BREAKING SAMPLED-TOKEN TUNNEL VISION ENABLES ROBUST CAPABILITY TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) efficiently transfers capabilities acquired through costly post-training by providing dense teacher supervision along student-generated trajectories. Yet sampled-token OPD can become unreliable in cross-origin teacher–student settings. Through controlled comparisons between successful and failed training regimes, we identify *sampled-token tunnel vision*: analogous to human tunnel vision, the advantage estimator focuses on a single sampled token's teacher–student probability ratio while overlooking the update's effects on surrounding alternatives. This narrow view can obscure the suppression of teacher-preferred candidates or the amplification of less-preferred ones, undermining capability transfer. Simply aggregating reverse-KL contributions over nearby tokens can also lose directional information: the same aggregate may arise when the sampled token should be promoted in one case but suppressed in another, relative to its neighbors. We therefore propose **Neighborhood Contrastive On-Policy Distillation** (`NC-OPD`), which reconstructs the sampled-token advantage through pairwise comparisons within a lightweight neighborhood of tokens ranked above the sampled token by the teacher or below it by the student. This provides explicit relative-preference guidance while preserving the original sampled-token policy-gradient interface. Across cross-scale and cross-reasoning-mode transfer settings, `NC-OPD` consistently improves mathematical reasoning, code generation, and multi-domain performance. These results highlight neighborhood-relative advantage estimation as an effective research line to robust on-policy capability transfer.

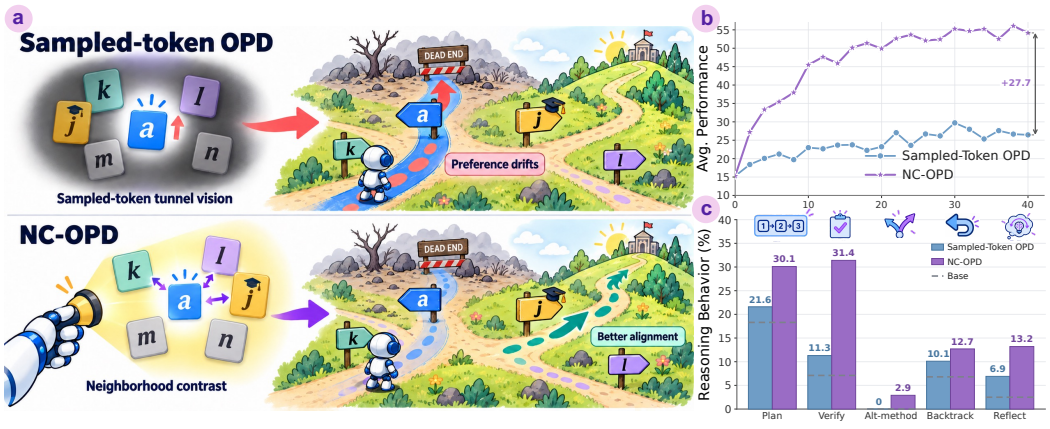

Figure 1: (a): **Sampled-token tunnel vision.** Standard sampled-token OPD determines its advantage from a single token's teacher–student probability discrepancy, overlooking the update's effects on surrounding alternatives. Consequently, reinforcing the sampled token can weaken teacher-preferred neighbors relative to it, steering the student toward less desirable generation patterns. `NC-OPD` reconstructs the advantage through neighborhood contrasts to correct such misdirected updates and support more robust capability transfer. (b): `NC-OPD` substantially improves the average math (AIME24, AIME25, AMC23) performance of `Qwen3-8B-Base` under supervision from `Qwen3-8B-Thinking-DAPO-RL`. (c): `NC-OPD` injects the thinking behaviors into the student, enabling students to use a variety of reasoning skills frequently. We followed the pattern classification in Gandhi et al. (2025) and used GPT-5.6 Terra to perform statistical analysis (prompt: Section C.1).

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.

1 INTRODUCTION

Reliably transferring capabilities across models (Hinton et al., 2015; Zhuang et al., 2020) is fundamental to scalable general intelligence (Hendrycks et al., 2025; Sun et al., 2025). Rather than repeatedly rediscovering advanced capabilities through costly post-training for each model (Kumar et al., 2026), an ideal mechanism should decouple capability discovery from dissemination, allowing skills discovered by a capable teacher to propagate efficiently across students: *explore once, transfer many*. On-policy distillation (OPD) (Agarwal et al., 2024) offers a compelling path toward this vision by transferring teacher capabilities along student-generated trajectories (Gu et al., 2024; Yang et al., 2025; Lu & Thinking Machines Lab, 2025). Specifically, the student first generates rollouts under its current policy, after which the teacher provides dense, token-level supervision through reverse-KL shaping on the prefixes the student actually visits. This replaces the outcome-level feedback commonly used in Reinforcement Learning with verifiable rewards (RLVR) (Shao et al., 2024; Yu et al., 2026) with dense learning signals, enabling the student to learn not only from its native reasoning states but also from states induced by its own mistakes (Li et al., 2026a).

While OPD has demonstrated strong results on same-origin teacher–student pairs that share a base model and pretraining pipeline (Yang et al., 2026), its efficiency can degrade sharply as the teacher and student diverge in capability and generation patterns (Li et al., 2026b; Fu et al., 2026). Recent studies mitigate this challenge through trust-region constraints (Xing et al., 2026; Xie et al., 2026), reward clipping (Ko et al., 2026), uncertainty filtering (Sun et al., 2026; Xu et al., 2026a) and entropy-aware adaptation (Jin et al., 2026; Xu et al., 2026b). These mechanisms primarily regulate where and how strongly updates are applied, mitigating the impact of unreliable advantage signals. Yet constraining an update does not, by itself, ensure that the deviation in the original advantage has been eliminated (Zhang et al., 2026; Oh et al., 2026), nor that the underlying advantage provides reliable directional guidance for redistributing probability mass among competing tokens. Such safeguards may therefore suppress harmful updates without correcting the misleading signals that hinder learning from the teacher on high-disagreement prefixes. This motivates our central question: **Can we reconstruct the advantage to provide reliable directional guidance for robust capability transfer?**

We revisit sampled-token OPD (standard OPD) and identify **sampled-token tunnel vision** as a source of unreliable updates under substantial teacher–student divergence (Figure 1(a)). In its REINFORCE-style formulation, the advantage is determined by the teacher–student log-probability ratio associated with a single sampled token, although the resulting update affects the entire vocabulary through softmax coupling. Analogous to human tunnel vision, where a restricted central view obscures peripheral information (Findley & Scott, 2006), this estimator focuses on the sampled token's probability discrepancy while overlooking the update's effects on surrounding alternatives. Consequently, an update that appears justified for the sampled token can nevertheless move its relative preference against competing tokens away from the teacher's preference structure. Our analysis identifies two characteristic consequences: (i) over-promoting a sampled token relative to alternatives preferred by the teacher, and (ii) over-suppressing a semantically valid token, thereby favoring alternatives that the student already overestimates. These effects can reinforce less desirable generation patterns and undermine capability transfer. A natural response is to incorporate neighboring tokens from the teacher's and student's rankings. However, aggregating their reverse-KL contributions can still lose directional information: the same aggregate may arise when the sampled token should be promoted in one case but suppressed in another, relative to its neighbors. Correcting the advantage therefore requires preserving how alternatives should be preferred relative to the sampled token.

To broaden the advantage estimator's field of view while explicitly grounding its information to the sampled token, we propose **Neighborhood Contrastive On-Policy Distillation** (`NC-OPD`). `NC-OPD` replaces the pointwise advantage with pairwise comparisons between the sampled token and its local competitors, directly capturing whether the student over-ranks the sampled token or under-ranks alternatives preferred by the teacher. Rather than asking *Does the teacher assign a higher probability to the sampled token than the student does?*, `NC-OPD` asks a relational question: *Does the teacher prefer the sampled token over its neighbors more than the student does?* This converts local preference information into explicit, anchor-relative directional guidance. We further analyze which categories of neighboring tokens provide useful supervision under the joint preference structures of the teacher and student. This systematic analysis yields an efficient neighborhood that retains only the informative competitors. An adaptive inhibition-of-accessibility reweighting further balances their complementary contributions to the resulting advantage calculation.

We evaluate NC−OPD across diverse settings, including cross-scale transfer and cross-reasoning-mode transfer. NC−OPD delivers strong and consistent improvements in mathematical reasoning Figure 1(b), code generation, and multi-domain capability transfer. Notably, by recasting pointwise advantage estimation as informative neighborhood-relative comparisons, NC−OPD provides stable and robust optimization guidance even under cross-domain expert supervision, enabling a smaller student to improve mathematical reasoning when guided by a code-specialized teacher. These results suggest that OPD advantage-function design is a promising research direction for on-policy capacity transfer.

2 PRELIMINARIES

On-policy distillation (OPD) trains the student on its own generated trajectories rather than on fixed teacher trajectories. A Sampled-Token OPD (standard OPD) objective minimizes the reverse KL divergence from the student policy π_θ to the teacher policy π_T:

$$\mathcal{J}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\log \pi_\theta(y \mid x) - \log \pi_T(y \mid x)\right]. \tag{1}$$

Equivalently, Equation 1 corresponds to minimizing $D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_T(\cdot \mid x))$ on student-induced trajectories. Since the trajectory is sampled from the current student, the teacher provides supervision on states that the student actually visits during training. For each sampled response y, the standard implementation of OPD uses the sequence-level log-ratio as the advantage function $A^{\mathrm{OPD}}(x, y) = \log \pi_\theta(y \mid x) - \log \pi_T(y \mid x)$ to construct a policy-gradient update:

$$\nabla_\theta \mathcal{J}_{\mathrm{sample}}(\theta) = \mathbb{E}_{x,y\sim\pi_\theta} \left[A^{\mathrm{OPD}}(x, y) \nabla_\theta \log \pi_\theta(y \mid x)\right].$$

This sample-based estimator avoids full-vocabulary KL computation and only requires the log-probabilities of the sampled trajectory under the student and teacher. However, because the supervision is tied to the sampled token, the signal can have high variance and may become ineffective when the sampled tokens are far from the teacher's preferred region.

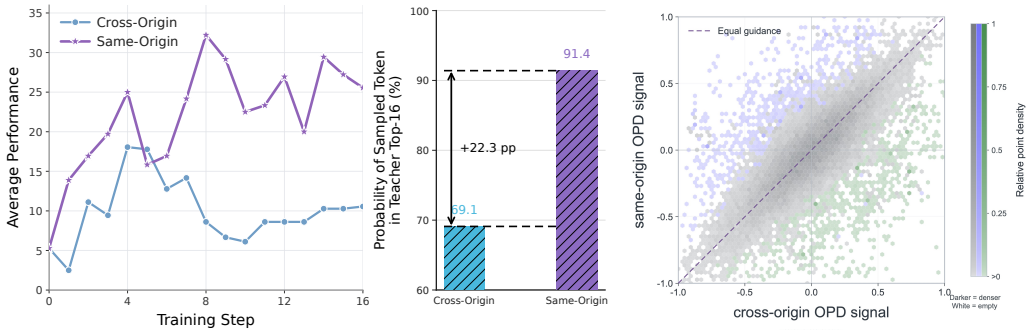

Figure 2: **Left:** OPD struggles to transfer capabilities in the cross-origin setting. **Middle:** Sampled tokens appear less frequently in the teacher's top-16 predictions, indicating reduced representativeness. **Right:** Comparison of same-origin and cross-origin advantages during 20 step training on the same data. Deviations from $y = x$ illustrate signal misalignment, with blue and green highlighting two sampled token tunnel vision cases, i.e., over-boosted suboptimal token and over-suppression a harmless token, respectively. The color intensity represents the degree of concentration.

3 UNCOVERING SAMPLED-TOKEN TUNNEL VISION

Our investigation begins with a failure in cross-origin capability transfer: We let the student π_θ (Qwen3-4b-base) distill math reasoning from the teacher π_T (Qwen3-8b), which enhanced via 400-step RL on the DAPO dataset (Yu et al., 2026). OPD fails to improve the student's mathematical reasoning performance, as shown in Figure 2 (Left).

Sampled-token advantages become directionally unreliable. To isolate the source of failure, we compare the advantages produced by the same OPD objective on the same training data under the

two regimes above: We use the successful same-origin teacher–student setting as a healthy empirical reference. If the sampled-token advantage remained reliable under divergence, the paired estimates from the two settings would concentrate around the identity line $y = x$. Instead, Figure 2 (Right) shows that they disperse substantially away from this line, accompanied by markedly higher advantage variance and frequent deviations from the healthy update direction. We trace this unreliability to the reduced representativeness of student-sampled tokens. Compared with the same-origin setting, sampled tokens under large teacher–student divergence appear far less often in the teacher's top-k predictions (Figure 2 (Middle)). A sampled token is therefore more likely to fall outside the teacher's salient preference region and can no longer serve as a faithful proxy for its local preference structure. Consequently, the pointwise density ratio at this unrepresentative anchor produces unreliable probability-redistribution signals and, ultimately, misdirected policy updates.

Sampled-token tunnel vision. To understand what drives the dominant off-diagonal deviations in advantage estimation, we conduct an LLM-assisted analysis [1] on the samples in Figure 2 (Right). This analysis reveals two recurring failure modes (rendered in blue and green). *(i) Over-boosting a teacher-suboptimal token relative to a preferred neighbor.* As illustrated in Figure 3 (Top), when $\pi_T(a_t) > \pi_\theta(a_t)$, Sampled-Token OPD assigns a_t a positive advantage, increasing its logit while suppressing all non-sampled tokens through softmax coupling. Although increasing the marginal probability of a_t may appear reasonable, the teacher may prefer an unobserved neighbor i substantially more than a_t. The update nevertheless promotes a_t relative to i, thereby suppressing the candidate that better reflects the teacher's actual preference. Because such teacher-preferred candidates may have low probability under the student, they are also less likely to be sampled and receive corrective updates later. *(ii) Over-suppressing a valid token with misdirected probability redistribution.* As illustrated in Figure 3 (Bottom), when the teacher assigns low probability to a semantically valid but stylistically different token, Sampled-Token OPD may produce an excessively negative advantage. Suppressing the sampled token releases probability mass, but it redistributes this mass among non-sampled tokens primarily according to the student's current distribution rather than the teacher's preference structure. Consequently, alternatives the student already overestimates receive the largest increases, whereas teacher-preferred but student-underestimated candidates receive little benefit. This misallocation can amplify semantic drift and training instability.

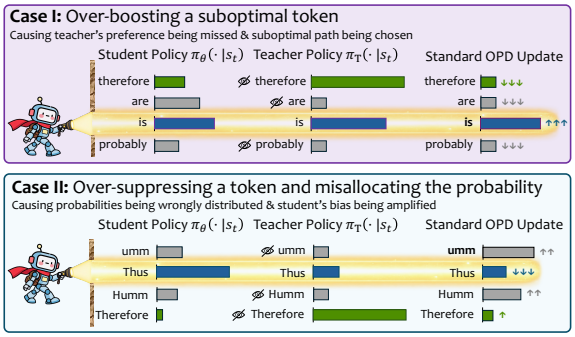

Figure 3: **Both failure modes focusing on the sampled token's probability discrepancy overlook harmful effects on surrounding candidates. Top:** A positive advantage reinforces the sampled token, causing a teacher-preferred alternative to lose ground relative to it. **Bottom:** A negative advantage penalizes a semantically valid but stylistically different token, favoring less desirable alternatives already overestimated by the student.

Despite producing opposite sampled-token updates, both cases share the same underlying cause: OPD determines the update direction solely from the sampled token's probability difference, without assessing whether the resulting changes to nearby alternatives agree with the teacher's preferences. Consequently, a seemingly justified update can suppress teacher-preferred candidates or amplify less-preferred ones through probability redistribution, steering the student toward less desirable generation patterns. We term this phenomenon *sampled-token tunnel vision*: analogous to human tunnel vision, fixation on a narrow focal cue, the sampled token's probability discrepancy—obscures consequential changes in the surrounding context.

From diagnosis to design. This diagnosis reveals a clear path toward restoring reliable sampled-token OPD. The advantage estimator should incorporate a compact set of local alternatives that expose the student's preference structure around the sampled token while including candidates that more faithfully represent the teacher's structured preference.

[1] We use GPT-5.6 Terra as the discriminator to compute the word sense category (decision related, decision not related) corresponding to the token. Combined with the automated script, the logits of each prefix before and after the vocabulary update are compared.

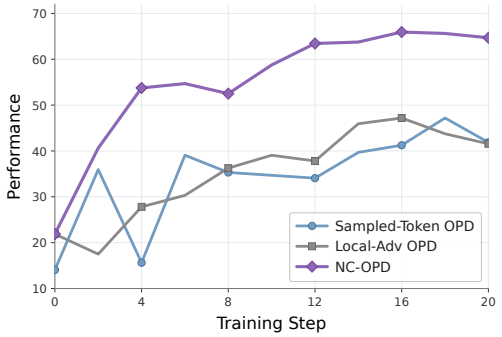

Figure 4: **AMC23 performance during training.** Local aggregation advantage provides limited gains over sampled-token OPD.

Method	$\pi_\theta(a)$	$\pi_\theta(j)$	$\pi_T(a)$	$\pi_T(j)$	Signal	Relative Update
Case 1: suppress a relative to j						
Sampled-Token OPD	0.2	0.1	0.3	0.4	+0.405	$a\uparrow$ ✗
Local Aggregation	0.2	0.1	0.3	0.4	+0.2197	$a\uparrow$ ✗
Neighbor Contrast	0.2	0.1	0.3	0.4	−0.981	$a\downarrow$ ✓
Case 2: promote a relative to j						
Sampled-Token OPD	0.3	0.2	0.2	0.05	−0.405	$a\downarrow$ ✗
Local Aggregation	0.3	0.2	0.2	0.05	−0.3989	$a\downarrow$ ✗
Neighbor Contrast	0.3	0.2	0.2	0.05	+0.981	$a\uparrow$ ✓

Table 1: **Illustrative single-neighbor cases for three types of advantage**. Correctness is defined by the π_T–π_θ relative odds of a versus j.

4 Neighborhood Contrastive Advantages Mitigate Tunnel Vision

A natural but ineffective remedy: local aggregation. A natural response to sampled-token tunnel vision is to expose the advantage estimator to more local alternatives. We construct a neighborhood around the sampled token a_t by collecting nearby tokens in both rankings: $\mathcal{N}_t^k(a_t) = \mathcal{N}_{\pi_\theta}^k(a_t) \cup \mathcal{N}_{\pi_T}^k(a_t)$, where $\mathcal{N}_{\pi_*}^k(a_t)$ contains the k closest tokens above and below a_t under the ranking induced by π_*. This neighborhood exposes the estimator to both student-preferred competitors and alternatives favored by the teacher. A straightforward extension of Sampled-Token OPD is then to aggregate the token-wise reverse-KL contributions over the local support $\mathcal{S}_t(a_t) = a_t \cup \mathcal{N}_t^k(a_t)$: $D_{\text{local}} = \sum_{j \in \mathcal{S}_t(a_t)} \pi_\theta(j \mid s_t) \log \frac{\pi_\theta(j|s_t)}{\pi_T(j|s_t)}$, $A_t^{\text{Local}} = -D_{\text{local}}$. Despite observing substantially more tokens than the sampled-token estimator, this local aggregation provides only limited improvement (Figure 4).

Why does local aggregation fail? The limitation is not insufficient local coverage, but *directional non-identifiability*. The policy-gradient update is still applied through the sampled action a_t, so the relevant question is whether a_t should be promoted or suppressed *relative to its competitors*. Consider a sampled token a and one competitor j, with all remaining probability mass unchanged (as shown in Table 1). Suppose $\pi_\theta(a) = 0.2,\ \pi_\theta(j) = 0.3,\ \pi_T(a) = 0.3,\ \pi_T(j) = 0.2$. The teacher prefers a over j more strongly than the student does, so the desired update should *increase* a relative to j. However, the local reverse-KL contribution is $D_{\text{local}} \approx 0.405$, giving $A_t^{\text{Local}} \approx -0.405$, which instead suppresses the sampled token. More importantly, reversing the two preferences, $\pi_\theta(a) = 0.3,\ \ \pi_\theta(j) = 0.2,\ \pi_T(a) = 0.2,\ \pi_T(j) = 0.3$, produces exactly the same $D_{\text{local}} \approx 0.405$ and hence the same value, although the correct anchor-relative update is now the opposite: a should be suppressed in favor of j. Thus, local aggregation can detect that the neighborhood is distributionally mismatched, but cannot identify the orientation of that mismatch with respect to the sampled token.

Neighborhood contrast advantage calculation. The example above suggests that correcting tunnel vision requires not merely aggregating more token-wise information, but preserving how each alternative is preferred *relative to the sampled anchor*. Let $s_t = (x, y_{<t})$ and $r_t(j) = A_t^{\text{OPD}}(j) = \log \frac{\pi_T(j|s_t)}{\pi_\theta(j|s_t)}$. We compare the sampled token with each neighbor through $\Delta_t(a_t, j) = r_t(a_t) - r_t(j) = \log \frac{\pi_T(a_t|s_t)/\pi_T(j|s_t)}{\pi_\theta(a_t|s_t)/\pi_\theta(j|s_t)}$. Its sign specifies whether the student's odds of a_t relative to j should increase ($\Delta_t > 0$) or decrease ($\Delta_t < 0$) to match the teacher. For the two cases, the contrasts are -0.981 and $+0.981$, respectively, recovering both relative directions. Averaging these contrasts subtracts a neighborhood baseline from the sampled-token signal, calibrating its sign and magnitude using local relative preferences.

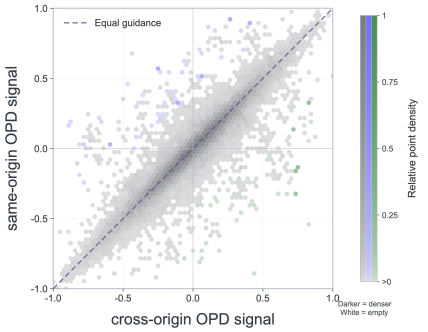

Figure 5: Compared with the original advantage calculation (Figure 2, right), `NC-OPD` obviously reduces the blue and green off-diagonal clusters associated with the tunnel-vision modes on the same data.

We define Neighborhood Contrastive advantage by averaging these anchor-relative comparisons:

$$A_t^{\text{NC-OPD}} = \frac{1}{|\mathcal{N}_t(a_t)|} \sum_{j \in \mathcal{N}_t(a_t)} \left(A_t^{\text{OPD}}(a_t) - A_t^{\text{OPD}}(j) \right)$$
$$= \underbrace{\log \frac{\pi_T(a_t \mid s_t)}{\pi_\theta(a_t \mid s_t)}}_{\text{sampled-token signal}} - \underbrace{\frac{1}{|\mathcal{N}_t(a_t)|} \sum_{j \in \mathcal{N}_t(a_t)} \log \frac{\pi_T(j \mid s_t)}{\pi_\theta(j \mid s_t)}}_{\text{neighborhood baseline}}. \tag{2}$$

This decomposition offers another interpretation of NC-OPD: the neighborhood baseline calibrates the pointwise OPD signal based on how the surrounding alternatives are jointly misranked by the student. Consequently, the sign of the resulting advantage no longer depends solely on the sampled token's density ratio, but on whether that token is over- or under-preferred relative to its local competitors. As shown in Figure 5, neighborhood contrast substantially reduces erroneous advantage signs and stabilizes advantage estimation, enabling better performance (Figure 4).

5 Efficient Neighborhood Selection for Advantage Estimation

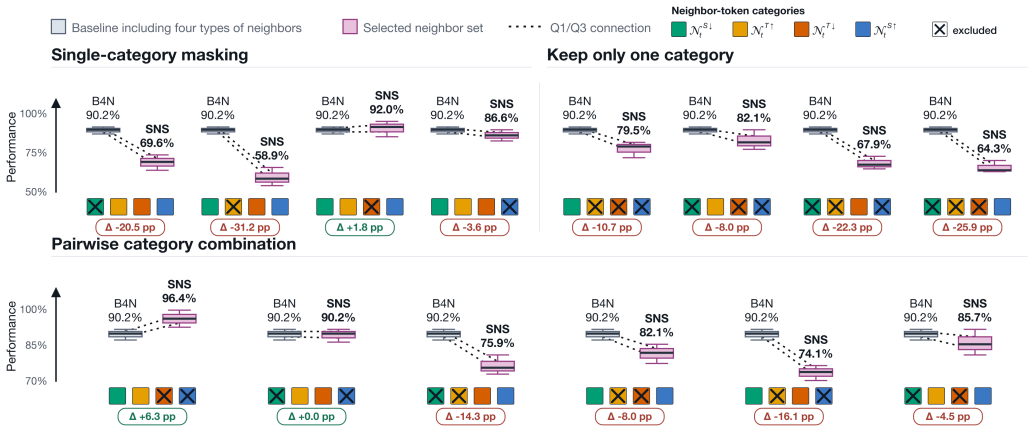

Figure 6: **Functional analysis of neighborhood token categories. Top left:** Removing one category at a time from the full neighborhood tests its marginal contribution. **Top right:** Keeping only one category evaluates its standalone utility. **Bottom:** Retaining each pair of categories tests their complementarity. Gray boxes (B4N) represent the baseline using all four categories; purple boxes (SNS) represent the selected neighborhood. Combining $\mathcal{N}_t^{S\downarrow}$ with $\mathcal{N}_t^{T\uparrow}$ achieves the best performance. Each box summarizes performance across four per-category neighborhood sizes ($k \in \{4, 8, 12, 16\}$), to eliminate the influence of the neighborhood size on the conclusion deviation.

Section 4 reconstructs the sampled-token advantage by comparing the sampled token against a local set of neighbors (alternatives) $\mathcal{N}_t^k(a_t)$ which contains the k closest tokens above and below a_t in the ranking induced by the student's preference distribution π_θ and teacher's preference distribution π_T, separately[2]. Although this design provides broad local coverage, it treats all neighboring tokens as equally useful and may therefore introduce redundant or even misleading comparisons. This raises a more specific question: *which local alternatives provide the information required to correct sampled-token tunnel vision?* We systematically analyze the functional roles of different token categories and use the resulting insights to construct an effective neighborhood for NC-OPD.

A controlled functional analysis of neighboring tokens. We first define four marginal categories according to the broad sampled token centric neighbor set $\mathcal{N}_t^k(a_t)$:

$$N_t^{S\uparrow} = \left\{ j \in \mathcal{N}_t^k(a_t) : \pi_\theta(j) < \pi_\theta(a_t) \right\}, \quad \mathcal{N}_t^{S\downarrow} = \left\{ j \in \mathcal{N}_t^k(a_t) : \pi_\theta(j) > \pi_\theta(a_t) \right\},$$

[2]If k exceeds the selectable limit, then all the tokens belonging to that category will be selected.

$$\mathcal{N}_t^{T\uparrow} = \left\{ j \in \mathcal{N}_t^k(a_t) : \pi_T(j) < \pi_T(a_t) \right\}, \quad \mathcal{N}_t^{T\downarrow} = \left\{ j \in \mathcal{N}_t^k(a_t) : \pi_T(j) > \pi_T(a_t) \right\}.$$

$S\uparrow$ ($T\uparrow$) and $S\downarrow$ ($T\downarrow$) denote tokens ranked above and below the sampled token by the student (teacher). We evaluate these categories under the setting introduced in Section 3.

$\mathcal{N}_t^{S\uparrow}$ and $\mathcal{N}_t^{T\downarrow}$ provide uninformative low-quality guidance. Figure 6 reports the single-category masking results [3]. Two categories provide little positive marginal contribution. First, removing $\mathcal{N}_t^{S\uparrow}$ does not affect performance. Because these tokens already receive high probability under the student. They are therefore more likely to appear as sampled actions, so they are extremely rare and contribute little to the advantage calculation. Second, removing $\mathcal{N}_t^{T\downarrow}$ improves performance because most teacher-below tokens are also low-ranked by the student and therefore represent choices on which the two policies already broadly agree. Their pairwise comparisons involve uninformative signal and may instead introduce estimation noise. A smaller subset in $\mathcal{N}_t^{T\downarrow}$ is informative, which are preferred by the student but disfavored by the teacher. but these tokens are more likely to receive direct correction through future on-policy sampling. Their aggregate marginal contribution is therefore limited.

$\mathcal{N}_t^{S\downarrow}$ and $\mathcal{N}_t^{T\uparrow}$ provide the critical guidance. Masking $\mathcal{N}_t^{T\uparrow}$ or $\mathcal{N}_t^{S\downarrow}$ causes a substantial performance degradation (Figure 6). Tokens in $\mathcal{N}_t^{T\uparrow}$ explicitly reveal alternatives that the teacher prefers over the sampled token. They are essential for correcting an overly optimistic advantage estimation. In particular, even when $\pi_T(a_t) > \pi_\theta(a_t)$ produces a positive value in sampled-token advantage, a candidate j, which ranks higher than a_t under teacher's preference, may satisfy $\log \frac{\pi_T(a_t)}{\pi_\theta(a_t)} - \log \frac{\pi_T(j)}{\pi_\theta(j)} < 0$, indicating that the student over-ranks a_t relative to j. Such candidates provide explicit negative evidence against incorrectly reinforcing a teacher-suboptimal token. Tokens in $\mathcal{N}_t^{S\downarrow}$ serve a complementary role. Because the student ranks them below a_t, they are unlikely to be sampled and optimized directly, yet they represent the immediate alternatives against which suppressing a_t should be evaluated. These comparisons determine whether the candidates that would benefit from decreasing a_t are actually preferred by the teacher. They can therefore prevent an excessively negative advantage from shifting probability toward alternatives the teacher considers even worse, or increase the intensity of reducing a_t so the teacher's preferred candidates inherit logits.

Cooperation of $\mathcal{N}_t^{S\downarrow}$ and $\mathcal{N}_t^{T\uparrow}$ produce a combined force. We next evaluate pairwise combinatorial test, as shown in Figure 6, the best performance is $\mathcal{N}_t^{\text{best}} = \mathcal{N}_t^{S\downarrow} \cup \mathcal{N}_t^{T\uparrow}$. Notably, this combination is more effective than using any single token set alone. Figure 7 visualizes the contribution of $\mathcal{N}_t^{T\uparrow}$ and $\mathcal{N}_t^{S\downarrow}$ to the overall advantage during the training process. $\mathcal{N}_t^{T\uparrow}$ contributes predominantly negative values, thereby preventing overestimated positive advantages. In contrast, $\mathcal{N}_t^{S\downarrow}$ contributes predominantly positive values, thereby preventing excessive suppression of the sampled token when the available student-side alternatives are not supported by the teacher. This phenomenon empirically supports the previous explanation of the complementary effect of the two token sets, and the competition between them dynamically corrects deviations from the original advantage, ensuring the effectiveness of the updated direction.

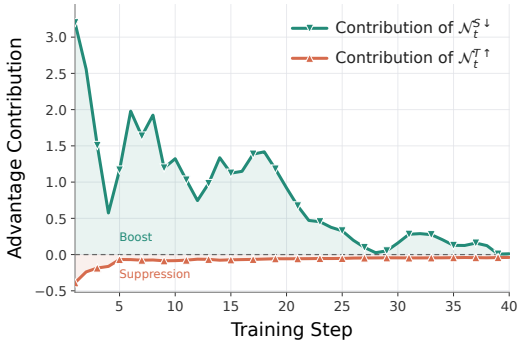

Figure 7: **Complementary neighborhood contributions to the NC-OPD advantage.** $\mathcal{N}_t^{T\uparrow}$ contribute predominantly negative signals, counteracting over-promotion of the sampled token. $\mathcal{N}_t^{S\downarrow}$ contribute predominantly positive signals, mitigating excessive suppression. Their contribution magnitudes generally decrease as training progresses.

Practically, the two neighbor subsets may contribute unevenly: teacher-preferred alternatives that the student currently assigns low probability can be underrepresented in advantage calculation, making the resulting reference overly centered on the student's existing preference. However, such neighbors can provide students with the exploration encouragement signal of whether there are options

[3] The masked tokens, even if they exist in other categories, will not be included in the advantage calculation.

that are difficult to achieve but preferred by the teacher. To counteract this bias, we introduce a normalized inverse-preference weight $w_{t,j} = \frac{\pi_\theta(j|x,y_{<t})^\alpha}{\sum_{k\in\mathcal{N}_t^{\text{best}}(y_t)} \pi_\theta(k \mid x, y_{<t})^\alpha}$, $\alpha < 0$. A mildly negative α assigns greater influence to neighbors currently underestimated by the student, while preserving the normalization $\sum_j w_{t,j} = 1$. This encourages the advantage estimator to retain informative teacher-preferred alternatives without being dominated by the student's current preference. Therefore, the optimized loss of NC-OPD can be ultimately rewritten as:

$$\mathcal{L}_{\text{NC-OPD}}(\theta) = -\mathbb{E}\left[\sum_{t=1}^{T} \text{sg}\left(\log \frac{\pi_T(y_t \mid s_t)}{\pi_\theta(y_t \mid s_t)} - \sum_{j\in\mathcal{N}_t^{\text{best}}} w_{t,j} \log \frac{\pi_T(j \mid s_t)}{\pi_\theta(j \mid s_t)}\right) \log \pi_\theta(y_t \mid s_t)\right]. \quad (3)$$

6 Experiments

Although the previous sections verified the effectiveness of NC-OPD through numerous experiments under small-sized LLMs, to comprehensively evaluate the improvement brought by NC-OPD after alleviating tunnel vision, (i) we compared our method with mainstream baselines under cross-scale and cross-reasoning-mode transfer (Section 6.2), (ii) we verified the generalization of NC-OPD in various domains such as code, multi-teacher distillation, and cross-domain expert supervision(Section 6.3).

6.1 Experimental Setup

Tasks and models. We evaluate mathematical reasoning in two settings: *Base* → *Thinking*, with `Qwen3-8B-Base` as the student and `Qwen3-8B-thinking` with 400-step RL on DAPO dataset (Yu et al., 2025) as the teacher (Yang et al., 2025); and *Small* → *Large*, with `Qwen3-8B-Base` as the student and `Qwen3-30B-A3B-Instruct-2507` as the teacher. For code generation, we use `Qwen3-30B-A3B-Instruct-2507` as teacher to guide `Qwen3-8B-Base`. Following Yang et al. (2026), the multi-teacher setting uses math and code experts obtained via domain-specific RL from student `Qwen3-4B-Base`. Following Li et al. (2026b), we evaluate cross-domain transfer by using the above coding teacher to guide math reasoning of `Qwen3-4B-Base`.

Baselines. We compare against two groups of baselines: (i) Sampled-Token OPD (standard OPD) and its advanced variant Trust Region OPD (Xing et al., 2026); and (ii) distribution-alignment methods, including Teacher Top-K, Student Top-K, and Hybrid OPD (Fu et al., 2026; Li et al., 2026a). Hybrid OPD combines the teacher's top-$K/2$ and the student's top-$K/2$ candidates. $K = 256$.

Training and evaluation. We use the default hyperparameters and hold shared settings fixed; see Appendix B for training details. For NC-OPD, each of the two neighbor subsets is capped at $k = 8$ tokens, the most stable setting in Figure 6. Maximum training lengths: 16k and 32k for evaluation. All methods are implemented based on VeRL (Sheng et al., 2024). We evaluate math reasoning on AIME24, AIME25, and AMC23 using avg@32, and code generation on HumanEval+, MBPP+ (Liu et al., 2023), and LiveCodeBench+ (Jain et al., 2025) using avg@4.

Table 2: Performance comparison under two settings. The best result is shown in bold, and the second-best result is underlined. Subscripts in Avg. denote improvements over the basic score.

	Base → Thinking				Small → Large			
Method	**AMC**	**AIME 24**	**AIME 25**	**Avg.**	**AMC**	**AIME 24**	**AIME 25**	**Avg.**
Student (Base Score)	34.75	7.09	5.00	$15.613_{+0.00}$	34.75	7.09	5.00	$15.613_{+0.00}$
Student Top-K	78.92	34.58	37.67	$50.39_{+34.78}$	57.31	28.95	24.18	$36.81_{+21.20}$
Teacher Top-K	<u>82.31</u>	<u>45.07</u>	<u>38.12</u>	<u>$55.17_{+39.56}$</u>	59.18	34.39	<u>26.42</u>	$40.00_{+24.39}$
Hybrid OPD	81.17	40.33	36.09	$52.53_{+36.92}$	<u>62.86</u>	<u>35.13</u>	26.04	<u>$41.34_{+25.73}$</u>
Sampled-Token OPD	55.31	25.39	20.83	$33.84_{+18.23}$	52.11	22.08	18.96	$31.05_{+15.44}$
Trust Region OPD	79.08	43.75	32.91	$51.91_{+36.30}$	56.24	27.31	21.62	$35.06_{+19.45}$
NC-OPD	**85.75**	**47.92**	**40.83**	**$58.17_{+42.56}$**	**68.52**	**36.11**	**28.90**	**$44.51_{+28.90}$**

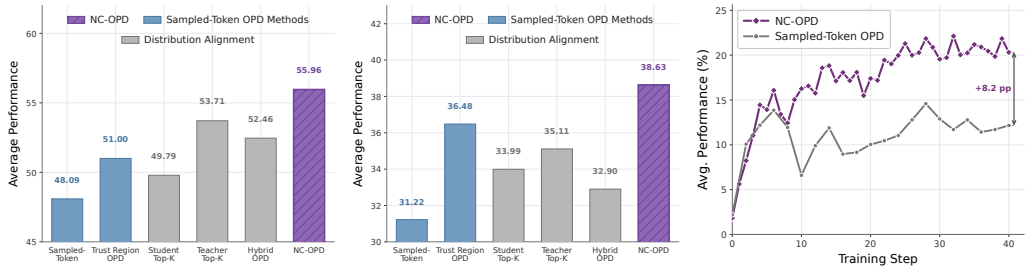

Figure 8: **Generalization across transfer settings. Left**: code-generation performance. **Middle**: performance under multi-teacher distillation. **Right**: math score under cross-domain expert supervision.

6.2 Main Results

Table 2 reports mathematical reasoning performance under both transfer settings. `NC-OPD` achieves the highest score on every benchmark and the highest average in each setting. For Base → Thinking, it reaches an average of 58.17, exceeding Sampled-Token OPD by 24.32 points and the strongest baseline, Teacher Top-K, by 3.00 points. For Small → Large, it improves the average from 31.05 to 44.51, outperforming Sampled-Token OPD by 13.46 points and the strongest baseline, Hybrid OPD, by 3.17 points. These results support the effectiveness of `NC-OPD`.

6.3 Generalization Across Transfer Settings

Code generation. As shown in Figure 8 (left), `NC-OPD` achieves the highest average score of 55.96, improving on Sampled-Token OPD (48.09) by 7.87 points and the strongest baseline, Teacher Top-K (53.71), by 2.25 points. This extends the gains beyond mathematical reasoning to code generation.

Multi-teacher distillation. We next investigate NC-OPD under multi-teacher distillation; the two domain teachers, i.e., math and code, are the 400-step RL variants of student `Qwen3-4B-Base`. `NC-OPD` achieves the highest average score of 38.63, exceeding Sampled-Token OPD by 7.41 points and the strongest baseline, Trust Region OPD, by 2.15 points (Figure 8, middle). These results indicate that neighborhood contrast also supports capability integration from multiple domain experts.

Cross-domain expert supervision. We use a code-specialized teacher, i.e., `Qwen3-4B-Base` with 400-step RL on Code data used in (Zhao et al., 2026), to guide math reasoning of `Qwen3-4B-Base`. Figure 8 (right) reports the average performance on three benchmarks, and `NC-OPD` outperforms the baseline. This result suggests that `NC-OPD` can achieve stable learning and extract useful supervision even when the teacher's specialization differs from the target pattern.

7 Conclusion

We identify sampled-token tunnel vision as a key limitation of OPD under substantial teacher–student divergence: a single-token density ratio can misrepresent the teacher's local preferences and misdirect probability redistribution. To address this limitation, `NC-OPD` constructs the sampled-token advantage through pairwise comparisons within a lightweight asymmetric neighborhood, providing anchor-relative directional guidance while preserving the original single token policy-gradient interface. Experiments across cross-scale, cross-reasoning-mode, and cross-domain expert supervision settings demonstrate its effectiveness in mathematical reasoning, code generation, and multi-domain capability transfer. Our findings highlight advantage-function design as a worthwhile and novel research direction for reliable Sampled-Token OPD.

Limitations. Despite broad empirical validation, formal guarantees for the optimization stability and capability transfer of `NC-OPD` remain to be established. Future work will investigate these theoretical foundations and extend evaluation to more model families, scales, and transfer settings to further establish the generality and limits of our findings.

AI USAGE STATEMENT

We used large language models to improve the grammar, clarity, and conciseness of the manuscript; we also utilized powerful AI to analyze large-scale data. The authors reviewed all AI-assisted revisions to ensure they accurately reflected the intended scientific content. The authors take full responsibility for the final manuscript, including its claims, results, and conclusions.

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 *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Keith Findley and Michael S. Scott. The multiple dimensions of tunnel vision in criminal cases. 2006. URL https://api.semanticscholar.org/CorpusID:147146926.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=QGJ9ttXLTy.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Dan Hendrycks, Dawn Song, Christian Szegedy, Honglak Lee, Yarin Gal, Erik Brynjolfsson, Sharon Li, Andy Zou, Lionel Levine, Bo Han, et al. A definition of agi. *arXiv preprint arXiv:2510.18212*, 2025.

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

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-Aware On-Policy Distillation of Language Models. *arXiv preprint arXiv:2603.07079*, 2026. URL https://arxiv.org/abs/2603.07079.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026a.

Zhaoyi Li, Deyang Kong, Yuan Wei, Evan Yang, Ranran Shen, Mahardika Krisna Ihsani, Ming Yang, Wei Zhang, Chuan Hao, Jian Yang, et al. Every coin has two sides: On the dual nature of generalization in on-policy distillation of large language models. *arXiv preprint arXiv:2608.16647*, 2026b.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Kevin Lu and Thinking Machines Lab. On-policy distillation. Thinking Machines Lab: Connectionism, 2025. URL https://thinkingmachines.ai/blog/on-policy-distillation.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. KL for a KL: On-Policy Distillation with Control Variate Baseline. *arXiv preprint arXiv:2605.07865*, 2026. URL https://arxiv.org/abs/2605.07865.

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.

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

Lillian Sun, Emma Yang, and Arif Kerem Dayi. Capability transfer from large to small models with synthetically-generated data. In *Tiny Titans: The next wave of On-Device Learning for Foundational Models (TTODLer-FM)*, 2025. URL https://openreview.net/forum?id=itjZv85LGk.

Yang Sun, Lichao Ma, Houyuan Qin, Yuxin Liu, Hanyang Lu, Yao Zhu, Pinlong Cai, and Guohang Yan. REOPD: Reliability-Adaptive Reward Extrapolation for On-Policy Distillation. *arXiv preprint arXiv:2608.11698*, 2026. URL https://arxiv.org/abs/2608.11698.

Zhengpeng Xie, Li Lyna Zhang, Zeke Xie, and Mao Yang. Trust region policy distillation. *arXiv preprint arXiv:2607.04751*, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust Region On-Policy Distillation. *arXiv preprint arXiv:2606.01249*, 2026. URL https://arxiv.org/abs/2606.01249.

Haoran Xu, Hongyu Wang, Yifei Gao, Jiaze Li, Xiaofeng Zhang, and Xiaosong Yuan. SG-OPD: Sign-Gated On-Policy Distillation via Sign-Consistency Gating and Phased Teacher Sampling. *arXiv preprint arXiv:2606.09304*, 2026a. URL https://arxiv.org/abs/2606.09304.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token Importance in On-Policy Distillation. *arXiv preprint arXiv:2604.14084*, 2026b. URL https://arxiv.org/abs/2604.14084.

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.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond Teacher: Generalized On-Policy Distillation with Reward Extrapolation. *arXiv preprint arXiv:2602.12125*, 2026. URL https://arxiv.org/abs/2602.12125.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Yuheng Zhang, Chenlu Ye, Shuowei Jin, Changlong Yu, Wei Xiong, Saurabh Sahu, and Nan Jiang. Rethinking importance sampling in llm policy optimization: A cumulative token perspective. *arXiv preprint arXiv:2605.07331*, 2026.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *Advances in Neural Information Processing Systems*, 38:105816–105879, 2026.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

A Related Work

A.1 On-Policy Distillation

On-policy distillation (OPD) (Agarwal et al., 2024; Gu et al., 2024; Yang et al., 2026; Lu & Thinking Machines Lab, 2025) trains a student on its own sampled trajectories while using the teacher's token-level likelihoods as dense supervision, thereby reducing the exposure bias of conventional off-policy distillation (Gu et al., 2024; Ko et al., 2024). However, recent analyses show that OPD relies on sufficient teacher–student distributional compatibility. However, when student and teacher models come from different origins, student-generated prefixes may lie outside the teacher's effective support, suppressing an unreliable update does not change the update direction targeting suboptimal reasoning behavior, nor does it make its underlying guidance more informative (Li et al., 2026a;b). This mismatch can make sampled-token OPD ineffective even when the teacher is stronger.

Recent studies have explored several ways to improve the reliability of sampled-token OPD under teacher–student distribution mismatch. Trust-region methods restrict OPD to regions where the teacher is considered reliable (Xing et al., 2026; Xie et al., 2026). REOPOLD relaxes the strict imitation objective of Sampled-Token OPD (standard OPD) by selectively applying reward clipping and entropy-based token sampling (Ko et al., 2026). Reliability-aware methods further adapt the teacher signal at the token level (Sun et al., 2026; Xu et al., 2026a). EOPD augments reverse KL with forward KL when the teacher distribution has high entropy (Jin et al., 2026), whereas TIP identifies informative tokens using student entropy and teacher–student divergence, including low-entropy tokens with high divergence (Xu et al., 2026b). These mechanisms primarily regulate the magnitude or reliability of an update, reducing damage from noisy advantages (Zhang et al., 2026). In this work, we take a more fundamental approach: we redesign the advantage function itself to transform noisy supervision under distribution mismatch into a directionally effective learning signal.

B Training Hyperparameters

We initialize the student policy from `Qwen3-8B-Base` and use `Qwen3-8B-TIR-DAPO` as the teacher. Since the student is a base model, we apply a Qwen3-compatible thinking chat template and use the tokenizer associated with `Qwen3-1.7B-SFT`. We verify that its token IDs are aligned with those of the teacher. Both `<|im_end|>` and `<|endoftext|>` are treated as valid end-of-sequence tokens.

Training data and sequence lengths. Training is performed on the level-6 subset of DeepMath-103K, containing 57,046 examples after prompt-length filtering. We shuffle the training set with random seed 42. The global rollout batch size and PPO mini-batch size are both 1,024, with a per-GPU micro-batch size of 1. Prompts longer than 2,048 tokens are removed, and the maximum generated response length is 16,384 tokens. The maximum token budget used by each actor update and by the rollout engine is 32,768 tokens. Each training prompt produces one on-policy response. Responses are sampled with temperature 1.0 and top-$p = 1.0$, without top-k truncation.

Rank-neighborhood on-policy distillation. Let q_θ and p_T denote the student and teacher distributions, respectively, and define the token-level log-ratio

$$r_i = \log q_\theta(i) - \log p_T(i). \tag{4}$$

For every sampled token a, we construct a candidate pool from the student's top-16 tokens and the first 60 tokens in the teacher ranking. The rank neighborhood contains the eight closest candidates ranked below a by the student, no student-ranked candidates above a, and all teacher candidates satisfying $p_T(i) > p_T(a)$. Duplicate candidates between the student and teacher channels are removed.

The normalized neighborhood weight is

$$\bar{w}_i = \frac{q_\theta(i)^{-0.15}\,|r_i - r_a|}{\sum_{j\in\mathcal{N}(a)} q_\theta(j)^{-0.15}\,|r_j - r_a|}, \tag{5}$$

and the rank-neighborhood advantage is computed as

$$A_{\mathrm{RN}}(a) = \sum_{i\in\mathcal{N}(a)} \bar{w}_i r_i - r_a. \tag{6}$$

Thus, the update compares the sampled token against nearby student-ranked alternatives and teacher-preferred alternatives. We do not renormalize the student or teacher distributions over the truncated candidate pool. Auxiliary rank losses, entropy regularization, explicit KL penalties, reward-level KL penalties, and confidence-based advantage shrinkage are disabled.

To reduce excessive supervision near the end of long reasoning trajectories, we apply a position-dependent multiplier for all the methods. The multiplier is one over the first half of each response and then decreases linearly to zero at its final valid token:

$$\lambda(f) = \begin{cases} 1, & f \le 0.5, \\ 2(1-f), & f > 0.5, \end{cases} \tag{7}$$

where $f \in [0, 1]$ is the normalized token position. The final advantage is $\lambda(f)A_{\mathrm{RN}}(a)$.

Optimization and distributed training. We optimize the student using AdamW with a constant learning rate of 1×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01, and gradient-norm clipping at 1.0. No learning-rate warmup is used. We perform one PPO epoch per rollout batch, use a clipping range of 0.2, and aggregate the policy loss by taking the mean over valid response tokens. Token-level importance-sampling correction is enabled with an upper threshold of 5.0.

Training uses bfloat16 precision, gradient checkpointing, FlashAttention, and FSDP on a single node with eight GPUs. Actor parameters and optimizer states are not offloaded, whereas the teacher parameters are offloaded when inactive. Rollout generation is implemented with vLLM using tensor parallelism of four and a GPU-memory-utilization target of 0.6.

Optional neighborhood-confidence calibration. NC-OPD generally works without additional advantage calibration. However, under extremely large teacher–student discrepancies, the log-ratios within a token neighborhood can become highly heterogeneous, making the estimated neighborhood advantage noisy. We therefore suggest using an optional confidence coefficient based on the uncertainty within the neighborhood. Specifically, let $V_{\mathcal{N}}$ denote the weighted variance of the neighborhood log-ratios and let $\mathrm{ESS}_{\mathcal{N}} = 1/\sum_i \bar{w}_i^2$ denote the effective neighborhood size. We compute

$$c(a) = \frac{A_{\mathrm{NC}}(a)^2}{A_{\mathrm{NC}}(a)^2 + \lambda_{\mathrm{conf}} V_{\mathcal{N}}/\mathrm{ESS}_{\mathcal{N}} + \epsilon}, \qquad \widetilde{A}_{\mathrm{NC}}(a) = c(a)A_{\mathrm{NC}}(a). \tag{8}$$

The coefficient approaches one when the neighborhood provides a strong and consistent preference signal, while shrinking weak advantages supported by highly inconsistent neighbors. Since $c(a) \ge 0$, this operation changes only the magnitude, rather than the direction, of the NC-OPD update. This mechanism is not required in ordinary settings and is disabled by default. It is intended only as a stabilization mechanism for extreme teacher–student mismatch, where suppressing uncertain updates allows the progressive gains of NC-OPD to emerge more reliably over training.

Validation. We evaluate on AIME 2024, AIME 2025, and AMC 2023 before training and every two optimization steps. For each validation problem, we sample eight responses with temperature 1.0 and top-$p = 1.0$, allowing at most 30,720 generated tokens. Checkpoints are saved every 50 optimization steps. The configured training schedule consists of three epochs, or 165 optimization steps.

C Prompt Templates

C.1 Prompts for Training and Inference

We use the same task-specific prompt format during on-policy training and evaluation. This avoids introducing a distribution shift caused by different instructions at training and inference time. All prompts are wrapped using the Qwen3 chat template with thinking mode enabled.

Mathematical reasoning. For mathematical problems, the original problem statement is followed by an instruction requesting explicit reasoning and a boxed final answer:

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

This template is used for the DeepMath training examples and for the AIME and AMC evaluation sets. The boxed-answer constraint enables deterministic extraction and rule-based verification of the final answer.

Code generation. For programming problems, we retain the complete problem statement, including its input format, output format, constraints, and examples. We then append:

````
Write Python code to solve the problem.  Present the code in
```python
Your code
```
at the end.
You need to think first then write the Python code.
````

The model is therefore encouraged to reason about the algorithm before producing its implementation, while the final Markdown code block allows the solution to be extracted and executed automatically. The same template is used for both code-domain training and evaluation.

C.2 Reasoning-Pattern Analysis

To analyze how different training objectives affect the structure of generated reasoning, we measure the occurrence of several interpretable reasoning patterns. Importantly, this analysis does not use an LLM-as-a-judge prompt. Instead, we employ a fixed, case-insensitive lexical matcher, making the analysis deterministic and avoiding potential evaluator-model bias.

We consider the following five pattern families:

- **Planning and decomposition**: cues such as *first*, *next*, *then*, *step*, *we need*, and *to solve*;
- **Verification**: *check*, *verify*, *make sure*, *confirm*, *double-check*, and *recheck*;
- **Alternative methods**: *alternatively*, *another way*, *another method*, *different approach*, and *using equations*;
- **Backtracking or uncertainty**: *wait*, *hmm*, *actually*, *mistake*, *wrong*, *hold on*, and expressions beginning with *let me re-*;
- **Reflection and meta-reasoning**: *I think*, *I believe*, *confident*, *therefore*, *thus*, and *makes sense*.

Generated responses can occasionally contain repeated terminal answers. Consequently, directly counting lexical cues may cause long repetition loops to dominate the denominator. We therefore construct a deduplicated reasoning body by retaining only the first occurrence of each non-empty response line. Pattern frequencies are computed on this body and reported as occurrences per 1,000 words:

$$F_k = \frac{1000\, N_k}{\max(N_{\text{words}}, 1)}, \tag{9}$$

where N_k is the number of matched cues from pattern family k. Repetition behavior is measured separately using the fraction of repeated non-empty lines, the maximum repetition count of a single line, and the number of boxed-answer occurrences. These lexical measurements characterize observable response behavior rather than the semantic correctness of the reasoning process.