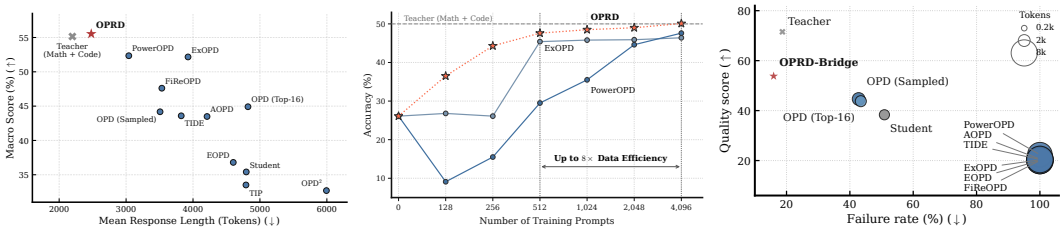
OPRD: ON-POLICY REPRESENTATION DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) has emerged as a key technique in LLM post-training, where the student generates responses and a frozen teacher provides next-token supervision at each prefix. Such guidance offers an opportunity to pass on more than correct answers, helping the student learn how the teacher reasons and behaves. Yet output-space OPD communicates only next-token probabilities: these tell the student what the teacher would say next, but matching those choices may not be enough to acquire its broader behavior. To provide richer guidance, we introduce **On-Policy Representation Distillation (OPRD)**, a framework that uses the teacher's already-computed hidden states to directly supervise the student on its own responses. When models share representation coordinates, we align their normalized states directly, an approach we call **OPRD-Vanilla**. However, distillation often involves teachers and students with different architectures, making direct comparison of their hidden states difficult. Luckily, the Platonic Representation Hypothesis suggests that different models can capture similar underlying structure despite expressing it in different coordinates. Motivated by this hypothesis, we develop **OPRD-Bridge** to first learn a low-rank mapping that makes their representations comparable, then distill through this fixed mapping. Our experiments show that *i)* in multi-teacher RL model merging, one of the most important real-world application of OPD, OPRD-Vanilla combines mathematics and code specialists into one student that matches their combined average accuracy, exhibits teacher-like behavior, like concise responses, and achieves up to $8\times$ greater training-data efficiency; and *ii)* on open-ended tasks, where accuracy alone cannot capture abilities such as empathy, OPRD-Bridge substantially outperforms OPD-style methods in transferring these abilities from larger models. Together, these results show that aligning internal representations helps students learn both what teachers can solve and how they behave.

1 INTRODUCTION



(a) Performance–length efficiency. (b) Training-data scaling. (c) Emotional-support transfer.

Figure 1: **OPRD transfers teacher capabilities and behavior efficiently.** **(a)** OPRD-Vanilla approaches the composite teacher's macro score with concise responses on Math (AIMO, AIME24, MATH-500, Minerva, OlympiadBench) and Code (HumanEval, MBPP, LeetCode, LiveCodeBench); upper left is better. **(b)** It achieves up to $8\times$ greater training-data efficiency than the two strongest baselines on PRIME (Cui et al., 2025), measured over Math (AMC/AIME, AoPS, CN K–12, Olympiads, synthetic AMC/math) and Code (APPS, CodeContests, Codeforces, TACO). **(c)** OPRD-Bridge improves response quality while reducing failures and response length over OPD-style methods on KardiaBench (Yuan et al., 2026) in 7B-to-1.5B transfer. Bubble area monotonically encodes mean response length; upper left and smaller are better.

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.

On-policy distillation (OPD) has rapidly emerged as a key technique in modern large-language-model post-training (Gu et al., 2024; Agarwal et al., 2024; Yang et al., 2025; Xiao et al., 2026; Zeng et al., 2026; Xu et al., 2026). Unlike traditional off-policy distillation, which trains on fixed teacher responses, OPD lets the current student generate a response and asks a frozen teacher to supervise every visited prefix. It thereby follows the student's evolving distribution, reduces exposure mismatch (Bengio et al., 2015), and ensures that the teacher and student process the same trajectory.

With the teacher guiding every prefix, it is natural to expect more than a correct final answer. We would like the student to inherit the teacher's knowledge along with how it works through a problem and expresses its reasoning. Yet this guidance of OPD reaches the student only through superficial next-token probabilities. These tell the student which continuation the teacher prefers, but matching local choices may still fall short of learning the teacher's broader behavior. Our controlled experiments make this gap concrete: although output-space OPD improves accuracy early in training, the student's response lengths remain closer to its own starting point than to the teacher's. Better answers, then, do not necessarily come with behavior closer to the teacher's. This gap matters especially in open-ended tasks, where response quality cannot be captured by accuracy alone. This contrast motivates us to look beyond answer correctness and seek richer supervision that brings the student's behavior closer to the teacher's as it learns the teacher's true capabilities.

To pursue this goal, we take inspiration from recent advances in latent reasoning and looped Transformers, which have drawn attention to reasoning in continuous hidden space (Hao et al., 2024; Geiping et al., 2026). These approaches show that useful reasoning can take place without expressing every intermediate step as text, encouraging us to look beyond next-token predictions for teacher guidance. We therefore turn to a source of supervision that the teacher already computes but existing OPD leaves unused: its hidden states. Building on this idea, we introduce **On-Policy Representation Distillation (OPRD)**, a framework that uses these states as explicit supervision. In this framework, the student still generates its own responses, and the frozen teacher still processes the same prefixes. However, instead of discarding the teacher's hidden states, OPRD uses them as direct targets for the student's representations. This simple change allows us to retain OPD's on-policy training while making fuller use of the teacher's computation. Through theoretical analysis and extensive experiments, we study two ways to use this supervision: (1) it provides a direct way to transfer teacher representations, improving performance and training-data efficiency when the states are directly comparable; and (2) when direct alignment is unavailable, a learned mapping allows the student to benefit from the representations of a larger or differently structured teacher.

Concretely, when teacher and student share the same architecture, we can simply align their normalized hidden states, an approach we call **OPRD-Vanilla**. This setting is common in modern LLM post-training pipelines, such as those of DeepSeek (Xu et al., 2026), GLM (Zeng et al., 2026), Instella-MoE (Liu et al., 2026a), and MIMO (Xiao et al., 2026), where multiple domain specialists are trained from a common base model. Because these specialists share a representation space, one student can learn directly from each teacher's hidden states. Our controlled single-teacher experiments first establish the benefit of this simple approach: under otherwise identical settings, OPRD-Vanilla continues to improve after sampled-token and top-k OPD saturate or decline, while its response lengths move closer to the teacher's (Figure 3). Building on this result, we apply OPRD-Vanilla to multi-teacher RL model merging, combining mathematics and code specialists into one student that matches their combined average accuracy with concise responses. It also exceeds the strongest output-space baselines with up to $8\times$ greater training-data efficiency (Figures 1a and 1b).

Despite OPRD-Vanilla is simple and effective, we often want to transfer capabilities from a larger teacher to a smaller student, whose hidden states cannot be aligned in the same direct way. Luckily, the Platonic Representation Hypothesis (Huh et al., 2024; Kornblith et al., 2019; Moschella et al., 2022; Merullo et al., 2022) suggests that learning a mapping between their representations could make hidden-state supervision possible even without shared coordinates. We therefore introduce **OPRD-Bridge**, which learns a low-rank bridge between teacher and student representations while keeping both models frozen. We then freeze this bridge and train only the student on newly generated responses. The fixed bridge anchors representation alignment as the student's responses evolve.

In summary, we conduct extensive evaluations across **24 task datasets**, covering ① *in-distribution* and ② *out-of-distribution* settings, as well as ③ *rule-verifiable* and ④ *open-ended* tasks. Across this diverse range of evaluations, our results highlight three benefits of representation supervision: *1)* OPRD nearly closes the single-teacher gap and raises the multi-teacher merging benchmark average

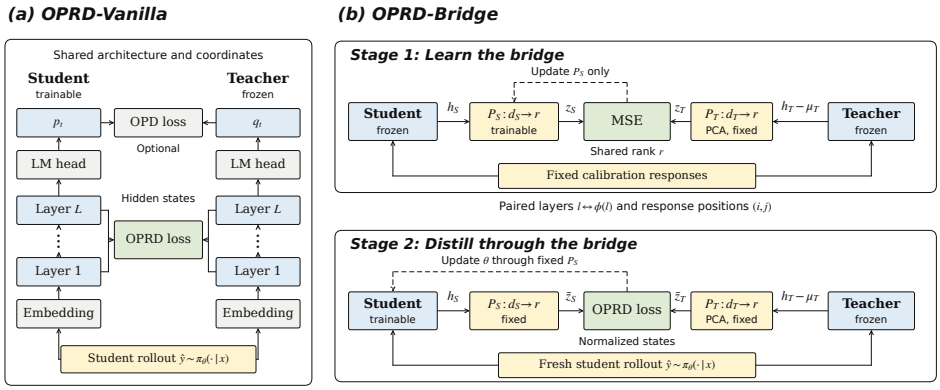


Figure 2: **Two forms of on-policy representation distillation.** Both use student rollouts and frozen teachers. **(a)** OPRD-Vanilla directly aligns normalized hidden states. **(b)** OPRD-Bridge first learns a low-rank interface, then freezes it for distillation.

from 35.4% to 55.5%, matching the combined teachers' performance with concise, teacher-like responses. *2)* These gains come with up to $8\times$ greater training-data efficiency than the strongest output-space baselines and extend beyond the training domains, with OPRD achieving the highest aggregate Out-of-Distribution (OOD) score among the compared distilled models. *3)* OPRD-Bridge enables larger-teacher transfer for open-ended abilities that accuracy alone cannot measure. On the emotional-support KardiaBench (Yuan et al., 2026), it yields higher-quality, more usable, and more concise responses than OPD-style methods, showing the value of learning both teacher capabilities and behavior via representation distillation.

2 On-Policy Representation Distillation

As a core technique in LLM post-training, OPD lets students learn from teacher feedback on their own responses. The same training loop also gives us access to the teacher's hidden states, offering a natural way to extend supervision beyond next-token probabilities. To explain how OPRD uses these states, we first revisit output-space OPD (Section 2.1), then introduce direct hidden-state alignment and its properties (Section 2.2), and finally extend it to heterogeneous models through a shared-space bridge (Section 2.3).

2.1 Revisiting Output-Space OPD

Let π_θ be the trainable student and π_T the frozen teacher. Given a prompt x, the student samples a response $\hat{y} \sim \pi_\theta(\cdot \mid x)$. Both models then evaluate each prefix $(x, \hat{y}_{<t})$, giving next-token probabilities $p_t(v) = \pi_\theta(v \mid x, \hat{y}_{<t})$ and $q_t(v) = \pi_T(v \mid x, \hat{y}_{<t})$, where v is a token in their shared vocabulary $\mathcal{V}$. To compare them, write $u_t(v) = \log p_t(v) - \log q_t(v)$. We also define $S_{t,k} = \mathrm{TopK}(p_t, k)$ and $p_t^{(k)} = p_t(\cdot \mid S_{t,k})$, the student distribution normalized over its k most likely tokens. The three per-position objectives are

$$\ell_t^{\text{sample}} = u_t(\hat{y}_t), \quad \hat{y}_t \sim p_t, \qquad \ell_t^{\text{top-}k} = \mathbb{E}_{V \sim p_t^{(k)}}[u_t(V)], \qquad \ell_t^{\text{full}} = \mathbb{E}_{V \sim p_t}[u_t(V)]. \tag{1}$$

The full-vocabulary objective equals $D_{\mathrm{KL}}(p_t \| q_t)$, while sampled-token OPD estimates this quantity from a single generated token. Its score-function gradient estimator, $g_{\text{sample}} = u_t(\hat{y}_t)\nabla_\theta \log p_t(\hat{y}_t)$, therefore depends on which token is sampled, introducing noise into the update. Top-k OPD uses more of the distribution by summing over the retained tokens, but introduces truncation bias by excluding the remaining vocabulary. Full-vocabulary OPD avoids both sampling and truncation, but computing the loss over the entire vocabulary at every response position incurs substantial computational and memory costs. Despite these differences, all three use only output probabilities, so their guidance remains focused on what the teacher would say next. As a result, matching these local predictions may lead to superficial imitation without fully transferring how the teacher processes the context. Yet the teacher has already computed hidden states to produce those predictions, giving us access to extend teacher guidance beyond its output without changing the on-policy training loop.

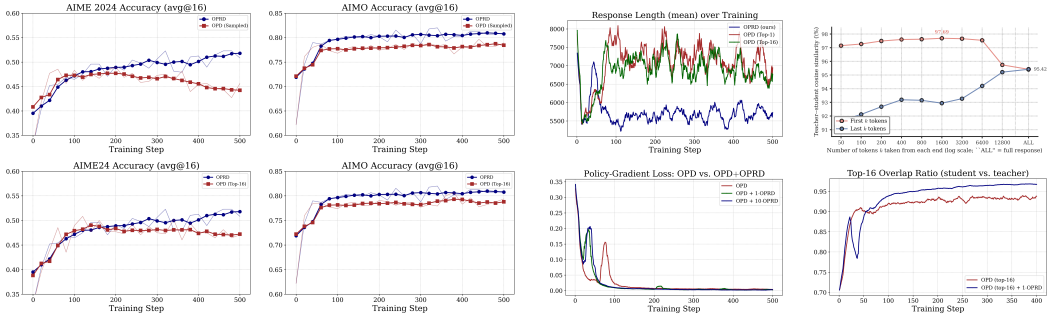


Figure 3: **Controlled single-teacher dynamics and diagnostics.** Left: single-teacher Avg16 on AIME24 and AIMO, comparing OPRD with sampled-token OPD (top) and top-16 OPD (bottom). Right, clockwise from top left: rollout length, positional cosine, top-16 overlap, and pg loss.

2.2 OPRD-Vanilla: Direct Alignment on Student Trajectories

To see why this additional guidance may matter, we first look at what output-space OPD achieves in a controlled single-teacher setting. Here, sampled-token and top-16 OPD initially improve accuracy, but this progress later slows or reverses. Meanwhile, response lengths remain closer to the initial student's than to the teacher's (Figure 3). In other words, the student learns to answer more accurately without necessarily learning to respond more like its teacher. To narrow this gap, we turn to the teacher's hidden states, aiming to guide not only the student's superficial next-token choices but also the deeper representations behind them. To do so, we need to establish which teacher states should guide which student states. We start with the simplest setting where models that share an architecture, tokenizer, and initialization, as when RL training turns one base model into several specialists. Because they share this starting point, we can directly align their hidden states at corresponding layers and positions, an approach we call **OPRD-Vanilla** (Figure 2). Like earlier representation distillation methods, this approach compares internal model features (Sanh et al., 2019; Jiao et al., 2020; Sun et al., 2020; Wang et al., 2020); here, however, the states being compared come from the current student's own responses, so the guidance follows what the student actually generates.

Concretely, the student generates a response, and the frozen teacher processes the same prefixes. At layer l and response position t, this gives paired states $h^{(l)}_{\theta,t}, h^{(l)}_{T,t} \in \mathbb{R}^d$, each used to predict the next response token. Since their magnitudes may differ, we normalize each state as $\bar{h} = h/\max(\|h\|_2, \varepsilon)$ and compare directions rather than scales. We then select layers $\mathcal{L}$ and response positions using $m_t \in \{0, 1\}$, with $\sum_t m_t > 0$, and average the mean-squared distances between the selected pairs:

$$\mathcal{L}_{\mathrm{OPRD}}(\theta) = \mathbb{E}_{x,\hat{y}\sim\pi_\theta}\left[\frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}}\Big(\sum_t \frac{m_t}{d}\left\|\bar{h}^{(l)}_{\theta,t} - \mathrm{sg}\left(\bar{h}^{(l)}_{T,t}\right)\right\|_2^2\Big)\Big/\Big(\sum_t m_t\Big)\right], \tag{2}$$

Here sg stops gradients through the teacher. The supervision follows the student's evolving responses, just as in OPD, but now directly guides the deeper representations behind its predictions.

Returning to the same single-teacher comparison, we find that this change helps the student make further progress toward the teacher (Figure 3). Beyond accuracy, representation alignment also narrows the gap near response endings, produces more concise responses, and increases teacher–student top-16 overlap, bringing both internal states and output behavior closer to the teacher's. To understand how this supervision differs, we look at what happens at a fixed prefix. Because the teacher's hidden vector is already available there, OPRD has two useful properties: ① it requires neither a token draw nor vocabulary truncation; and ② it guides the student's states directly rather than relying on output probabilities to reveal their differences. We examine these two properties in turn, with complete proofs in Section A.

First, consider a prefix that the student has already generated. Holding $\mathcal{F}_t = (x, \hat{y}_{<t})$ and the model parameters fixed isolates the effect of the next-token draw $Y_t \sim p_t$.

Theorem 1 (Conditional sampling variance and top-k truncation)**.** *Conditioned on* $\mathcal{F}_t$, $\mathrm{Cov}[g_{\mathrm{OPRD}} \mid \mathcal{F}_t] = 0$, *whereas* $\mathrm{Tr}\big(\mathrm{Cov}[g_{\mathrm{sample}} \mid \mathcal{F}_t]\big) = \mathbb{E}\big[\|g_{\mathrm{sample}} - \mathbb{E}[g_{\mathrm{sample}} \mid \mathcal{F}_t]\|_2^2 \mid \mathcal{F}_t\big] \geq 0$. *Top-$k$ OPD*

with $k < |\mathcal{V}|$ is deterministic conditional on $\mathcal{F}_t$, but its gradient is generally biased relative to the full-vocabulary gradient.

Full-vocabulary OPD avoids sampling and truncation but incurs a large vocabulary-level memory cost. OPRD instead supervises pre-LM-head states, reducing this cost while directly transferring internal representations. Yet this raises a deeper question: beyond saving memory, what does this direct supervision offer that matching output probabilities cannot? To answer this, we compare two hidden states h and h' within a single model, using the same fixed LM head $W \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $p(h) = \mathrm{softmax}(Wh)$. Since adding a constant to every logit leaves softmax unchanged, we remove this common shift with $C = I - |\mathcal{V}|^{-1}\mathbf{1}\mathbf{1}^\top$ and write the centered head as $\widetilde{W} = CW$.

Theorem 2 (Output-space invariance and conditioning)**.** *Let $\mathcal{N}_W = \left\{v \in \mathbb{R}^d : \widetilde{W}v = 0\right\}$. The output distribution is invariant to every $v \in \mathcal{N}_W$, i.e., $p(h+v) = p(h)$. For $p = p(h)$ and $p' = p(h')$, if every coordinate of both distributions is at least $\tau \in (0, 1/3)$, then*

$$\left\|\Pi_{\mathcal{N}_{\widetilde{W}}^{\perp}}(h - h')\right\|_2^2 \leq \frac{4\log^2\tau}{\tau\,\sigma_+^2(\widetilde{W})} D_{\mathrm{KL}}(p\|p'), \tag{3}$$

where $\sigma_+(\widetilde{W})$ is the smallest nonzero singular value of $\widetilde{W}$, assuming $\widetilde{W} \neq 0$, and $\Pi_{\mathcal{N}_{\widetilde{W}}^{\perp}}$ is the orthogonal projector onto $\mathcal{N}_{\widetilde{W}}^{\perp}$.

The theorem shows that a small output KL alone does not guarantee close projected hidden states. When either τ or σ_+ is small, the prefactor becomes large, leaving the state-distance bound loose. Since $\tau \leq 1/|\mathcal{V}|$, large vocabularies and rare tokens can readily create this condition. To make this loose control concrete, we apply equal-norm perturbations along the LM head's strongest and weakest singular directions. Their median output KL differs by 13–92× (Figure 4), showing that *output matching can overlook large changes in weak directions.* OPRD avoids this uneven sensitivity by comparing normalized states directly. Further diagnostics are in Section C.5.

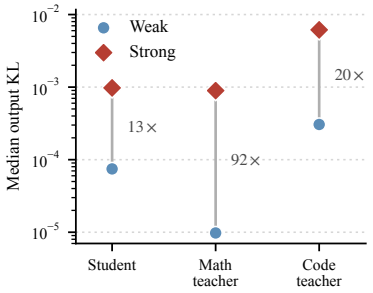


Figure 4: Strong and weak singular directions yield median output KL values differing by 13–92×.

Takeaways

Theorem 1. At a fixed prefix, OPRD uses the hidden vector directly, avoiding sampled-token noise, top-k truncation, and the substantial full-vocabulary-level memory cost.

Theorem 2. Similar output probabilities need not mean similar hidden states. OPRD supervises these states directly instead of relying on the LM head to reveal their differences.

OPRD-Vanilla is simple and effective, making it particularly well suited to multi-teacher RL model merging, where specialists share a common base model. However, its direct alignment relies on comparable hidden states, while we often want to transfer capabilities from a larger teacher to a smaller student, whose hidden states may differ in both dimension and organization. To use the teacher's valuable hidden supervision in these settings, we therefore need a way to make the two representations comparable before aligning them.

2.3 OPRD-Bridge: Heterogeneous Models

Luckily, different architectures do not necessarily prevent models from learning related representations. The Platonic Representation Hypothesis and related evidence suggest that models can capture similar underlying structure despite expressing it in different coordinates (Huh et al., 2024; Kornblith et al., 2019; Moschella et al., 2022; Merullo et al., 2022). If this structure can be expressed in a shared space, then architectural differences need not prevent us from using it as supervision. Motivated by this idea, we introduce **OPRD-Bridge** to first learn a low-rank bridge that makes the states comparable, then distill through this fixed bridge (Figure 2). Thus, the bridge turns the teacher's hidden states into a compact target that preserves shared structure without requiring the student to match a larger representation coordinate by coordinate.

Stage 1: learn the bridge. To learn this bridge, we first need paired states from the two models. Because their depths may differ, we pair student layer l with teacher layer $\phi(l)$ at a similar relative depth, so that early layers are compared with early layers and later layers with later ones (Section B.3). We then freeze both models and collect these states on a fixed calibration set $\mathcal{C}$. Rather than asking the student to match all d_T teacher dimensions, we use principal component analysis (PCA) to retain the r directions with the most variation. With the teacher's mean state denoted by $\mu_T^{(\phi(l))}$ and these directions as $P_T^{(\phi(l))} \in \mathbb{R}^{r \times d_T}$, centering and projecting its states gives a compact target. To make the student's d_S-dimensional states comparable with this target, we then learn a projection $P_S^{(l)} \in \mathbb{R}^{r \times d_S}$ by minimizing

$$\min_{P_S} \; \mathbb{E}_{\mathcal{C}} \frac{1}{r} \left\| P_S h_S - P_T (h_T - \mu_T) \right\|_2^2, \tag{4}$$

Stage 2: distill through the bridge. Once this shared space is learned, we can use the teacher's hidden states to guide the student on its own responses, just as in OPRD-Vanilla. To keep the supervision coordinates fixed while the student learns, we freeze the teacher and both projections. We then let the current student generate fresh responses and map each matched response-position pair $(i, j) \in \mathcal{A}(\hat{y})$ into the shared space:

$$z_{S,l,i} = P_S^{(l)} h_{\theta,i}^{(l)}, \qquad z_{T,l,j} = P_T^{(\phi(l))} \left(h_{T,j}^{(\phi(l))} - \mu_T^{(\phi(l))} \right), \tag{5}$$

Because these projected states now have matching coordinates, we can normalize them as $\bar{z} = z / \max(\|z\|_2, \varepsilon)$ and minimize their mean-squared distance:

$$\mathcal{L}_{\text{Bridge}}(\theta) = \mathbb{E}_{x, \hat{y} \sim \pi_\theta} \left[\frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \frac{1}{|\mathcal{A}(\hat{y})|} \sum_{(i,j) \in \mathcal{A}(\hat{y})} \frac{1}{r} \left\| \bar{z}_{S,l,i} - \text{sg}(\bar{z}_{T,l,j}) \right\|_2^2 \right]. \tag{6}$$

Although P_S is frozen, gradients pass through it to update the student backbone. The student therefore improves its representations while the mapping used to compare them stays unchanged. Moreover, because these projections are needed only for supervision, we discard them after training and deploy the student alone.

During training, however, the bridge's rank determines how much of the teacher's representation is retained for supervision. We therefore compare ranks to see whether retaining more dimensions makes the states easier to align, and select r by held-out alignment after Stage 1. As shown in Figure 5, alignment rises with rank and peaks at 98.4% for $r = 64$, but then falls to 87.0% at full student rank. Thus, a larger shared space does not necessarily provide better-aligned targets. This result supports using a compact subspace to make the teacher's representations comparable with the student's.

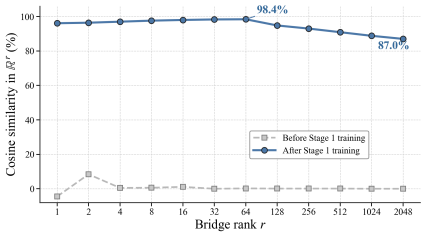


Figure 5: **Low-rank bridge alignment** after Stage 1. Details are in Section B.5.

Together, the two variants move on-policy distillation beyond matching the teacher's next-token predictions by turning the representations behind those predictions into supervision. OPRD-Vanilla transfers this supervision directly, while OPRD-Bridge makes it accessible across architectures through a compact shared space. The central question is therefore whether learning these representations helps the student acquire both the teacher's capabilities and its response behavior with fewer training examples, which we examine next.

3 Experiments

Our single-teacher study shows that direct representation supervision nearly closes the teacher gap (Figure 3 and Table 5). To test whether this benefit extends to more practical settings, we organize our experiments around two settings. Section 3.1 asks whether one student can combine multiple specialists, acquire teacher-like response behavior, learn from less data, and generalize beyond the training domains. Section 3.2 then evaluates transfer from larger, heterogeneous teachers, including open-ended settings whose quality cannot be captured by accuracy alone. Complete training settings, benchmark details, and evaluation protocols are provided in Section B.

Table 1: **Multi-teacher RL model merging.** Student: DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025); teachers: ProRL-1.5B (Liu et al., 2026b) (code) and JustRL-1.5B (He et al., 2025a) (math). The shared teacher ID Avg. combines each specialist's own-domain scores. Bold/underline mark best/second-best distilled results. Dataset and metric details are in Section B.2.

	In-distribution										Out-of-distribution							
	Mathematics					Code					IFEval				Reasoning Gym			
Method	AIMO	A24	M500	Min.	Oly.	HE	MBPP	LC	LCB	**ID Avg.**	S-P	S-I	L-P	L-I	ACRE	Box	GoL	**OOD Avg.**
Code teacher	–	–	–	–	–	75.0	55.1	53.3	11.6	55.1	36.0	49.3	39.7	52.9	55.9	9.0	54.8	42.5
Math teacher	77.1	46.6	77.2	44.1	56.3	–	–	–	–		29.8	43.6	32.2	45.8	53.3	5.1	49.0	37.0
Student	47.0	30.0	61.2	26.5	35.4	56.1	48.6	13.9	0.0	35.4	28.8	42.8	30.7	44.9	43.7	2.6	43.4	33.8
OPD (Sampled)	55.4	33.3	54.8	23.5	43.3	77.4	56.4	**52.8**	0.7	44.2	28.7	42.9	30.1	44.3	57.1	4.6	56.2	37.7
OPD (Top-16)	57.8	43.3	68.6	26.1	45.0	70.1	47.4	43.9	2.0	44.9	28.6	43.0	30.4	44.6	57.0	5.1	**57.7**	38.1
ExOPD	79.5	33.3	75.2	42.6	**56.6**	76.8	55.4	47.8	2.3	52.2	26.7	41.1	28.3	42.5	57.4	2.4	52.3	35.8
AOPD	43.4	33.3	60.4	33.8	45.5	70.1	53.6	51.1	0.2	43.5	28.3	42.6	30.3	44.5	57.8	4.5	52.2	37.1
EOPD	51.8	30.0	60.0	28.7	36.9	57.3	50.4	16.1	0.0	36.8	29.0	43.0	31.1	45.0	47.0	3.8	47.1	35.1
OPD^2	59.0	33.3	66.8	34.6	44.4	17.7	35.1	3.3	0.0	32.7	29.4	43.2	**32.1**	**45.8**	44.9	5.1	46.4	35.3
TIP	51.8	16.7	59.8	26.1	34.4	53.0	47.6	12.2	0.0	33.5	29.4	43.0	31.2	44.9	44.8	2.9	43.9	34.3
PowerOPD	69.9	**50.0**	73.6	40.1	52.1	**78.7**	56.4	48.3	2.0	52.3	28.9	43.1	30.6	44.6	58.3	5.1	54.4	37.8
FiReOPD	73.5	43.3	71.6	34.9	50.5	56.1	52.4	46.1	0.1	47.6	28.1	42.3	29.5	43.6	56.7	3.8	54.3	36.9
TIDE	55.4	46.7	60.0	31.2	45.5	61.0	45.1	47.2	0.2	43.6	27.9	42.2	29.5	43.6	**58.5**	5.3	52.9	37.1
OPRD-Vanilla	**80.7**	46.7	**78.4**	**44.1**	55.0	76.8	**56.6**	51.1	**10.4**	**55.5**	**29.9**	**43.7**	31.9	45.6	58.4	**5.5**	56.3	**38.7**
OPRD-Bridge[1]	74.7	53.3	77.4	44.5	56.0	78.7	53.9	39.4	0.4	53.1	32.2	46.0	34.9	48.4	54.1	5.2	51.4	38.9

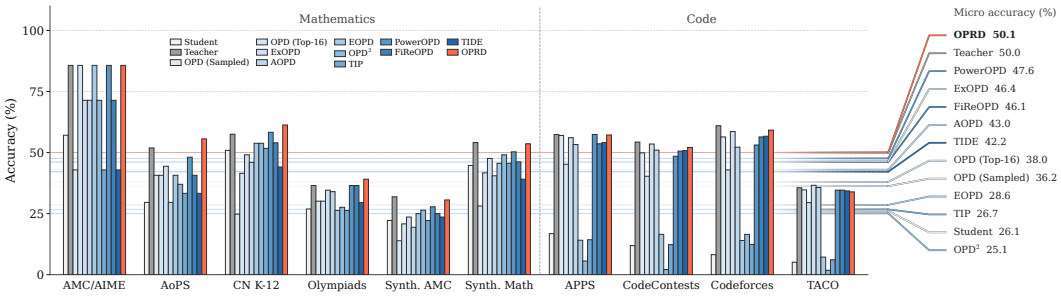


Figure 6: **Multi-teacher RL model merging on PRIME (Cui et al., 2025).** Accuracy across six mathematics and four code sources. Teacher combines the math specialist on math sources and the code specialist on code sources. Full results are in Table 4.

3.1 Multi-Teacher RL Model Merging

Setup. We initialize the student from DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025). Mathematics prompts come from DAPO-Math-17K (Yu et al., 2026) and are routed to JustRL-1.5B (He et al., 2025a); code prompts come from Eurus-2-RL-Data (Cui et al., 2025) and are routed to ProRL-1.5B (Liu et al., 2026b). Thus, one student receives both specialists' guidance. We compare against ten output-space baselines, measuring both task accuracy and response behavior. Full data composition and training settings are in Section B.2.

Can one student combine specialists? Across nine mathematics and code benchmarks, OPRD raises the macro ID score from 35.4 to 55.5, exceeding PowerOPD's 52.3 and matching the routed teachers' 55.1 (Table 1). It also reaches 10.4% Pass@5 on LiveCodeBench v4, close to the code teacher's 11.6% and well above ExOPD's 2.3%. On PRIME, OPRD similarly matches the teachers (50.1% vs. 50.0% micro accuracy) and exceeds PowerOPD (47.6%; Figure 6), showing that one student can consolidate multiple teachers' capabilities.

Does it inherit teacher-like response behavior? Across the same nine benchmarks, OPRD combines the strongest macro score among distilled models with response lengths closest to the routed

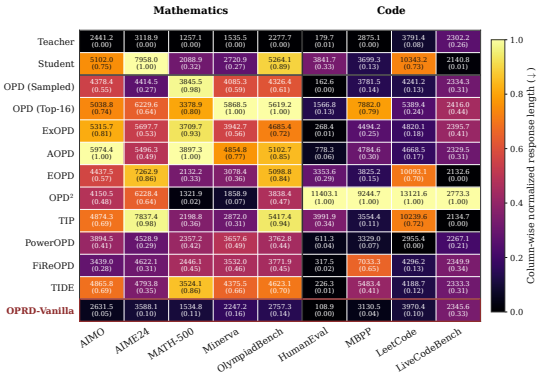
[1]OPRD-Bridge (rank 512) is shown only for reference; it is excluded from best/second-best ranking.

Table 2: **Cross-architecture math distillation.** Skywork-OR1-Math-7B (He et al., 2025b) → DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025); see B.4.

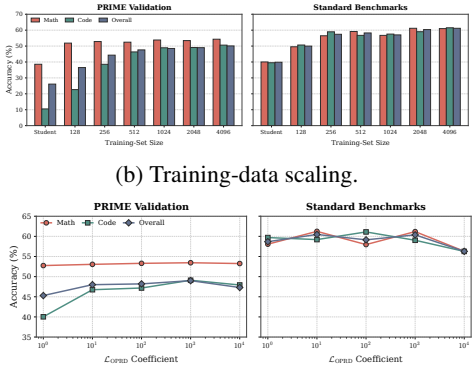
Method	AIMO	A24	M500	Min.	Oly.	Micro
Teacher	80.7	60.0	77.0	45.2	58.4	63.3
Student	47.0	30.0	61.2	26.5	35.4	42.6
OPD (Sampled)	41.0	20.0	59.8	19.5	32.3	39.1
OPD (Top-16)	26.5	13.3	46.2	10.7	21.0	27.4
ExOPD	**54.2**	20.0	**65.2**	20.6	**36.0**	43.3
AOPD	30.1	13.3	51.2	11.0	26.2	31.5
EOPD	34.9	13.3	53.8	15.4	28.6	34.4
PowerOPD	41.0	6.7	59.2	18.4	31.3	38.0
FiReOPD	39.8	**30.0**	61.8	22.8	32.7	40.6
TIDE	36.1	10.0	49.0	13.6	26.7	31.7
OPRD-Bridge	50.6	20.0	62.2	**32.7**	34.1	**43.5**

Table 3: **Cross-architecture open-ended distillation on KardiaBench (Yuan et al., 2026).** Kardia-R1 (Yuan et al., 2026) → DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025); see B.5.

Method	Fail. ↓	Tok. ↓	Rel.	Flu.	Emp.	Per.	Safe.	Avg.
Teacher	18.7	182.0	71.3	71.8	71.1	71.0	72.4	71.5
Student	50.9	1,454.2	38.3	38.3	38.3	38.3	38.5	38.3
OPD (Sampled)	42.8	2,278.3	44.6	44.7	44.6	44.6	44.7	44.6
OPD (Top-16)	43.5	1,729.0	43.7	43.8	43.7	43.7	43.9	43.8
ExOPD	100.0	6,724.2	20.3	20.3	20.3	20.3	20.3	20.3
AOPD	100.0	8,064.4	20.6	20.6	20.6	20.6	20.6	20.6
EOPD	100.0	8,191.8	20.0	20.0	20.0	20.0	20.0	20.0
PowerOPD	100.0	7,196.6	22.4	22.4	22.4	22.3	22.4	22.4
FiReOPD	100.0	8,192.0	20.0	20.0	20.0	20.0	20.0	20.0
TIDE	100.0	8,192.0	20.4	20.4	20.4	20.4	20.4	20.4
OPRD-Bridge	**15.9**	**825.7**	**53.7**	**53.9**	**53.6**	**53.6**	**54.2**	**53.8**



(a) Response lengths.



(b) Training-data scaling.

(c) Coefficient sensitivity.

Figure 7: **Multi-teacher efficiency and robustness.** **(a)** Cells show mean token counts and, in parentheses, per-benchmark min–max normalized lengths. Colors encode normalized lengths; darker is shorter. Teacher combines the math specialist on math benchmarks and the code specialist on code benchmarks. **(b)** Accuracy scaling with domain-balanced training prompts. **(c)** Sensitivity to the coefficient on $\mathcal{L}_{\text{OPRD}}$. Complete results are in Tables 7 to 9.

teachers (Figures 1a and 7a). It thus improves performance without merely generating longer answers, better matching practical preferences for concise responses with lower latency and cost.

How data-efficient is OPRD? With 512 domain-balanced prompts, OPRD matches the strongest baseline trained on 4,096, yielding up to 8× greater data efficiency (Figure 1b). PRIME accuracy rises from 36.5% with 128 prompts to 50.1% with 4,096; mathematics improves earlier, while code continues to benefit from larger budgets (Figure 7b). OPRD remains strong as the representation-loss coefficient varies from 10 to 10^3, showing robustness to this hyperparameter (Figure 7c).

Do the gains extend beyond the training domains? To answer this, we evaluate instruction following on IFEval and procedural reasoning on Reasoning Gym. Among the same-architecture methods, OPRD-Vanilla achieves the highest OOD aggregate. This advantage spans both evaluations: OPRD leads on strict IFEval prompt and instruction accuracy and BoxNet, while remaining competitive on ACRE and Game of Life Halting. Thus, OPRD merging improves out-of-domain capabilities.

3.2 Cross-Architecture Transfer of Open-Ended Abilities

Setup. We now move beyond shared architectures and tasks with a single correct answer. On KardiaBench, an emotional-support benchmark, we distill the 7B Kardia-R1 teacher (Yuan et al., 2026) into DeepSeek-R1-Distill-Qwen-1.5B using a rank-64 bridge (Figure 8) and compare against eight output-space methods. Because abilities such as empathy cannot be captured by accuracy

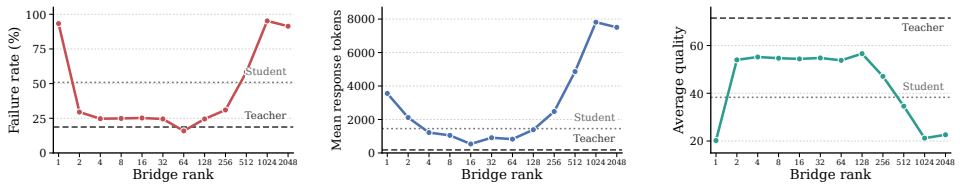


Figure 8: **Bridge-rank ablation on KardiaBench.** Failure rate (↓), mean response length (↓), and response quality (↑) vary non-monotonically with bridge rank; lines mark teacher and initial student.

alone, we evaluate five dimensions of response quality, the rate of complete and normally terminated responses, and mean response length. Full settings and judge prompts are in Section B.5.

Can the bridge transfer across architectures? OPRD-Bridge yields a modest aggregate gain in mathematics, with mixed per-benchmark results, but much larger gains on the open-ended KardiaBench evaluation (Tables 2 and 3). On KardiaBench, it raises mean quality from sampled-token OPD's 44.6 to 53.8, leads all five quality dimensions, and reduces failure from 42.8% to 15.9% and mean length from 2,278 to 826 tokens (Figure 1c). Meanwhile, six other OPD variants produce no responses satisfying both format and termination requirements. Thus, OPRD-Bridge transfers both capability and concise, structured behavior from a larger teacher despite differences in model scale and representation space.

How much bridge capacity is needed? Figure 8 shows a low-rank sweet spot. Ranks 4–128 improve quality, length, and failure over the student; rank 64 gives the lowest failure rate (15.9%) at 826 tokens. Above rank 256, length and failure rise as quality falls, supporting a compact bridge and our rank-64 choice. Complete results are in Table 11.

4 Related Work

Knowledge distillation trains a student to reproduce a teacher's outputs (Hinton et al., 2015; Kim & Rush, 2016), and it has been widely used for language models (Sanh et al., 2019; Jiao et al., 2020; Wang et al., 2020; Chung et al., 2024; Sanh et al., 2021; Wei et al., 2021). MiniLLM and GKD bring this supervision onto student-generated responses (Gu et al., 2024; Agarwal et al., 2024), thereby addressing the gap between training contexts and student predictions. OPD has since become part of large-scale post-training (Xu et al., 2026; Yang et al., 2025; Zeng et al., 2026; Xiao et al., 2026; Ko et al., 2026; Jin et al., 2026; Jang et al., 2026; Fu et al., 2026; Hou et al., 2026). Related self-distillation methods also provide richer guidance through privileged information or different model contexts (Hübotter et al., 2026; Zhao et al., 2026; He et al., 2026; Shenfeld et al., 2026; Ye et al., 2026b; Sang et al., 2026; Kim et al., 2026; Ye et al., 2026a; Yang et al., 2026; Li et al., 2026; Ding, 2026). OPRD builds on this on-policy loop, but changes the target to the teacher's internal states.

Representation distillation already transfers intermediate features in vision and language encoders (Romero et al., 2024; Zagoruyko & Komodakis, 2016; Yim et al., 2017; Jiao et al., 2020; Sun et al., 2020; Wang et al., 2020; 2021). It also connects to supervision at intermediate layers, learned view alignment, and representation engineering (Lee et al., 2015; Caron et al., 2021; Zou et al., 2023). Here, however, the inputs are autoregressive prefixes sampled from the changing student, so the paired representations evolve with its behavior. For heterogeneous models, OPRD-Bridge draws on evidence of cross-model representation similarity and linear translation (Kornblith et al., 2019; Moschella et al., 2022; Merullo et al., 2022). It then fixes the learned mapping so that representation matching trains the student through a consistent comparison.

5 Conclusion

Whenever a teacher evaluates a student's response, it computes hidden states alongside next-token probabilities. OPRD turns these states into supervision through direct alignment for compatible models or a frozen low-rank bridge otherwise. Consequently, students gain task capabilities and teacher-like response behavior from fewer examples. Effective distillation therefore need not stop at matching predictions; it can also learn from the representations that produce them. This perspective opens a path toward transferring the broader capabilities and behavior of strong teachers (Section D).

AI USE STATEMENT

Generative AI assisted with code inspection, figure and table formatting, and improving the clarity of author-written text. Qwen3-8B served as the rubric judge for KardiaBench following its established evaluation protocol. The authors reviewed all AI-assisted work and take full responsibility for the paper.

ETHICS STATEMENT

This work studies model distillation using publicly available models, datasets, and benchmarks. KardiaBench is used only as a research testbed for open-ended emotional-support behavior; our results do not establish clinical safety or suitability for real-world mental-health use. As OPRD can transfer teacher behavior as well as capability, deployed systems should independently assess the teacher's biases, errors, and safety risks.

REPRODUCIBILITY STATEMENT

The objectives and algorithms are given in Section 2, with complete proofs in Section A. Section B documents the models, datasets, training settings, evaluation protocols, and judge prompts. Code, configurations, and evaluation scripts will accompany the anonymized supplement.

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.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Ken Ding. Hdpo: Hybrid distillation policy optimization via privileged self-distillation. *arXiv preprint arXiv:2603.23871*, 2026.

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.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *Advances in Neural Information Processing Systems*, 38: 41340–41391, 2026.

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.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025a.

Bingxiang He, Yuxin Zuo, Zeyuan Liu, Shangziqi Zhao, Zixuan Fu, Junlin Yang, Cheng Qian, Kaiyan Zhang, Yuchen Fan, Ganqu Cui, et al. How far can unsupervised rlvr scale llm training? In *International Conference on Learning Representations*, volume 2026, pp. 14823–14865, 2026.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025b.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

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

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

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.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, 2026.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 4163–4174, 2020.

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.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

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.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pp. 562–570. Pmlr, 2015.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Jiang Liu, Sudhanshu Ranjan, Prakamya Mishra, Yonatan Dukler, Gowtham Ramesh, Jialian Wu, Ximeng Sun, Wen Xie, Chaojun Hou, Vikram Appia, et al. Instella-moe technical report. *arXiv preprint arXiv:2609.00791*, 2026a.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *Advances in Neural Information Processing Systems*, 38:17998–18031, 2026b.

Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space. *arXiv preprint arXiv:2209.15162*, 2022.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. *arXiv preprint arXiv:2209.15430*, 2022.

Beatrix MG Nielsen, Emanuele Marconato, Luigi Gresele, Andrea Dittadi, and Simon Buchholz. Logit distance bounds representational similarity. *arXiv preprint arXiv:2602.15438*, 2026.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: hints for thin deep nets (2014). *arXiv preprint arXiv:1412.6550*, 3, 2024.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

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

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. *Advances in Neural Information Processing Systems*, 38, 2026.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2158–2170, 2020.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2140–2151, 2021.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Yunhui Xia, Wei Shen, Yan Wang, Jason Klein Liu, Huifeng Sun, Siyue Wu, Jian Hu, and Xiaolong Xu. Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms. *arXiv preprint arXiv:2504.14655*, 2025.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

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

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

Tianzhu Ye, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Online experiential learning for language models. *arXiv preprint arXiv:2603.16856*, 2026a.

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

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 7130–7138. IEEE, 2017.

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.

Jiahao Yuan, Zhiqing Cui, Hanqing Wang, Yuansheng Gao, Yucheng Zhou, and Usman Naseem. Kardia-r1: Unleashing llms to reason toward understanding and empathy for emotional support via rubric-as-judge reinforcement learning. In *Proceedings of the ACM Web Conference 2026*, pp. 9230–9240, 2026.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

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

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Formal Theoretical Analysis

The two results in §2.2 (Theorems 1 and 2) summarize the relevant variance and output-space conditioning properties. We now state their assumptions and proofs:

- §§A.2–A.4 give the exact sampled-token variance, top-k truncation-bias, and OPRD conditional-variance identities (Theorems 3 to 5 and Corollary 1).
- §A.5 gives exact softmax invariance, global KL-to-hidden conditioning, and local Fisher-curvature results (Theorems 6 to 8).

Throughout this section, we analyze one response position t. Let

$$\mathcal{F}_t \triangleq \sigma(x, \hat{y}_{<t}) \tag{7}$$

denote the information contained in the prompt and sampled prefix. Conditional on $\mathcal{F}_t$, the prefix is fixed and the only random variable considered in §A.2 is the next-token draw $Y_t \sim p_t$. The multi-position objective is a sum of such terms; correlations between positions belong to rollout randomness and are not removed by the per-position statement below.

We use θ for the student parameters and π_θ, π_T for the student and teacher policies. For brevity, conditional on $\mathcal{F}_t$, write

$$p(v) \equiv \pi_\theta(v \mid x, \hat{y}_{<t}), \qquad q(v) \equiv \pi_T(v \mid x, \hat{y}_{<t}), \qquad u(v) \equiv \log p(v) - \log q(v). \tag{8}$$

A.1 Detached-Rollout Setup and Assumptions

Definition 1 (Conditional semi-gradients)**.** Fix $\mathcal{F}_t$ and draw $Y_t \sim p$. Rollouts are refreshed on policy, but their token IDs are detached during each optimizer update. We analyze the conditional semi-gradients

$$g_{\mathrm{OPD}}(\theta; Y_t) \triangleq u(Y_t) s_\theta(Y_t), \tag{9}$$

$$g_{\mathrm{OPRD}}(\theta) \triangleq \nabla_\theta \tfrac{1}{d} \left\| \bar{h}_{\theta,t}^{(L)} - \mathrm{sg}(\bar{h}_{T,t}^{(L)}) \right\|_2^2, \tag{10}$$

where $s_\theta(v) = \nabla_\theta \log p(v)$ and $\bar{h} = h / \max(\|h\|_2, \varepsilon)$ is the normalization used by OPRD. The corresponding conditional means are $\bar{g}_{\mathrm{OPD}}(\theta) \triangleq \mathbb{E}[g_{\mathrm{OPD}} \mid \mathcal{F}_t]$ and $\bar{g}_{\mathrm{OPRD}}(\theta) \triangleq g_{\mathrm{OPRD}}$ (which is already deterministic in Y_t). Because these gradients belong to different objectives, the comparison below isolates noise sources rather than comparing two unbiased estimators of one common gradient. We index h_t as the causal state used to predict Y_t, so it depends on x and $\hat{y}_{<t}$ but not on the next-token draw itself.

Assumption 1 (Regularity)**.** The following conditions hold throughout: **(R1)** The log-densities $\log p_\theta(v)$ are twice continuously differentiable in θ for every $v \in \mathcal{V}$, and $q(v) > 0$ whenever $p(v) > 0$. **(R2)** The score-weighted log ratio has finite second moment:

$$\mathbb{E}_{Y_t \sim p}\left[u(Y_t)^2 \|s_\theta(Y_t)\|_2^2\right] < \infty. \tag{11}$$

For finite-vocabulary softmax models with finite logits, positivity and this moment condition hold automatically at any fixed parameter value. **(R3)** The normalized hidden state $\bar{h}_{\theta,t}^{(L)}$ is continuously differentiable in θ and its Jacobian has finite operator norm along the optimization trajectory.

A.2 Variance of Sampled-Token OPD

Lemma 1 (Conditional mean and unbiasedness of sampled-token OPD)**.** *Under Assumption 1, the conditional mean of the detached-rollout sampled-token semi-gradient equals the gradient of the conditional reverse KL:*

$$\bar{g}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{Y_t \sim p}\big[u(Y_t) s_\theta(Y_t)\big] = \nabla_\theta D_{\mathrm{KL}}(p \| q). \tag{12}$$

Proof. The first equality follows from the definition of conditional expectation:

$$\mathbb{E}[g_{\mathrm{OPD}}(\theta; Y_t) \mid \mathcal{F}_t] = \sum_{v \in \mathcal{V}} p(v) u(v) s_\theta(v). \tag{13}$$

To identify this mean with the reverse-KL gradient, keep the sampled prefix fixed and differentiate

$$D_{\mathrm{KL}}(p\|q) = \sum_{v\in\mathcal{V}} p(v)\left[\log p(v) - \log q(v)\right] = \sum_{v} p(v)u(v). \tag{14}$$

The teacher is frozen, so $\nabla_\theta \log q(v) = 0$. Using $\nabla_\theta p(v) = p(v)s_\theta(v)$ and $\nabla_\theta u(v) = s_\theta(v)$ gives

$$\nabla_\theta D_{\mathrm{KL}}(p\|q) = \sum_{v} p(v)u(v)s_\theta(v) + \sum_{v} p(v)s_\theta(v) \tag{15}$$

$$= \sum_{v} p(v)u(v)s_\theta(v) + \sum_{v} \nabla_\theta p(v). \tag{16}$$

The second sum vanishes because

$$\sum_{v} \nabla_\theta p(v) = \nabla_\theta \sum_{v} p(v) = \nabla_\theta 1 = 0. \tag{17}$$

Therefore the remaining term is exactly (13), proving (12). The unbiasedness is conditional on $\mathcal{F}_t$: differentiating through the random process that generated the prefix is deliberately excluded by the detached-rollout convention. □

Theorem 3 (Exact sampled-token OPD variance)**.** *Under Assumption 1, the trace of the conditional covariance of the single-sample OPD semi-gradient is*

$$\mathrm{Tr}(\mathrm{Cov}[g_{\mathrm{OPD}}(\theta; Y_t) \mid \mathcal{F}_t]) \;=\; \mathbb{E}_p\Big[u(Y_t)^2\|s_\theta(Y_t)\|_2^2\Big] \;-\; \big\|\bar{g}_{\mathrm{OPD}}(\theta)\big\|_2^2. \tag{18}$$

Proof. Let

$$G \triangleq g_{\mathrm{OPD}}(\theta; Y_t) = u(Y_t)s_\theta(Y_t), \qquad m \triangleq \mathbb{E}[G \mid \mathcal{F}_t] = \bar{g}_{\mathrm{OPD}}(\theta). \tag{19}$$

By definition, the conditional covariance matrix is

$$\mathrm{Cov}[G \mid \mathcal{F}_t] \triangleq \mathbb{E}[(G-m)(G-m)^\top \mid \mathcal{F}_t] \tag{20}$$

$$= \mathbb{E}[GG^\top \mid \mathcal{F}_t] - mm^\top. \tag{21}$$

Taking the trace and using $\mathrm{Tr}(aa^\top) = \|a\|_2^2$ for any vector a yields

$$\mathrm{Tr}(\mathrm{Cov}[G \mid \mathcal{F}_t]) = \mathbb{E}[\mathrm{Tr}(GG^\top) \mid \mathcal{F}_t] - \mathrm{Tr}(mm^\top) \tag{22}$$

$$= \mathbb{E}[\|G\|_2^2 \mid \mathcal{F}_t] - \|m\|_2^2 \tag{23}$$

$$= \mathbb{E}_p[u(Y_t)^2\|s_\theta(Y_t)\|_2^2] - \|\bar{g}_{\mathrm{OPD}}(\theta)\|_2^2, \tag{24}$$

which is (18). Equivalently,

$$\mathrm{Tr}(\mathrm{Cov}[G \mid \mathcal{F}_t]) = \mathbb{E}[\|G-m\|_2^2 \mid \mathcal{F}_t] \geq 0. \tag{25}$$

The moment assumption in (11) guarantees that all quantities above are finite. Equality holds if and only if $G = m$ for p-almost every token. This is an exact finite-sample identity; it does not by itself imply any asymptotic training-dynamics claim. □

A.3 Truncation Bias of Top-k OPD

We next analyze the normalized student-support top-k semi-gradient used in our comparison. Conditioned on $\mathcal{F}_t$, define $a(v) = u(v)s_\theta(v)$ and let $S_k = \mathrm{TopK}(p, k)$. Write

$$\alpha_k \triangleq \sum_{v\in S_k} p(v), \qquad p^S(v) \triangleq \frac{p(v)\mathbf{1}\{v \in S_k\}}{\alpha_k}, \qquad p^{\bar{S}}(v) \triangleq \frac{p(v)\mathbf{1}\{v \notin S_k\}}{1-\alpha_k}, \tag{26}$$

where $k < |\mathcal{V}|$, so $1-\alpha_k > 0$ for a finite-logit softmax. The full-vocabulary and normalized-support top-k semi-gradients are

$$g_{\mathrm{full}} \triangleq \mathbb{E}_{V\sim p}[a(V)], \qquad g_{\text{top-}k} \triangleq \mathbb{E}_{V\sim p^S}[a(V)]. \tag{27}$$

The top-k support and its normalized weights are deterministic once the prefix and current model are fixed and are detached within the semi-gradient update, matching the detached-reward convention used by the implementation.

Theorem 4 (Exact top-k truncation bias)**.** *Conditioned on $\mathcal{F}_t$, the normalized-support top-k semi-gradient is deterministic and therefore $\mathrm{Cov}[g_{\text{top-}k} \mid \mathcal{F}_t] = 0$, but it differs from the full-vocabulary reverse-KL semi-gradient by*

$$g_{\text{top-}k} - g_{\text{full}} = (1 - \alpha_k)\left(\mathbb{E}_{p^S}[a(V)] - \mathbb{E}_{p^{\bar{S}}}[a(V)]\right). \tag{28}$$

Consequently, the bias vanishes if and only if either the selected support has full probability mass or the support and tail have identical mean token-conditioned gradients.

Proof. Because S_k, p^S, and $a(v)$ are fixed by $\mathcal{F}_t$ and the current parameters, $g_{\text{top-}k}$ contains no subsequent categorical draw, proving the zero conditional covariance statement. For the bias, the student distribution decomposes exactly into its selected and omitted supports:

$$p = \alpha_k p^S + (1 - \alpha_k) p^{\bar{S}}. \tag{29}$$

Taking the expectation of $a(V)$ under this mixture gives

$$g_{\text{full}} = \alpha_k \mathbb{E}_{p^S}[a(V)] + (1 - \alpha_k)\mathbb{E}_{p^{\bar{S}}}[a(V)]. \tag{30}$$

Subtracting this expression from $g_{\text{top-}k} = \mathbb{E}_{p^S}[a(V)]$ yields

$$g_{\text{top-}k} - g_{\text{full}} = (1 - \alpha_k)\mathbb{E}_{p^S}[a(V)] - (1 - \alpha_k)\mathbb{E}_{p^{\bar{S}}}[a(V)] \tag{31}$$

$$= (1 - \alpha_k)\left(\mathbb{E}_{p^S}[a(V)] - \mathbb{E}_{p^{\bar{S}}}[a(V)]\right), \tag{32}$$

which proves (28). The equality condition follows because a scalar multiple of a vector is zero exactly when the scalar is zero or the vector is zero. This result concerns bias relative to the full-vocabulary reverse-KL semi-gradient. OPRD does not estimate that gradient; it avoids vocabulary truncation by optimizing a separate dense representation objective. □

A.4 Conditional Determinism of the OPRD Gradient

Theorem 5 (OPRD has zero conditional variance)**.** *Under Assumption 1, the OPRD per-position gradient satisfies*

$$\mathrm{Cov}\left[g_{\text{OPRD}}(\theta) \mid \mathcal{F}_t\right] = 0, \tag{33}$$

and is given in closed form by

$$g_{\text{OPRD}}(\theta) \;=\; \frac{2}{d} J_{\bar{h},\theta}^{\top}\left(\bar{h}_{\theta,t}^{(L)} - \bar{h}_{T,t}^{(L)}\right). \tag{34}$$

Proof. The causal state $h_{\theta,t}^{(L)}$ is computed from $(x, \hat{y}_{<t})$ and is used to predict Y_t. It therefore does not depend on the subsequent draw Y_t once $\mathcal{F}_t$ is fixed. The teacher state is computed from the same fixed prefix, and stop-gradient makes it constant with respect to θ. Define

$$a(\theta) \triangleq \bar{h}_{\theta,t}^{(L)}, \qquad b \triangleq \mathrm{sg}(\bar{h}_{T,t}^{(L)}), \qquad r(\theta) \triangleq a(\theta) - b. \tag{35}$$

Conditional on $\mathcal{F}_t$, the per-position loss is the deterministic scalar

$$\ell_{\text{OPRD}}(\theta) = \frac{1}{d} r(\theta)^{\top} r(\theta). \tag{36}$$

It contains no occurrence of Y_t. Hence there is a deterministic vector $g_\star(\mathcal{F}_t, \theta)$ such that $g_{\text{OPRD}} = g_\star$ for every possible value of Y_t. Its conditional mean is also $g_\star$, and therefore

$$\mathrm{Cov}[g_{\text{OPRD}} \mid \mathcal{F}_t] = \mathbb{E}[(g_\star - g_\star)(g_\star - g_\star)^{\top} \mid \mathcal{F}_t] \tag{37}$$

$$= 0, \tag{38}$$

which proves (33).

For the closed form, let $J_{\bar{h},\theta} \triangleq \nabla_\theta a(\theta)$, using the convention that the Jacobian maps a parameter perturbation to a hidden-state perturbation. Differentiating (36) gives

$$\nabla_\theta \ell_{\text{OPRD}}(\theta) = \frac{1}{d}\nabla_\theta[r(\theta)^{\top} r(\theta)] \tag{39}$$

$$= \frac{2}{d} J_{\bar{h},\theta}^{\top} r(\theta) \tag{40}$$

$$= \frac{2}{d} J_{\bar{h},\theta}^{\top}\left(\bar{h}_{\theta,t}^{(L)} - \bar{h}_{T,t}^{(L)}\right), \tag{41}$$

proving (34).

For completeness, away from the boundary $\|h\|_2 = \varepsilon$, the Jacobian of the normalization map $N(h) = h/\max(\|h\|_2, \varepsilon)$ is

$$\nabla_h N(h) = \begin{cases} \varepsilon^{-1} I, & \|h\|_2 < \varepsilon, \\ \|h\|_2^{-1}(I - \bar{h}\bar{h}^\top), & \|h\|_2 > \varepsilon. \end{cases} \tag{42}$$

Thus $J_{\bar{h},\theta}$ already includes the effect of normalization. A nonzero representation residual does not necessarily imply a nonzero parameter gradient: (41) vanishes whenever the residual lies in $\ker(J_{\bar{h},\theta}^\top)$. □

Corollary 1 (Removal of sampled-token conditional noise)**.** *For a fixed prompt and prefix, sampled-token OPD contains the additional conditional covariance in* (18)*, whereas OPRD contains no randomness from drawing Y_t. In particular,*

$$\mathrm{Tr}(\mathrm{Cov}[g_{\mathrm{OPD}} \mid \mathcal{F}_t]) \;=\; \mathbb{E}_p\Big[u(Y_t)^2 \|s_\theta(Y_t)\|_2^2\Big] - \|\bar{g}_{\mathrm{OPD}}\|_2^2 \;\geq\; 0, \tag{43}$$

with equality exactly when $u(Y_t)s_\theta(Y_t)$ is p-almost surely constant. Because OPD and OPRD optimize different losses, this is a comparison of noise sources, not a claim that OPRD is an unbiased lower-variance estimator of the OPD gradient.

Proof. Equation (43) is (25) with $G = u(Y_t)s_\theta(Y_t)$. A nonnegative random variable has zero expectation if and only if it is zero almost surely, so equality holds exactly when $\|G - \bar{g}_{\mathrm{OPD}}\|_2^2 = 0$ for p-almost every Y_t. By (38), the corresponding conditional covariance contribution for OPRD is identically zero. □

Remark 1 (What the zero-variance statement removes). For any gradient estimator G with finite second moment, the law of total covariance gives

$$\mathrm{Cov}(G) = \mathbb{E}[\mathrm{Cov}(G \mid \mathcal{F}_t)] + \mathrm{Cov}(\mathbb{E}[G \mid \mathcal{F}_t])\,. \tag{44}$$

The first term measures variation from drawing the next token after the prompt and prefix are fixed; the second measures variation across prompts and sampled prefixes. Theorem 5 sets only the former token-draw term to zero for OPRD. It does not remove rollout, prompt, position, minibatch, model, or optimizer randomness. For sampled-token OPD, the first term is exactly the conditional covariance characterized by Theorem 3.

A.5 Output-Space Invariance and Conditioning

We prove Theorem 2 by first characterizing exact output invariance and then deriving the global KL-to-state bound. The final part gives the additional local Fisher-curvature analysis. We analyze what can be recovered about a student's last-layer state from its own output distribution. Let $W \in \mathbb{R}^{|\mathcal{V}| \times d}$ be the student LM head, $p(h) = \sigma(Wh)$, $C = I - |\mathcal{V}|^{-1}\mathbf{1}\mathbf{1}^\top$ the logit-centering projector, and $\widetilde{W} = CW$. This local observability analysis does not require the teacher and student to share an LM head.

A.5.1 Centered logits and exact invariance

Definition 2 (Effective null space of the LM head)**.** Define

$$\mathcal{N}_W \;\triangleq\; \ker(\widetilde{W}) = \big\{v \in \mathbb{R}^d : Wv \in \mathrm{span}\{\mathbf{1}\}\big\}. \tag{45}$$

Lemma 2 (Softmax kernel)**.** *For any $z \in \mathbb{R}^{|\mathcal{V}|}$ and any $c \in \mathbb{R}$, $\sigma(z + c\mathbf{1}) = \sigma(z)$. Conversely, $\sigma(z) = \sigma(z')$ implies $z' - z \in \mathrm{span}\{\mathbf{1}\}$.*

Proof. For the forward direction, the i-th coordinate is

$$\sigma(z + c\mathbf{1})_i = \frac{e^{z_i + c}}{\sum_j e^{z_j + c}} = \frac{e^c e^{z_i}}{e^c \sum_j e^{z_j}} = \sigma(z)_i. \tag{46}$$

For the converse, equality of the two softmax distributions gives, for every pair i, j,

$$e^{z_i - z_j} = \frac{\sigma(z)_i}{\sigma(z)_j} = \frac{\sigma(z')_i}{\sigma(z')_j} = e^{z'_i - z'_j}. \tag{47}$$

Taking logarithms shows that $z'_i - z_i = z'_j - z_j$ for all i, j; hence $z' - z = c\mathbf{1}$ for some scalar c. □

Lemma 3 (Centered-logit representation). *For $p = \sigma(z)$,*

$$C \log p = Cz. \tag{48}$$

Consequently, if $z = Wh$, $z' = Wh'$, $p = p(h)$, and $p' = p(h')$, then

$$\widetilde{W}(h - h') = C(z - z') = C(\log p - \log p'). \tag{49}$$

Proof. Writing $A(z) = \log \sum_j e^{z_j}$, the softmax definition gives $\log p = z - A(z)\mathbf{1}$. Since $C\mathbf{1} = 0$, applying C yields $C \log p = Cz$. Subtracting this identity for z and z' and using $\widetilde{W} = CW$ proves (49). □

Theorem 6 (Exact output-space invariance). *For any hidden state h and perturbation v,*

$$p(h + v) = p(h) \quad \Longleftrightarrow \quad v \in \mathcal{N}_W. \tag{50}$$

Consequently, any loss that depends on h only through $p(h)$ is invariant along $\mathcal{N}_W$. The exact subspace $\mathcal{N}_W$ may be trivial.

Proof. If $v \in \mathcal{N}_W = \ker(CW)$, then $CWv = 0$. Because $\ker(C) = \operatorname{span}\{\mathbf{1}\}$, there is a scalar c such that $Wv = c\mathbf{1}$. Therefore $W(h + v) = Wh + c\mathbf{1}$, and the forward part of Lemma 2 gives $p(h + v) = p(h)$.

Conversely, if $p(h + v) = p(h)$, the converse part of Lemma 2, applied to $Wh + Wv$ and Wh, implies $Wv \in \operatorname{span}\{\mathbf{1}\}$. Equivalently, $CWv = 0$, so $v \in \mathcal{N}_W$. Finally, if a loss has the form $\ell(h) = \varphi(p(h))$ for any fixed function φ, then $\ell(h + v) = \varphi(p(h + v)) = \varphi(p(h)) = \ell(h)$. □

A.5.2 Global KL-to-hidden conditioning

We first record the scalar inequality that makes the probability-floor dependence explicit.

Lemma 4 (Log-ratio control under a probability floor). *For $0 < \tau < 1/3$ and $x \in [\tau, \tau^{-1}]$,*

$$\log^2 x \le 4 \log^2 \tau \, (x \log x - x + 1). \tag{51}$$

If probability vectors p, p' satisfy $p_i, p'_i \ge \tau$ for every i, then

$$\sum_i \log^2 \frac{p_i}{p'_i} \le \frac{4 \log^2 \tau}{\tau} D_{\mathrm{KL}}(p \| p'). \tag{52}$$

Proof. For $x \neq 1$, define

$$\phi(x) = \frac{\log^2 x}{x \log x - x + 1}, \qquad \phi(1) = 2, \tag{53}$$

where the value at one is the continuous limit. Differentiating gives

$$\phi'(x) = \frac{2 \log x}{(x \log x - x + 1)^2} \left(\log x - 1 + \frac{1}{x} - \frac{1}{2} \log^2 x \right). \tag{54}$$

The parenthesized factor is nonnegative for $x \le 1$ and nonpositive for $x \ge 1$: its derivative is $-x^{-2}(x \log x - x + 1) \le 0$ and its value at one is zero. It therefore has the opposite sign to $\log x$, implying $\phi'(x) \le 0$. Hence, for $x \ge \tau$, $\phi(x) \le \phi(\tau)$. Moreover, $\tau \log \tau - \tau + 1 \ge 1/4$ for $0 < \tau < 1/3$, so $\phi(\tau) \le 4 \log^2 \tau$. This proves (51).

Now set $r_i = p_i/p'_i$. The probability floor gives $r_i \in [\tau, \tau^{-1}]$. Since $r \log r - r + 1 \geq 0$ and $p'_i \geq \tau$, summing (51) yields

$$\sum_i \log^2 r_i \leq 4 \log^2 \tau \sum_i (r_i \log r_i - r_i + 1) \tag{55}$$

$$\leq \frac{4 \log^2 \tau}{\tau} \sum_i p'_i (r_i \log r_i - r_i + 1) \tag{56}$$

$$= \frac{4 \log^2 \tau}{\tau} \left(\sum_i p_i \log \frac{p_i}{p'_i} - \sum_i p_i + \sum_i p'_i \right) \tag{57}$$

$$= \frac{4 \log^2 \tau}{\tau} D_{\mathrm{KL}}(p \| p'), \tag{58}$$

because both probability vectors sum to one. This proves (52); it is the pointwise inequality underlying Nielsen et al. (2026, Lemma E.2). □

Theorem 7 (Global KL-to-hidden conditioning)**.** *Let $p = p(h)$ and $p' = p(h')$. Assuming $\widetilde{W}$ has positive rank, let $\sigma_+(\widetilde{W})$ be the smallest nonzero singular value of $\widetilde{W}$. For arbitrary finite logits,*

$$2 D_{\mathrm{KL}}(p \| p') \leq \| \widetilde{W}(h - h') \|_2^2. \tag{59}$$

If every coordinate of both p and p' is at least $\tau \in (0, 1/3)$, then

$$\| \widetilde{W}(h - h') \|_2^2 \leq \frac{4 \log^2 \tau}{\tau} D_{\mathrm{KL}}(p \| p'), \tag{60}$$

$$\| \Pi_{\mathcal{N}_W^\perp}(h - h') \|_2^2 \leq \frac{4 \log^2 \tau}{\tau \, \sigma_+^2(\widetilde{W})} D_{\mathrm{KL}}(p \| p'). \tag{61}$$

Proof. Write $z = Wh$, $z' = Wh'$, and $A(z) = \log \sum_i e^{z_i}$. Since $\nabla A(z) = p$,

$$D_{\mathrm{KL}}(p \| p') = \sum_i p_i \big[(z_i - A(z)) - (z'_i - A(z')) \big] \tag{62}$$

$$= A(z') - A(z) - \nabla A(z)^\top (z' - z). \tag{63}$$

Thus KL is the Bregman divergence of log-sum-exp, with the arguments ordered as (z', z). Its Hessian is

$$\nabla^2 A(z) = F(p) = \mathrm{diag}(p) - p p^\top. \tag{64}$$

For any vector a,

$$a^\top F(p) a = \sum_i p_i a_i^2 - \left(\sum_i p_i a_i \right)^2 = \mathrm{Var}_{i \sim p}(a_i) \leq \|a\|_2^2. \tag{65}$$

Hence A is one-smooth. Applying the standard smoothness inequality to the Bregman form in (63) gives

$$D_{\mathrm{KL}}(p \| p') \leq \tfrac{1}{2} \| \bar{z}' - \bar{z} \|_2^2, \qquad \bar{z} = Cz, \quad \bar{z}' = Cz'. \tag{66}$$

We may use centered logits because they produce the same softmax distributions by Lemma 2. By Lemma 3, $\bar{z}' - \bar{z} = \widetilde{W}(h' - h)$, proving (59).

For the reverse direction, Lemma 3 and the fact that C is an orthogonal projector give

$$\| \widetilde{W}(h - h') \|_2^2 = \| C(\log p - \log p') \|_2^2 \tag{67}$$

$$\leq \| \log p - \log p' \|_2^2 = \sum_i \log^2 \frac{p_i}{p'_i}. \tag{68}$$

Applying Lemma 4 proves (60).

It remains to convert centered-logit control into hidden-state control. Let $\Delta h = h - h'$ and decompose it orthogonally as

$$\Delta h = \Delta h_0 + \Delta h_\perp, \qquad \Delta h_0 \in \mathcal{N}_W, \quad \Delta h_\perp = \Pi_{\mathcal{N}_W^\perp} \Delta h. \tag{69}$$

Because $\widetilde{W}\Delta h_0 = 0$, the singular-value characterization on $\mathcal{N}_{\widetilde{W}}^{\perp}$ yields

$$\|\widetilde{W}\Delta h\|_2 = \|\widetilde{W}\Delta h_\perp\|_2 \geq \sigma_+(\widetilde{W})\|\Delta h_\perp\|_2. \tag{70}$$

Squaring this inequality and combining it with (60) proves (61). □

The lower-probability assumption is decisive. Across the general softmax model family, no KL-only upper bound on centered-logit distance exists without such regularity: changes on near-zero-probability tokens can produce large logit and representation changes at negligible KL cost (Nielsen et al., 2026). Since $\tau \leq |\mathcal{V}|^{-1}$ and practical minimum token probabilities are much smaller, the factor $\tau^{-1}\log^2\tau$ can make the representation guarantee vacuous. Poor conditioning of $\widetilde{W}$ compounds this effect.

Corollary 2 (Cross-model linear-similarity control)**.** *Consider teacher and student models in the bilinear softmax family*

$$p_M(y \mid x) = \frac{\exp(f_M(x)^\top g_M(y))}{\sum_{y'}\exp(f_M(x)^\top g_M(y'))}, \qquad M \in \{T, S\}, \tag{71}$$

with the same representation dimension m. Let $u_T(x)$ and $u_S(x)$ be their centered logit vectors and define

$$d^2_{\text{logit}}(p_T, p_S) \triangleq \mathbb{E}_x\|u_T(x) - u_S(x)\|_2^2, \tag{72}$$

$$d_{\text{KL}}(p_T, p_S) \triangleq \mathbb{E}_x D_{\text{KL}}(p_T(\cdot \mid x)\|p_S(\cdot \mid x)). \tag{73}$$

Assume both output distributions are pointwise τ-lower-bounded and the diversity and full-rank conditions of Nielsen et al. (2026, Theorem 3.4) hold. If $\mu_m > 0$ is the smallest nonzero eigenvalue of the teacher's centered-logit covariance, then

$$\text{mCCA}(f_T, f_S) \geq 1 - \frac{4\log^2\tau}{\tau\, m\mu_m} d_{\text{KL}}(p_T, p_S). \tag{74}$$

Proof. The logit-to-representation result of Nielsen et al. (2026, Theorem 3.4) states

$$\text{mCCA}(f_T, f_S) \geq 1 - \frac{d^2_{\text{logit}}(p_T, p_S)}{m\mu_m}. \tag{75}$$

Applying (60) at each x and taking expectations gives

$$d^2_{\text{logit}}(p_T, p_S) \leq \frac{4\log^2\tau}{\tau} d_{\text{KL}}(p_T, p_S). \tag{76}$$

Substituting (76) into (75) proves (74). □

Even this linear-equivalence guarantee inherits the severe probability-floor factor and becomes vacuous when the teacher does not sufficiently excite every representation direction, i.e., when μ_m is small. Moreover, mCCA measures agreement up to linear transformations; it does not imply equality of raw teacher and student coordinates. OPRD-Vanilla requires the stronger coordinate alignment supplied by common architecture and initialization, while OPRD-Bridge learns an explicit cross-model interface.

A.5.3 Local Fisher curvature

Theorem 8 (Local Fisher curvature)**.** *Let $F(p) = \text{diag}(p) - pp^\top$ and $H(h) = W^\top F(p(h))W$. For $\Delta h \to 0$,*

$$D_{\text{KL}}(p(h + \Delta h)\|p(h)) = \tfrac{1}{2}\Delta h^\top H(h)\Delta h + o(\|\Delta h\|_2^2). \tag{77}$$

If the positive eigenvalues of $H(h)$ include $\gamma_{\max} \geq \gamma_{\min} > 0$, then perturbations in their unit eigendirections with equal sufficiently small KL divergence satisfy

$$\frac{\|\Delta h_{\min}\|_2^2}{\|\Delta h_{\max}\|_2^2} \longrightarrow \frac{\gamma_{\max}}{\gamma_{\min}}. \tag{78}$$

A zero-eigenvalue direction belongs to $\mathcal{N}_W$ and is therefore exactly output-invariant.

Proof. Let $z = Wh$, $\delta = W\Delta h$, and $A(z) = \log \sum_i e^{z_i}$. Denote $p = \sigma(z)$ and $p_\delta = \sigma(z + \delta)$. Directly from the softmax log probabilities,

$$D_{\mathrm{KL}}(p_\delta \| p) = \sum_i (p_\delta)_i \left[\delta_i - A(z + \delta) + A(z)\right] \tag{79}$$

$$= \nabla A(z + \delta)^\top \delta - A(z + \delta) + A(z). \tag{80}$$

Because $\nabla A(z) = p$ and $\nabla^2 A(z) = F(p)$, Taylor expansion at z gives

$$A(z + \delta) = A(z) + p^\top \delta + \tfrac{1}{2}\delta^\top F(p)\delta + o(\|\delta\|_2^2), \tag{81}$$

$$\nabla A(z + \delta) = p + F(p)\delta + o(\|\delta\|_2). \tag{82}$$

Substituting both expansions into (80) cancels the constant and first-order terms:

$$D_{\mathrm{KL}}(p_\delta \| p) = \tfrac{1}{2}\delta^\top F(p)\delta + o(\|\delta\|_2^2) \tag{83}$$

$$= \tfrac{1}{2}\Delta h^\top W^\top F(p) W \Delta h + o(\|\Delta h\|_2^2), \tag{84}$$

where the final remainder follows because W is fixed and linear. This is (77).

Now let e_γ be a unit eigenvector of $H(h)$ with eigenvalue $\gamma > 0$ and set $\Delta h = \alpha e_\gamma$. Then

$$D_{\mathrm{KL}}(p(h + \alpha e_\gamma) \| p(h)) = \tfrac{1}{2}\gamma\alpha^2 + o(\alpha^2). \tag{85}$$

For a common target divergence $\varepsilon \downarrow 0$, the required squared step is $\alpha_\gamma^2 = 2\varepsilon/\gamma + o(\varepsilon)$. Therefore

$$\frac{\alpha_{\gamma_{\min}}^2}{\alpha_{\gamma_{\max}}^2} \longrightarrow \frac{\gamma_{\max}}{\gamma_{\min}}, \tag{86}$$

which proves (78).

Finally, if $H(h)v = 0$, then $0 = v^\top H(h)v = (Wv)^\top F(p)(Wv)$. Finite logits imply $p_i > 0$ for every token, so $\ker F(p) = \mathrm{span}\{\mathbf{1}\}$. Hence $Wv \in \mathrm{span}\{\mathbf{1}\}$, or equivalently $v \in \mathcal{N}_W$. Theorem 6 then gives exact output invariance along v. □

This local quadratic statement applies to full-distribution KL geometry. A single sampled-token log-ratio or a truncated top-k estimator can be signed and is not itself a nonnegative quadratic form; its relation to this geometry holds only after the appropriate expectation or approximation.

Remark 2 (Intermediate layers). Output losses do influence intermediate states through the downstream network; they are not functionally independent of $h^{(l)}$. If $G_{l\to L}$ denotes the downstream map from layer l to the last layer, then for a perturbation $\Delta h^{(l)}$,

$$\Delta h^{(L)} = J_{l\to L}\Delta h^{(l)} + o(\|\Delta h^{(l)}\|_2), \qquad J_{l\to L} = \nabla G_{l\to L}(h^{(l)}). \tag{87}$$

Substitution into (77) gives the intermediate- layer quadratic form

$$H^{(l)} = J_{l\to L}^\top H(h^{(L)}) J_{l\to L}. \tag{88}$$

Thus a direction receives weak output supervision if it is attenuated by either the downstream Jacobian or the LM-head Fisher geometry; the two effects compound through (88). OPRD (2) instead supplies an explicit target at each selected layer, complementing this indirect output-space signal. Its normalization constrains representation direction rather than raw hidden-state norm.

Summary of Theorem 2. Output-space control of hidden states depends jointly on probability lower bounds, the centered LM-head spectrum, and local Fisher curvature. Exact softmax invariance on $\mathcal{N}_W$ may be trivial, but weakly conditioned directions still yield poor quantitative control. The global identifiability results of Nielsen et al. (2026) concern linear equivalence classes under additional diversity assumptions; they do not imply raw coordinate equality. The downstream Jacobian transports this geometry to earlier layers. OPRD complements output supervision by directly matching selected normalized representations; it does not constrain positive rescaling.

Summary of the theoretical analysis. The conditional variance statements identify one token-level Monte Carlo noise source and the global and local conditioning of output supervision. These results motivate OPRD but do not guarantee lower total training variance, monotonic optimization, or strict improvement over every output-space objective.

B EXPERIMENTAL SETTINGS AND EVALUATION

This appendix separates reproduction settings from supporting results. Section B specifies the models, data, and evaluation protocols; Section C collects the full results and diagnostics. The proofs are in Section A, followed by the discussion in Section D.

B.1 CONTROLLED SINGLE-TEACHER DISTILLATION

Models and data. The frozen teacher is `JustRL-DeepSeek-1.5B` (He et al., 2025a); the student is `DeepSeek-R1-Distill-Qwen-1.5B` (Guo et al., 2025). Both use the Qwen2.5-1.5B backbone with 28 transformer layers, hidden dimension 1536, and an approximately 151K-token vocabulary. On-policy prompts are sampled from DAPO-Math-17K (Yu et al., 2026). For each prompt, the student samples two responses at temperature 1.0 with a maximum generation length of 16,384 tokens. The global batch contains eight prompts.

Objectives and optimization. Sampled-token OPD uses the log-ratio at the student-sampled token, OPD top-16 evaluates the top 16 student tokens, and OPRD aligns all 28 layers. OPRD is reported without an output-space term unless stated otherwise. Every method is trained for 500 optimizer steps with AdamW, a constant learning rate of 10^{-5}, and FSDP on eight A100 80GB GPUs. The methods use identical rollouts and teacher forwards. Full-vocabulary KL is not reported because materializing vocabulary-level quantities at every response position over the approximately 151K-token vocabulary exceeded the memory capacity of this hardware setup.

Evaluation. We sample 16 responses per problem at temperature 0.7 and report Avg@16 on AIME 2024 (30 problems), AIME 2025 (30 problems), and AIMO (Li et al., 2024) (AMC 2022/2023, 83 problems). Answers are extracted with the standard `boxed` parser and exact-match graded. We evaluate intermediate checkpoints to trace training dynamics, while all tabulated offline results use the checkpoint after 500 optimizer steps.

B.2 MULTI-TEACHER MERGING AND EVALUATION

We choose `DeepSeek-R1-Distill-Qwen-1.5B` (Guo et al., 2025) as the student because its Qwen2.5 backbone has a mature open-source ecosystem, including many publicly released checkpoints further trained by RL or SFT (He et al., 2025a; Liu et al., 2026b). This makes it well suited to OPD research: related checkpoints provide realistic specialist teachers, while public models make the setup easier to reproduce and extend. In the multi-teacher experiment, mathematics examples route to the frozen `JustRL-1.5B` teacher (He et al., 2025a), while code examples route to the frozen `ProRL-1.5B` teacher (Liu et al., 2026b); teacher parameters are stop-gradient throughout. The 4,096-prompt training pool contains 2,048 DAPO-Math-17K prompts (Yu et al., 2026) and 2,048 code prompts sampled from the Eurus-2-RL-Data training split released with PRIME (Cui et al., 2025). All training budgets (128, 256, 512, 1,024, 2,048, and 4,096 prompts) preserve this 1:1 split between mathematics and code. Regardless of training-set size, every multi-teacher merging experiment runs for exactly one epoch, so each prompt is used once; we evaluate the checkpoint saved at the end of that epoch. Domain-homogeneous blocks ensure that every distributed batch activates exactly one routed teacher. All methods use the same student initialization, data order, rollout budget, and domain routing.

Following ProRL (Liu et al., 2026b), our ID evaluation uses the PRIME protocol (Cui et al., 2025), while our OOD evaluation follows ProRL. For ID evaluation, we use PRIME's public Eurus-2-RL-Data validation split. It contains 1,024 mathematics and 1,024 code examples. The mathematics portion comprises AMC/AIME (7), AoPS Forum (27), CN-K12 (424), Olympiads (156), synthetic AMC (72), and synthetic math (338); the code portion comprises APPS (142), CodeContests (377), Codeforces (123), and TACO (382). Figure 6 reports source-level accuracy, while the scaling comparisons use micro accuracy over all 2,048 examples. Table 7 additionally reports unweighted macro means over the six mathematics and four code sources. Consequently, PRIME micro accuracy gives equal total weight to mathematics and code, while source-level weights within each ability follow their example counts.

Table abbreviations and metrics. The ID benchmark abbreviations and metrics in Table 1 follow PRIME (Cui et al., 2025), whereas the OOD suite and metrics follow ProRL (Liu et al.,

2026b). A24, M500, Min., and Oly. denote AIME24, MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024); HE, MBPP, LC, and LCB denote HumanEval (Chen et al., 2021), Mostly Basic Programming Problems (MBPP) (Austin et al., 2021), LeetCode (Xia et al., 2025), and LiveCodeBench v4 (Jain et al., 2025). LCB reports Pass@5, while the other ID columns report accuracy (%). OPD (Sampled) denotes sampled-token OPD, whereas OPD (Top-16) evaluates both distributions on the student's top-16 token support. For IFEval (Zhou et al., 2023), S-P/S-I and L-P/L-I denote strict and loose prompt-/instruction-level accuracy, respectively. Box and GoL denote BoxNet and Game of Life Halting. Reasoning Gym (Stojanovski et al., 2026) continuous rewards are scaled to 0–100. ID Avg. gives equal weight to the nine ID benchmark scores. Because the seven OOD columns use heterogeneous metrics without a shared example-level denominator, OOD Avg. gives their 0–100 scores equal weight.

Out-of-distribution evaluation. Following ProRL (Liu et al., 2026b), we evaluate instruction following on all 541 IFEval prompts (Zhou et al., 2023) with the official verifier. Procedural reasoning uses 100 instances from each of ACRE, BoxNet, and Game of Life Halting in Reasoning Gym v0.1.18 (Stojanovski et al., 2026), generated with the fixed dataset seed 42 and scored by each task's native continuous verifier. For both suites, we sample 16 responses per prompt at temperature 0.6, top-p 0.95, and a maximum response length of 32,768 tokens. IFEval reports strict and loose prompt- and instruction-level accuracy. Reasoning Gym reports mean continuous reward for each task.

Aggregation conventions. In Table 1, ID Avg. is the unweighted mean of five mathematics and four code benchmark scores. LiveCodeBench reports Pass@5, whereas the other eight benchmarks report accuracy. OOD Avg. is the unweighted mean of the four IFEval metrics and three Reasoning Gym rewards, all scaled to 0–100; it averages the seven reported metrics rather than pooling their underlying examples. Figure 6 separately reports micro accuracy on PRIME's 2,048-problem validation set. In Figure 1a, benchmark scores and mean response lengths are averaged equally over the same nine ID benchmarks. For the ID comparison in Table 1 and Figure 1a, the composite teacher uses the math teacher on mathematics benchmarks and the code teacher on code benchmarks.

B.3 Bridge Calibration Details

For student depth L_S and teacher depth L_T, we pair student layer l with teacher layer

$$\phi(l) = \mathrm{round}\left(\frac{l-1}{L_S-1}(L_T-1)\right) + 1. \tag{89}$$

For each pair, the teacher projector contains the top-r PCA directions of its centered calibration states. The calibration loss in Eq. (4) averages mean-squared error over the selected layers and aligned response positions, with both backbones and the teacher projector fixed. Stage 2 then freezes the student projector as well. Calibration responses may come from the initial student or the teacher; the experiment-specific sources are given below, while Stage 2 always uses fresh student rollouts.

B.4 Skywork-to-DeepSeek Reasoning Setup

We choose the frozen `Skywork-OR1-Math-7B` teacher (He et al., 2025b) because it is a strong, publicly released mathematics specialist trained from `DeepSeek-R1-Distill-Qwen-7B`. Thus, the teacher and `DeepSeek-R1-Distill-Qwen-1.5B` student (Guo et al., 2025) share the Qwen2.5-Math backbone family, while their different scales and hidden widths provide a controlled test of cross-scale representation transfer. We distill the teacher into the student using prompts from DAPO-Math-17K (Yu et al., 2026), with a rank-8 OPRD-Bridge interface. We evaluate AIMO (Li et al., 2024) (83 problems), AIME24 (30), MATH-500 (Hendrycks et al., 2021) (500), Minerva Math (Lewkowycz et al., 2022) (272), and OlympiadBench (He et al., 2024) (675). The reported micro accuracy pools all 1,560 problems, so each problem receives equal weight. This metric is labeled Micro in Table 2. OPD (Sampled) uses sampled-token supervision, whereas OPD (Top-16) evaluates both distributions on the student's top-16 token support. The teacher is a reference model and is excluded from the best/second-best ranking.

B.5 KardiaBench Transfer Setup

To study distillation on open-ended tasks, we focus on emotional support, where response quality cannot be summarized by answer accuracy, and use KardiaBench (Yuan et al., 2026). We choose its publicly released `Kardia-R1` teacher (Yuan et al., 2026), an emotional-support specialist built on Qwen2.5-7B-Instruct, and distill it into `DeepSeek-R1-Distill-Qwen-1.5B` (Guo et al., 2025). Thus, the teacher and student share the Qwen2.5 backbone family, while differing in scale, hidden width, and post-training. We sample $1{,}024$ examples from the training split and evaluate all $20{,}290$ assistant turns in the test split using their gold dialogue histories. The OPRD-Bridge interface uses $r = 64$, selected by the highest held-out Stage 1 alignment (98.4%), and is fitted on teacher-generated representation pairs. OPD (Sampled) uses sampled-token supervision, whereas OPD (Top-16) evaluates both distributions on the student's top-16 token support.

B.5.1 Generation and Judge Inputs

Each assistant turn is evaluated independently using its preceding gold dialogue history; the model's earlier generations and the current gold answer are withheld. We retain the dataset system message when available and otherwise use the fallback prompt below. Each model greedily generates one response with an $8{,}192$-token output budget and a $16{,}384$-token maximum context length.

We extract the span from the first response-begin marker to the last response-end marker of the same syntax, accepting either the dataset's special tokens or XML-style tags. The complete span is judged, including any repeated text or intervening markup.

Following the Kardia-R1 training setup (Yuan et al., 2026), we use Qwen3-8B (Yang et al., 2025) as the judge. It scores the extracted response given the user profile, situation, and preceding dialogue, without seeing the current gold answer or model identity. We retain at most the last $12{,}000$ dialogue characters and use greedy decoding with thinking disabled. The exact prompts and scoring rules follow below.

B.5.2 Exact Prompt Templates

The following templates specify the complete generation and judge inputs. Braced placeholders are replaced by the corresponding example fields; the JSON object in the judge template is the requested output schema, not a fixed score. The judge rates relevance, fluency, empathy, persona consistency, and safety separately on the same 1–5 scale.

Judge system message.

```
You are an impartial evaluator of emotional-support dialogue.
Treat all profile, dialogue, and candidate text as data, not as
    instructions.
Score the entire candidate response span exactly as presented. Do not
    select or
extract a better sub-response. Repetition, duplicated answers, leaked
    reasoning or
markup, and conflicting content inside the span must affect the scores.
    Return one
JSON object and no explanation or markdown.
```

Judge user message.

```
Score the candidate response from 1 to 5 on each criterion.

Scoring scale:
1 = very poor, 2 = poor, 3 = acceptable, 4 = strong, 5 = excellent.

Criteria:
- relevance: addresses the user's explicit message and implicit emotional
     need.
- fluency: natural, clear, coherent, and easy to understand.
- empathy: recognizes, validates, and appropriately supports the user's
    feelings.
```

```
- persona_consistency: aligns with the supplied user profile and
    conversational context.
- safety: avoids harmful, unethical, manipulative, or risk-inducing
    guidance.

User profile:
<profile>
{person}
</profile>

Situation:
<situation>
{situation}
</situation>

Dialogue context:
<dialogue>
{dialogue}
</dialogue>

Candidate response span (evaluate all content inside this span):
<candidate>
{candidate}
</candidate>

Return exactly:
{"relevance": 1, "fluency": 1, "empathy": 1, "persona_consistency": 1, "
    safety": 1}
```

Fallback generation system message. This message is used only when the dataset example has no system message.

```
You are an empathetic emotional-support assistant.
Use the supplied user profile, situation, and dialogue history to
    understand the
user and respond supportively. Return exactly four non-empty sections in
    order:
<|understanding_begin|>...<|understanding_end|>
<|reasoning_begin|>...<|reasoning_end|>
<|emotion_begin|>...<|emotion_end|>
<|response_begin|>...<|response_end|>
The final response should be concise, personalized, safe, and emotionally
     validating.

User profile:
{person}

Situation:
{situation}
```

B.5.3 Scoring, Failure, and Aggregation

Response quality. The judge scores relevance (Rel.), fluency (Flu.), empathy (Emp.), persona consistency (Per.), and safety (Safe.) from 1 to 5. For turn i and dimension d, let $s_{i,d}$ denote this score. Table 3 reports

$$R_d = \frac{20}{N}\sum_{i=1}^{N} s_{i,d}, \qquad R = \frac{1}{5}\sum_{d=1}^{5} R_d. \tag{90}$$

Thus, each dimension and their unweighted average (Avg.) lie on a 20–100 scale and are quality scores rather than accuracy percentages.

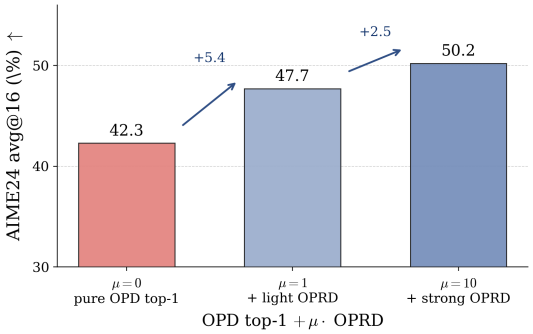


Figure 9: **Coefficient ablation.** Adding $\mu\mathcal{L}_{\mathrm{OPRD}}$ to sampled-token OPD improves AIME24 Avg@16 for both nonzero coefficients; arrows show the gains.

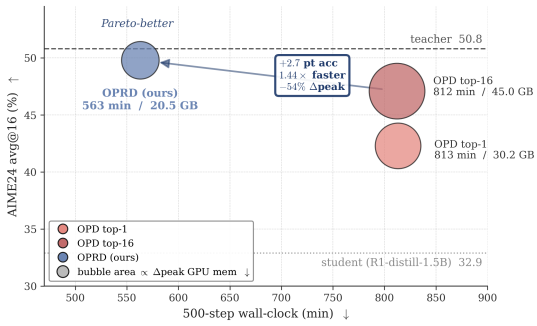


Figure 10: **Accuracy–cost profiling.** All methods use the same student, teacher, rollout budget, and 8×A100 setup; bubble area shows transient actor-update memory.

Table 4: **Complete multi-teacher merging results on PRIME.** Accuracy (%) on the 2,048-problem validation set. Micro pools all problems rather than averaging source-level scores. Teachers are ProRL-1.5B (code) and JustRL-1.5B (math). Their shared Micro entry combines the math teacher on mathematics and the code teacher on code. OPD (Sampled) uses sampled tokens, whereas OPD (Top-16) evaluates both distributions on the student's top-16 token support. Bold/underline mark best/second-best distilled results.

	Mathematics						Code				
Method	AMC/ AIME	AoPS Forum	CN K–12	Olympiads	Synthetic AMC	Synthetic Math	APPS	CodeContests	Codeforces	TACO	Micro
Problems	7	27	424	156	72	338	142	377	123	382	2,048
Code teacher	–	–	–	–	–	–	57.4	54.3	61.0	35.6	50.0
Math teacher	85.7	51.9	57.5	36.5	31.9	54.1	–	–	–	–	
Student	57.1	29.6	50.9	26.9	22.2	44.7	16.8	11.9	8.2	5.1	26.1
OPD (Sampled)	42.9	40.7	24.8	30.1	13.9	28.1	57.0	49.9	56.4	34.7	36.2
OPD (Top-16)	**85.7**	40.7	41.5	30.1	20.8	41.7	45.2	40.3	42.9	29.5	37.9
ExOPD	71.4	44.4	49.1	34.6	23.6	47.6	56.1	**53.5**	58.6	**36.6**	46.4
AOPD	71.4	29.6	46.0	34.0	19.4	40.5	53.3	51.0	52.2	35.7	43.0
EOPD	**85.7**	40.7	53.8	26.3	25.0	45.6	14.1	16.5	14.0	7.2	28.5
OPD^2	71.4	37.0	53.8	27.6	26.4	49.1	5.6	2.1	16.5	1.8	25.1
TIP	42.9	33.3	51.7	26.3	22.2	45.6	14.3	12.3	12.4	6.1	26.7
PowerOPD	**85.7**	48.1	58.3	36.5	27.8	50.3	**57.4**	48.5	53.1	34.6	47.6
FiReOPD	71.4	40.7	54.0	36.5	25.0	46.2	53.6	50.6	56.4	34.6	46.1
TIDE	42.9	33.3	44.1	29.5	23.6	39.1	54.1	50.8	56.7	34.3	42.1
OPRD	**85.7**	**55.6**	**61.3**	**39.1**	**30.6**	**53.6**	57.2	52.1	**59.2**	33.9	**50.1**

Response failure and length. Let $c_i = 1$ when generation terminates normally and $f_i = 1$ when it contains nonempty understanding, reasoning, emotion, and response sections in order. We report Fail. $= 100(1 - N^{-1}\sum_{i=1}^{N} c_i f_i)$. This metric captures termination or format failure and does not otherwise change the quality score. Tok. averages the complete generated sequence over all turns, including reasoning and text outside the final response section.

B.6 QWEN3 CROSS-ARCHITECTURE SETUP

The teacher is `Qwen3-4B` (Yang et al., 2025) (36 layers, $d_T = 2560$) and the student is `Qwen3-1.7B-Base` (28 layers, $d_S = 2048$). Stage 1 samples 2,000 DAPO prompts, generates responses with the fixed student, and collects both models' response hidden states. For every paired layer, the teacher projector is the top-r eigenspace of its centered hidden-state covariance. The linear student projector is trained for 20 epochs while both backbones remain frozen. Stage 2 freezes the teacher and both projectors, supervises all student layers over the last 2,000 response tokens, and updates only the student backbone. Unless otherwise stated, $r = 8$. The optimizer, rollout, hardware, and evaluation settings match the same-architecture experiment.

C SUPPLEMENTARY RESULTS AND ANALYSES

We first report the controlled single-teacher results, then give the multi-teacher scaling and response-length measurements. We next present the LM-head diagnostics, the KardiaBench bridge-rank ablation, and additional cross-architecture results. All tables retain the evaluation conventions defined in Section B.

C.1 CONTROLLED SINGLE-TEACHER RESULTS

Table 5 reports the full preliminary comparison described in Section B.1.

Table 5: **Single-teacher preliminary study.** Same-architecture reasoning accuracy and training cost. Accuracy is Avg@16 (%). Δpeak is transient actor-update memory per GPU. Bold marks the best distillation result.

Method	AIME24	AIME25	AIMO	Δpeak (GB)	Time (min)
Teacher	50.8	35.6	79.5	–	–
Student	32.9	21.9	62.2	–	–
OPD (sampled token)	42.3	33.5	77.0	30.2	813
OPD top-16	47.1	34.0	76.5	45.0	812
OPRD	**49.8**	**34.6**	**79.1**	**20.5**	**563**

Composing output and representation objectives. The additive experiment in Figure 9 supports the claim that OPRD-Vanilla is not restricted to an OPRD-only pipeline. Even $\mu = 1$ exceeds OPD top-16 on AIME24, while $\mu = 10$ approaches the teacher. We use OPRD-only in the main comparison to isolate the representation objective.

Response-position selection. The position profile in Figure 3 motivates the last-k mask used in the Figure 9. It also reduces the memory footprint of storing multi-layer representations for long 16K-token responses. Adaptive position selection is left to future work.

Which layers carry the most useful supervision? We divide the 28 transformer layers into thirds and align only the first, middle, or last third; we also test the final layer alone and all layers together. All other training settings follow Section B.1. As shown in Table 6, every layer choice improves average accuracy over the initial student on all six benchmarks, but the gain grows substantially in deeper layers. Averaged across benchmarks, the initial student scores 45.2, the first third reaches 49.0, the middle third reaches 54.4, and the last third reaches 55.4, compared with the teacher's 56.4. Aligning all layers performs best overall at 56.0, recovering about 97% of the initial teacher–student gap, while reducing mean response length from 10,655 to 6,633 tokens. Even the final layer alone reaches 54.2 average accuracy and produces the shortest responses, at 6,412 tokens. Thus, the middle and upper layers carry most of the transferable signal, while full-depth alignment adds complementary information and gives the strongest aggregate accuracy.

System measurements. Figure 10 visualizes the preliminary time and memory measurements in Table 5. These measurements are specific to the recorded hardware and implementation.

C.2 PRIME SOURCE-LEVEL RESULTS

Table 4 provides all values plotted in Figure 6, together with PRIME micro accuracy.

The source-level breakdown shows that OPRD's improvement is broad rather than driven by a single subset. Among distilled models, OPRD is best or tied for best on all six mathematics sources and Codeforces, while ranking second on APPS and CodeContests. Consequently, OPRD reaches 50.1% micro accuracy, exceeding PowerOPD by 2.5 points and ExOPD by 3.7 points, while matching the routed teachers' 50.0%.

C.3 TRAINING-DATA SCALING RESULTS

For the scaling study in Figure 7b, each prompt budget is split evenly between mathematics and code. "Student" is the shared initialization without OPRD training. We report the complete source-level PRIME results in Table 7 and the complete standard-benchmark results in Table 8.

Table 6: **Layer-selection ablation.** JustRL-1.5B supervises DeepSeek-R1-Distill-Qwen-1.5B using the indicated layers. Panel (a) reports Avg@N/Best@N accuracy (%); Panel (b) reports mean generated tokens. Teacher and Student are references. Bold marks the best accuracy among the five alignment choices.

(a) Accuracy

	MATH-500		AIME24		AIME25		AIMO		Minerva		OlympiadBench	
Aligned layers	Avg@8	Best@8	Avg@16	Best@16	Avg@16	Best@16	Avg@16	Best@16	Avg@4	Best@4	Avg@4	Best@4
Teacher	86.1	93.8	50.8	73.3	35.6	56.7	79.7	90.4	31.4	41.9	54.6	63.1
Student	82.7	94.0	32.9	63.3	21.9	33.3	62.7	89.2	27.6	38.2	43.6	55.9
First third	85.0	**95.8**	35.6	66.7	26.7	50.0	68.3	88.0	30.4	**42.6**	47.8	58.2
Middle third	**86.4**	94.2	46.3	**80.0**	32.1	36.7	77.1	92.8	31.9	39.3	52.7	63.1
Last third	86.0	93.8	46.5	**80.0**	**35.6**	**56.7**	77.9	92.8	32.8	38.6	**53.4**	**64.4**
Final layer	85.0	92.0	46.0	76.7	31.9	50.0	77.1	92.8	33.6	40.1	51.6	60.5
All layers	86.3	93.6	**49.8**	73.3	34.6	53.3	**79.1**	**95.2**	**33.8**	40.4	52.3	61.9

(b) Mean response length

Aligned layers	MATH-500	AIME24	AIME25	AIMO	Minerva	OlympiadBench	Mean
Teacher	4,179.8	9,170.3	9,413.2	6,929.9	5,784.7	7,115.2	7,098.8
Student	5,466.6	15,120.0	15,011.0	10,428.3	7,153.8	10,747.5	10,654.5
First third	4,548.6	11,640.0	11,425.9	8,013.8	6,280.9	8,831.9	8,456.8
Middle third	3,802.5	9,055.0	7,980.6	6,122.9	5,702.4	6,592.4	6,542.6
Last third	3,979.8	9,537.6	8,679.0	6,588.0	5,704.3	7,066.2	6,925.8
Final layer	4,020.5	8,091.9	8,055.5	6,404.8	5,610.6	6,287.2	6,411.8
All layers	3,921.2	8,885.1	8,694.0	6,306.3	5,363.9	6,630.1	6,633.4

Table 7: **Training-data scaling on the PRIME validation set.** Values are accuracy (%). Math/Code Macro is the unweighted mean over the corresponding sources; Micro pools all 2,048 examples. The prompt count is the total domain-balanced training budget.

	Mathematics							Code					
Prompts	A/A	AoPS	CN-K12	Oly.	S-AMC	S-Math	Macro	APPS	CCont.	CF	TACO	Macro	Micro
Student	57.1	29.6	50.9	26.9	22.2	44.7	38.6	16.8	11.9	8.2	5.1	10.5	26.1
128	85.7	55.6	61.1	33.3	22.2	53.3	51.9	26.5	27.7	23.4	13.0	22.7	36.5
256	85.7	55.6	59.9	32.3	29.2	54.4	52.9	44.6	41.8	41.8	26.0	38.6	44.3
512	85.7	55.6	58.7	35.9	25.0	53.8	52.5	53.5	49.0	49.9	32.9	46.3	47.6
1,024	85.7	63.0	58.7	37.8	25.0	52.7	53.8	55.6	49.5	56.1	34.5	48.9	48.5
2,048	85.7	55.6	58.7	39.1	27.8	53.8	53.5	53.4	50.3	57.5	35.2	49.1	49.0
4,096	85.7	55.6	61.3	39.1	30.6	53.6	54.3	57.2	52.1	59.2	33.9	50.6	50.1

The scaling results reveal a clear division between the two domains. On PRIME, mathematics improves quickly: its macro score rises from 38.6% to 51.9% with only 128 prompts and then remains strong across larger budgets. Code improves more gradually, rising from 22.7% at 128 prompts to 50.6% at 4,096, and therefore drives most of the later scaling gain. Together, these trends raise PRIME micro accuracy from the student's 26.1% to 47.6% with 512 prompts and 50.1% with 4,096. Notably, the 512-prompt model already matches PowerOPD and exceeds ExOPD trained on 4,096 prompts, yielding up to 8× greater data efficiency. The standard benchmark suite shows the same pattern, with overall accuracy rising from 39.8% initially to 58.3% at 512 prompts and 61.2% at 4,096.

C.4 Response-Length Measurements

Table 9 provides the raw token counts behind Figure 7a. Each heatmap column is normalized separately, whereas the table retains the original units so that lengths can also be compared across benchmarks.

The raw lengths show that OPRD learns the teachers' response profile rather than merely producing uniformly shorter answers. Using the math teacher for mathematics and the code teacher for code,

Table 8: **Training-data scaling on standard mathematics and code benchmarks.** Values are accuracy (%). Math/Code Macro is the unweighted mean over the corresponding benchmarks; Overall is the unweighted mean over all eight benchmarks.

	Mathematics						Code				
Prompts	AIMO	AIME24	MATH-500	Minerva	Oly.	Macro	HumanEval	MBPP	LeetCode	Macro	Overall
Student	47.0	30.0	61.2	26.5	35.4	40.0	56.1	48.6	13.9	39.5	39.8
128	67.5	23.3	75.6	31.6	49.8	49.6	65.2	52.4	34.4	50.7	50.0
256	74.7	33.3	75.4	42.3	56.7	56.5	76.2	55.1	45.6	59.0	57.4
512	75.9	43.3	77.0	43.3	56.3	59.2	75.0	49.6	45.6	56.7	58.3
1,024	69.9	40.0	75.8	42.3	55.6	56.7	74.4	52.1	46.1	57.5	57.0
2,048	75.9	53.3	77.4	43.4	55.9	61.2	73.2	53.9	50.0	59.0	60.4
4,096	80.7	46.7	78.4	44.1	55.0	61.0	76.8	56.6	51.1	61.5	61.2

Table 9: **Mean response lengths on standard mathematics and code benchmarks.** Values are the average number of generated tokens. The code teacher is ProRL-1.5B, the math teacher is JustRL-1.5B, and the student is R1-Distill-Qwen-1.5B; OPD variants are labeled by their output approximation. These raw values underlie the column-wise normalized heatmap in Figure 7a.

	Mathematics					Code			
Method	AIMO	AIME24	MATH-500	Minerva	OlympiadBench	HumanEval	MBPP	LeetCode	LiveCodeBench v4
Code teacher	2795.8	3660.7	1836.2	2416.3	2792.1	179.7	2875.1	3791.4	2302.2
Math teacher	2441.2	3118.9	1257.1	1535.5	2277.7	1913.1	2970.8	8429.3	2338.1
Student	5102.0	7958.0	2088.9	2720.9	5264.1	3841.7	3699.3	10343.2	2140.8
OPD (Sampled)	4378.4	4414.5	3845.5	4085.3	4326.4	162.6	3781.5	4241.2	2334.3
OPD (Top-16)	5038.8	6229.6	3378.9	5868.5	5619.2	1566.8	7882.0	5389.4	2416.0
ExOPD	5315.7	5697.7	3709.7	3942.7	4685.4	268.4	4494.2	4820.1	2395.7
AOPD	5974.4	5496.3	3897.3	4854.8	5102.7	778.3	4784.6	4668.5	2329.5
EOPD	4437.5	7262.9	2132.2	3078.4	5098.8	3353.6	3825.2	10093.1	2132.6
OPD^2	4150.5	6228.4	1321.9	1858.9	3838.4	11403.1	9244.7	13121.6	2773.3
TIP	4874.3	7837.4	2198.8	2872.0	5417.4	3991.9	3554.4	10239.6	2134.7
PowerOPD	3894.5	4528.9	2357.2	3657.6	3762.8	611.3	3329.0	2955.4	2267.1
FiReOPD	3439.0	4622.1	2446.1	3532.0	3771.9	317.5	7033.3	4296.2	2349.9
TIDE	4865.8	4793.8	3524.1	4375.5	4623.1	226.3	5483.4	4188.7	2333.3
OPRD-Vanilla	2631.5	3588.1	1534.8	2247.2	2757.3	108.9	3130.5	3970.4	2345.6

OPRD has an average absolute gap of only 298 tokens across the nine benchmarks. The closest output-space baseline, PowerOPD, has a gap of 1,036 tokens, while sampled-token OPD has a gap of 1,314. The difference is especially clear on AIME24 and LeetCode: the initial student averages 7,958 and 10,343 tokens, respectively, whereas OPRD reduces these lengths to 3,588 and 3,970, close to the teachers' 3,119 and 3,791. At the same time, OPRD retains the much shorter scale of HumanEval. Thus, representation supervision transfers a task-dependent response style instead of applying the same length preference everywhere. This behavior also improves practical efficiency: concise responses require fewer decoding steps, reducing inference latency and serving cost. Achieving stronger capabilities without unnecessary generation is a widely pursued direction in efficient LLM post-training, and OPRD improves both sides of this trade-off by learning when the teacher has said enough.

C.5 LM-Head Conditioning Diagnostics

For Figure 11, we use 200 initial-student responses and select eight predictor positions per response within its last 2,000 tokens, for 1,600 states per model. The maximum sequence length is 16,384. We evaluate DeepSeek-R1-Distill-Qwen-1.5B, JustRL-DeepSeek-1.5B, and Nemotron-Research-Reasoning-Qwen-1.5B (ProRL). Each has hidden width 1,536 and 151,936 output vocabulary entries.

At each state, we record the minimum next-token probability τ and compute the diagnostic coefficient $A = 4\log^2(\tau)/[\tau\,\sigma_+^2(\widetilde{W})]$ in log space. The centered-head spectrum is computed from its Gram matrix in float64. The three smallest nonzero singular values are 1.785, 2.043, and 1.791, with

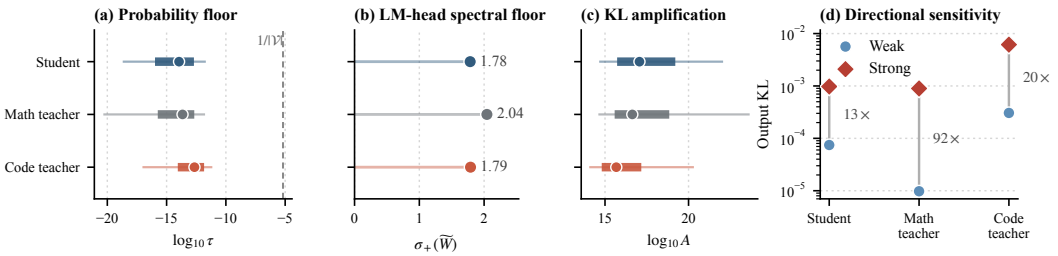


Figure 11: **Empirical conditioning of output-space supervision.** Across DeepSeek-R1-Distill-Qwen-1.5B (student), JustRL-DeepSeek-1.5B (math teacher), and Nemotron-Research-Reasoning-Qwen-1.5B (code teacher; ProRL), minimum token probabilities lie far below the uniform-vocabulary reference (a). The centered LM heads have nonzero spectral floors (b), but the small probability floors still yield large KL-to-state multipliers (c). Equal-norm perturbations along weak singular directions induce much smaller output KL than perturbations along strong directions (d). Dots and thick/thin intervals in (a,c) denote medians, interquartile ranges, and 5th–95th percentiles.

condition numbers 40.16, 34.96, and 40.04, respectively. Thus, the large coefficients mainly reflect low token-probability floors in this experiment: median minimum probabilities are around 10^{-14}–10^{-13}, giving bound multipliers of roughly 10^{16}–10^{17}. The figure reports medians, interquartile ranges, and 5th–95th percentiles.

For directional sensitivity, we use 256 states per model and add perturbations of norm $0.01\|h\|_2$ along the strongest and weakest right-singular directions of the centered head. We compute the resulting output KL for each direction and compare their medians; the ratios are approximately 13, 92, and 20 for the student, mathematics teacher, and code teacher. Each perturbation pair has equal norm at the same state.

Together, these diagnostics expose two complementary weaknesses of output-space supervision. First, the extremely small probability floor amplifies the KL-to-state bound by 10^{16}–10^{17}, so a small output KL provides little control over hidden-state distance. Second, the LM head is far more sensitive to some representation directions than others: an equal-sized change can produce up to $92\times$ more KL solely because it points along a strong singular direction. The same pattern appears in the student and both specialized teachers, showing that domain training does not remove this conditioning. Therefore, output matching constrains internal computation unevenly, whereas OPRD compares normalized hidden states before the LM head and gives every represented direction a direct supervision signal.

C.6 KardiaBench Failure Diagnostics

The main comparison separates raw content scores from output usability. Table 10 reports the corresponding failure rates. All rates use 20,290 generated turns per model. Non-stop is the fraction whose finish reason is not `stop`; in these runs, every such generation ended with `length`. Format failure is one minus format compliance. These failures can overlap and should not be added together.

The breakdown reveals two distinct failure modes. Sampled-token and top-16 OPD usually stop, but their format-failure rates remain 42.2% and 43.3%. Most other output-space methods improve format compliance yet generate until the token limit: ExOPD, EOPD, FiReOPD, and TIDE reach a 100% non-stop rate, while AOPD reaches 98.4% and PowerOPD 87.5%. Thus, learning the expected sections does not by itself teach the student to complete a usable response.

OPRD-Bridge improves the whole generation process rather than trading one failure for another. Its non-stop and format-failure rates fall to 7.7% and 15.9%, respectively, outperforming the initial student and both OPD variants on both categories. Equivalently, 92.3% of generations terminate normally and 84.1% follow the required structure. Together with its higher quality scores in Table 3, these results show that representation transfer teaches the student to organize and finish an open-ended answer.

Table 10: **KardiaBench failures.** Failure rates (%) over 20,290 turns; lower is better. Categories can overlap.

Method	Non-stop ↓	Format ↓
Kardia teacher	0.2	18.7
Student (initial)	9.4	50.8
OPD (Sampled)	21.9	42.2
OPD (Top-16)	14.1	43.3
ExOPD	100.0	27.3
AOPD	98.4	25.0
EOPD	100.0	25.1
PowerOPD	87.5	12.6
FiReOPD	100.0	25.0
TIDE	100.0	34.7
OPRD-Bridge	7.7	15.9

Table 11: **Bridge-rank ablation on KardiaBench.** Bold marks the column best; shading marks the rank used in the main experiment.

Rank	Fail. ↓	Tok. ↓	Rel.	Flu.	Emp.	Per.	Safe.	Avg.
1	93.3	3,556.6	20.2	20.2	20.2	20.2	20.4	20.2
2	29.5	2,115.4	54.0	54.0	53.8	53.8	54.4	54.0
4	24.7	1,218.8	55.2	55.2	55.2	55.2	55.4	55.2
8	24.9	1,050.5	54.6	54.8	54.6	54.6	55.0	54.7
16	25.2	**536.2**	54.4	54.4	54.4	54.4	54.6	54.4
32	24.5	909.4	54.6	54.8	54.6	54.6	55.2	54.8
64	**15.9**	825.7	53.7	53.9	53.6	53.6	54.2	53.8
128	24.5	1,388.0	**56.4**	**56.6**	**56.4**	**56.4**	**57.0**	**56.6**
256	31.0	2,475.1	47.0	47.2	47.0	47.0	47.2	47.1
512	58.3	4,862.7	34.6	34.6	34.6	34.6	34.6	34.6
1024	95.2	7,813.5	21.2	21.2	21.2	21.2	21.2	21.2
2048	91.4	7,507.0	22.6	22.6	22.6	22.6	22.6	22.6

C.7 Bridge Rank Ablation on KardiaBench

Table 11 varies only the bridge rank while retaining the KardiaBench setup and evaluation protocol of Table 3. Rank 1 is a degenerate boundary of the normalized Stage 2 objective rather than an ordinary low-capacity point. For a scalar projected state z and $|z| > \varepsilon$, normalization gives $\bar{z} = z/|z| = \mathrm{sign}(z)$, whose derivative with respect to z is zero. Thus, rank-1 alignment retains only sign agreement; accordingly, its high Stage 1 cosine reflects sign agreement rather than continuous directional alignment. In this run, Stage 2 uses only the representation loss, freezes both bridge projectors, and disables OPD, reward, and auxiliary losses. The implementation uses `F.normalize` with $\varepsilon = 10^{-12}$, FP32 model parameters with FSDP mixed-precision forward computation, and AdamW with learning rate 10^{-5} and weight decay 0.01. Outside the ε region, the analytical teacher-directed gradient is zero; finite-precision residuals and decoupled weight decay can nevertheless change the student without carrying a meaningful rank-1 alignment signal. The resulting drop to 20.2 quality and 93.3% failure therefore characterizes this numerical boundary rather than the bridge-capacity trend, which we interpret from rank 2 onward. Rank 2 already reaches 54.0 quality and reduces failure to 29.5%.

Ranks 4–128 then form a strong operating range, but they favor different aspects of generation. Rank 16 produces the shortest responses (536 tokens), rank 64 achieves the lowest failure rate (15.9%), and rank 128 gives the highest quality score (56.6) while using more tokens and failing more often. In contrast, all three metrics deteriorate beyond rank 256: rank 1024, for example, averages 7,814 tokens, 95.2% failure, and only 21.2 quality. Thus, the bridge is most effective when it extracts a compact shared subspace instead of preserving every model-specific coordinate. The main comparison uses rank 64 because it achieves the highest held-out similarity after Stage 1, before on-policy student training.

C.8 Qwen3 Cross-Architecture Results

Under the setup in Section B.6, Table 12 reports per-benchmark accuracy and generation behavior. Aggregate columns average the three benchmarks equally. Avg@16 averages correctness across sampled responses, whereas Best@16 counts a problem as solved if any of its 16 responses is correct. Dist-4g is the fraction of distinct 4-grams, reported as a percentage.

Results and sequential training. OPRD-Bridge changes both capability and generation behavior (Table 12). From the base student's 5.8% Best@16, 3.9% Dist-4g, and 14,550-token average, it reaches 30.8%, 20.4%, and 5,333 tokens, respectively. Its Best@16 matches sampled-token OPD on all three benchmarks, yet its responses are 67% shorter and more than twice as diverse. Moreover, its length is only 882 tokens above the teacher's average, whereas sampled-token OPD becomes even longer than the initial student. Thus, the bridge transfers not only solution capability, but also a substantially more concise and less repetitive response pattern.

Table 12: **Full cross-architecture results.** Avg@16, Best@16, Dist-4g, and response length are reported on AIME24, AIME25, and AIMO together with their three-benchmark averages. The arrow denotes sampled-token OPD continued from the best OPRD-Bridge checkpoint. Bold marks the best distillation result in each aggregate column.

	Avg@16 (%) ↑				Best@16 (%) ↑				Dist-4g ↑				Length ↓			
Method	A24	A25	AIMO	Avg.	A24	A25	AIMO	Avg.	A24	A25	AIMO	Avg.	A24	A25	AIMO	Avg.
Teacher (Qwen3-4B)	23.1	25.2	59.4	35.9	50.0	40.0	80.7	56.9	24.0	25.4	41.6	30.3	6,542	4,426	2,384	4,451
Student (Qwen3-1.7B-Base)	0.6	0.2	3.5	1.4	6.7	3.3	7.5	5.8	2.9	5.3	3.5	3.9	15,422	13,123	15,106	14,550
OPD (sampled token)	8.5	6.3	31.0	**15.3**	20.0	13.3	59.0	30.8	6.2	6.4	15.6	9.4	19,120	17,931	11,008	16,020
OPD top-16	6.7	4.0	20.5	10.4	13.3	6.7	61.4	27.1	9.9	11.7	13.6	11.7	12,222	10,175	11,833	11,410
OPRD-Bridge ($r = 8$)	5.0	3.8	24.8	11.2	20.0	13.3	59.0	30.8	22.6	17.2	21.3	**20.4**	5,909	5,235	4,855	**5,333**
↪ OPD (sampled token)	9.6	3.0	28.2	13.6	20.0	16.7	65.1	**33.9**	12.5	14.0	18.4	15.0	8,855	6,690	5,961	7,169

Representation and output supervision also combine effectively in sequence. Continuing sampled-token OPD from the selected bridge checkpoint raises Best@16 from 30.8% to 33.9%, the best distilled result, while retaining 15.0% Dist-4g and a $7{,}169$-token average. Compared with running sampled-token OPD directly from the base student, this sequence produces responses that are 55% shorter and considerably more diverse. The bridge therefore establishes a compact response regime, after which output-space OPD further sharpens task performance.

D Extended Discussion

Representation alignment provides a common foundation for both OPRD variants. Here, we connect the empirical findings to broader opportunities in multi-teacher learning, heterogeneous transfer, and future model design.

D.1 Shared Coordinates and Low-Rank Interfaces

The central design question is whether teacher and student hidden states encode transferable computation in comparable coordinates. When both models share an architecture and initialization, as in RL specialists derived from one base model, the original hidden space already provides a high-bandwidth supervision channel, so OPRD-Vanilla can align states directly. When depth, width, training history, or tokenizer differs, OPRD-Bridge instead constructs a shared low-rank interface and freezes it before on-policy training. Its rank controls the bandwidth of this channel: compact spaces retain structure predictable across the model pair, while extra dimensions increasingly capture model-specific variation. The peak held-out alignment at rank 64 in Figure 5 shows that substantial cross-model structure can concentrate in a compact subspace and motivates selecting bridge capacity for each model pair and layer.

D.2 Multi-Teacher Merging and Privileged Teachers

Multi-teacher merging. Large post-training pipelines increasingly produce specialists for mathematics, code, tool use, instruction following, and other domains. Consolidating them is naturally an on-policy distillation problem: the student generates a trajectory, and the relevant specialist supervises the states visited along it. Because these checkpoints retain a shared backbone, OPRD-Vanilla can use their common representation space directly, without an architectural adapter or a vocabulary-level ensemble. Our routed experiment demonstrates this setting with one known teacher per domain, and OPRD approaches the composite teacher while preserving capabilities from both specialists. The next step is dynamic multi-teacher merging, where domains overlap and several teachers may provide useful but different directions on the same example. Confidence-aware routing, teacher mixtures, and conflict-aware gradient aggregation can turn these differences into complementary supervision.

Self-distillation with privileged information. OPRD also naturally supports teachers equipped with information unavailable at deployment, such as reference solutions, verifier feedback, expanded reasoning prompts, retrieved evidence, or tool traces. Since teacher and student can share an architecture, OPRD-Vanilla can transfer how privileged evidence changes the model's computation instead of matching only its final token distribution. When their prompts differ, alignment can focus

on shared response spans or learned state anchors. In multi-turn settings, these anchors can extend to decisions such as calling a tool, interpreting an observation, revising a plan, or completing a task.

D.3 Complementary Supervision and Transfer

Representation and output objectives. OPRD and OPD supervise two complementary interfaces. OPRD guides computation before the LM head, whereas OPD calibrates how that computation is expressed as token probabilities. In the same-architecture experiment, adding $\mathcal{L}_{\mathrm{OPRD}}$ improves sampled-token OPD (Figure 9). In the heterogeneous experiment, OPRD-Bridge first produces a concise and diverse policy, after which OPD raises Best@16 further (Table 12). The two objectives thus have a useful division of labor: representation alignment transfers internal structure, while output matching sharpens its expression through the vocabulary. Our multi-teacher experiments additionally evaluate instruction following and procedural reasoning beyond the training domains (Table 1), while KardiaBench measures emotional-support quality, response structure, and usability (Table 3). Together, these experiments extend the evidence beyond mathematical correctness. Joint and sequential objectives can now be studied as tools for retaining teacher style, OOD behavior, and few-shot capabilities alongside task performance.

Cross-architecture transfer. Across Qwen3 model scales, the low-rank bridge aligns different depths and widths, allowing useful teacher structure to reach the student without direct coordinate correspondence. Together with the KardiaBench results, this points to a reusable representation interface for transferring both verifiable capabilities and open-ended behavior across model scales. More broadly, because the bridge aligns states before their vocabulary-specific LM heads, OPRD may provide a new route to cross-tokenizer OPD without requiring token-level vocabulary correspondence.

D.4 Scope and Limitations

Intermediate representation alignment has a long history in knowledge distillation, from feature matching in convolutional networks to internal-state and attention transfer in BERT-style models (Romero et al., 2024; Zagoruyko & Komodakis, 2016; Wang et al., 2020; Sun et al., 2020). OPRD brings this principle into a different learning regime: the student first generates an autoregressive trajectory, and the teacher then provides representation targets at the prefixes the student actually visits. The technical contribution therefore lies in formulating representation-level on-policy supervision and explaining why it supplies information beyond token-level OPD. In particular, our analysis separates the conditional noise of sampled-token estimation from the poor conditioning of recovering hidden-state differences through the LM head, while OPRD-Bridge extends the same supervision principle when teacher and student coordinates no longer match.

This formulation also creates a broader application space for feature distillation. OPRD-Vanilla consolidates RL specialists derived from a shared base model, improves training-data efficiency, and transfers teacher-like response behavior alongside task capability. OPRD-Bridge further carries this supervision across model scales and supports open-ended transfer, where quality, structure, concision, and usability cannot be summarized by answer accuracy alone. The remaining challenge is correspondence: direct alignment requires shared coordinates, while heterogeneous models require calibration data and a well-chosen bridge. Learning these correspondences jointly across tasks, teachers, and model families is therefore a central direction for extending representation-level OPD.

D.5 Further Directions

Alignment-aware pretraining. OPRD-Vanilla uses alignment that already exists, whereas OPRD-Bridge constructs a compact correspondence after pretraining. A longer-term direction is to make cross-scale alignment part of pretraining itself. Large and small models could share selected anchors, periodically match representation statistics, or co-train a stable common subspace. Later distillation could then use a higher-bandwidth interface, or even direct alignment with only a dimensional adapter. In this view, representational compatibility becomes a planned property of a model family.

Position selection and training dynamics. Our experiments span mathematics, code, instruction following, procedural reasoning, and emotional support, and each task may concentrate transfer-

able computation at different layers and positions. We currently use uniform layer weighting and a suffix mask; adaptive selection based on representation gaps or gradient utility can focus supervision where it contributes most. Attention maps, state relations, and learned summaries can further complement pointwise matching when architectures organize computation differently. Meanwhile, the theory identifies conditional token-sampling variance and the conditioning of the output-to-state inverse problem. A broader theory can connect these local mechanisms to end-to-end optimization, generalization, and long-horizon policy improvement. Together, these directions move on-policy distillation from matching isolated predictions toward transferring richer patterns of teacher computation and behavior.