et al., 2026), environmental feedback, or external verifiers, where the learning dynamics may differ substantially.

Conclusion We analyzed on-policy self-distillation with sampled demonstrations (SDSD) through the lens of rollout diversity. While self-distilled models can achieve strong average accuracy, we find that they often exhibit pass@k curves with shallow or nearly flat slopes, indicating collapsed functional diversity. Theoretically, we showed that the optimal self-distillation policy is obtained by tilting the base distribution according to a pointwise conditional mutual information (PCMI) score rather than reward. Unlike standard RL objectives, this mechanism can amplify pre-existing probability imbalances among equally correct rollouts, preferentially reinforcing solutions that already align with the demonstrations and base policy. Empirically, we observed this diversity collapse across controlled graph path-finding, scientific QA, and synthetic settings.Taken together, our results suggest that average accuracy alone may provide an incomplete picture of post-training quality in self-distillation systems. Functional and semantic diversity appear to be key quantities at risk of collapse, and should therefore be explicitly monitored when evaluating or deploying SDSD-style methods.

Scope and limitations. Our theoretical analysis assumes a teacher frozen at the base policy, whereas most practical SDSD implementations, including those used in our experiments, employ an EMA teacher derived from the student itself. In addition, our derivation assumes demonstrations are sampled from the base policy, which more closely resembles OPSD (Zhao et al., 2026) and our external-demonstration graph experiments than the fully self-generated setup of Hübotter et al. (2026). In practice, both EMA teachers and self-generated demonstrations introduce additional forms of self-selection bias beyond those captured by our analysis. Finally, our derivation is presented at the sequence level; however, an analogous token-level derivation yields the same PCMI-based tilt at each next-token distribution, causing the effect to compound autoregressively along a trajectory, as discussed in Section B.3.

References

Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Garea, S. R., Geist, M., and Bachem, O. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

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

Chen, Z., Qin, X., Wu, Y., Ling, Y., Ye, Q., Zhao, W. X., and Shi, G. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

Chow, Y., Tennenholtz, G., Gur, I., Zhuang, V., Dai, B., Thiagarajan, S., Boutilier, C., Agarwal, R., Kumar, A., and Faust, A. Inference-aware fine-tuning for best-of-N sampling in large language models. *arXiv preprint arXiv:2412.15287*, 2024.

Feng, K. et al. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint*, 2024.

GX-Chen, A., Prakash, J., Guo, J., Fergus, R., and Ranganath, R. Kl-regularized reinforcement learning is designed to mode collapse. *arXiv preprint arXiv:2510.20817*, 2025.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hübotter, J., Lübeck, F., Behric, L., Baumann, A., Bagatella, M., Marta, D., Hakimi, I., Shenfeld, I., Kleine Buening, T., Guestrin, C., and Krause, A. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Kazemnejad, A., Aghajohari, M., Sordoni, A., Courville, A., and Reddy, S. Nano aha! moment: Single file "rl for llm" library. https://github.com/McGill-NLP/nano-aha-moment, 2025. GitHub repository.

Kirk, R., Mediratta, I., Nalmpantis, C., Luketina, J., Hambro, E., Grefenstette, E., and Raileanu, R. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations*, 2024.

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

Li, T., Zhang, Y., Yu, P., Saha, S., Khashabi, D., Weston, J., Lanchantin, J., and Wang, T. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025.

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

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

Nagarajan, V., Wu, C. H., Ding, C., and Raghunathan, A. 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.

Penaloza, E., Vattikonda, D., Gontier, N., Lacoste, A., Charlin, L., and Caccia, M. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741, 2023.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shenfeld, I., Damani, M., Hübotter, J., and Agrawal, P. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Wu, F., Xuan, W., Lu, X., Liu, M., Dong, Y., Harchaoui, Z., and Choi, Y. The invisible leash: Why rlvr may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang, C., Qin, C., Si, Q., Chen, M., Gu, N., Yao, D., Lin, Z., Wang, W., Wang, J., and Duan, N. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

Ye, T., Dong, L., Wu, X., Huang, S., and Wei, F. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

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

Zhao, S., Xie, Z., Liu, M., Huang, J., Pang, G., Chen, F., and Grover, A. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

A. Additional Results

A.1. Science QA: results

We show the average performance and diversity scores evolution across training steps in Fig. 7.

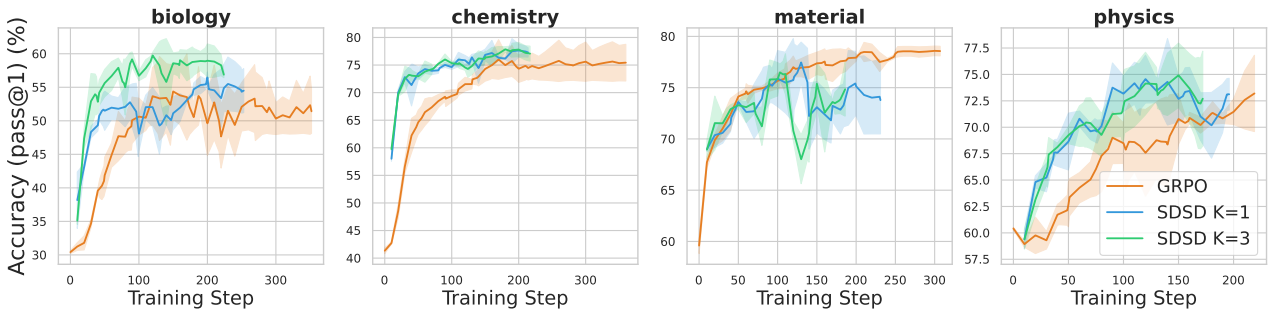


Figure 7. Pass@1 scores of SDSD and GRPO across 5h of training time.

B. Detailed Derivations and Additional Results

B.1. Standard KL-Regularized RL

We first consider the standard KL-regularized reinforcement learning objective

$$\max_{\pi} \; \mathbb{E}_{y\sim\pi(\cdot\,|\,x)}[R(y\,|\,x)] - \beta_{\mathrm{RL}}\,\mathrm{KL}(\pi(\cdot\,|\,x)\,\|\,\pi_0(\cdot\,|\,x))\,. \tag{9}$$

Proposition B.1 (Optimal policy for standard KL-regularized RL)**.** *The optimizer of* (9) *is*

$$\pi^*_{\mathrm{RL}}(y\,|\,x) \propto \pi_0(y\,|\,x)\exp\left(\frac{1}{\beta_{\mathrm{RL}}}R(y\,|\,x)\right). \tag{10}$$

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 (10). □

Remark B.2. Equation (10) 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

$$\pi_0(y \mid x, y^{\mathrm{corr}}). \tag{11}$$

The corresponding pointwise conditional mutual information (PCMI) is

$$i(y; y^{\mathrm{corr}} \mid x) := \log \frac{\pi_0(y \mid x, y^{\mathrm{corr}})}{\pi_0(y \mid x)}. \tag{12}$$

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

$$y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x) \tag{13}$$

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

$$q_{y^{\mathrm{corr}}}(y \mid x) := \pi_0(y \mid x, y^{\mathrm{corr}}). \tag{14}$$

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

$$\min_{\pi} \; \mathbb{E}_{y^{\mathrm{corr}}} \Big[\mathrm{KL}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x, y^{\mathrm{corr}})) \Big] + \beta \, \mathrm{KL}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x)) \,. \tag{15}$$

Proposition B.3 (Optimal policy for SDSD-KL)**.** *The optimizer of* (15) *is*

$$\pi^*_{\text{SD-KL}}(y \mid x) \propto \pi_0(y \mid x) \exp\left(\frac{1}{1+\beta} \, \mathbb{E}_{y^{\mathrm{corr}} \sim p_{\mathrm{corr}}(\cdot \mid x)} \big[i(y; y^{\mathrm{corr}} \mid x) \big] \right). \tag{16}$$

Proof. Fix x and suppress it in the notation. Expanding (15) gives

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

Interchanging expectation and summation,

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

The Lagrangian is therefore

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

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

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

Hence,

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

Using (12),

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

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

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

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:

$$\pi(y) \propto \pi_0(y) \exp\left(\frac{1}{1+\beta}\,\mathbb{E}_{y^{\text{corr}}}\big[i(y; y^{\text{corr}})\big]\right). \tag{17}$$

Restoring x to the notation gives (16). □

Remark B.4 (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,

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

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

Remark B.5 (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 $k > 0$. Then, by (16), their ratio under the optimal SDSD-KL policy is

$$\frac{\pi^*_{\text{SD-KL}}(y_1 \mid x)}{\pi^*_{\text{SD-KL}}(y_2 \mid x)} = k\, \exp\left(\frac{1}{1+\beta}\mathbb{E}_{y^{\text{corr}} \sim p_{\text{corr}}(\cdot \mid x)}\big[i(y_1; y^{\text{corr}} \mid x) - i(y_2; y^{\text{corr}} \mid x)\big]\right). \tag{18}$$

Equivalently,

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

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 token-level objective. We will see that exactly the same derivation 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:

$$\min_{\pi} \mathbb{E}_{y^{corr} \sim p_{corr}(\cdot|x)}\left[\text{KL}(\pi(\cdot|x, y_{<t}) || \pi_0(\cdot|x, y_{<t}, y^{corr}))\right] + \beta \text{KL}(\pi(\cdot|x, y_{<t}) || \pi_0(\cdot|x, y_{<t})) \tag{19}$$

Proposition B.6 (Optimal policy for token-level SDSD-KL)**.** *The optimizer of the token-level objective is:*

$$\pi^*_{SD-KL}(y_t|x, y_{<t}) \propto \pi_0(y_t|x, y_{<t}) \exp\left(\frac{1}{1+\beta}\mathbb{E}_{y^{corr} \sim p_{corr}(\cdot|x)}[i(y_t; y^{corr}|x, y_{<t})]\right) \tag{20}$$

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

$$i(y_t; y^{corr}|x, y_{<t}) := \log \frac{\pi_0(y_t|x, y_{<t}, y^{corr})}{\pi_0(y_t|x, y_{<t})} \tag{21}$$

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 position. 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

codeblock listing only, breakable, colback=gray!5, colframe=gray!40, arc=2mm, boxrule=0.4pt, left=6pt, right=6pt, top=6pt, bottom=6pt, listing options= basicstyle=, `breaklines=true` In the Concept Graph task, each query represents a problem of finding a path in a graph given to the LLM in context. 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 $\boxed{start, node_1name, ..., triangle}$, 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.