Beyond GRPO and On-Policy Distillation: An Empirical Sparse-to-Dense Reward Principle for LLM Post-Training

Hejian Sang* **Yuanda Xu*†** **Zhengze Zhou*** **Ran He*** **Zhipeng Wang** **Alborz Geramifard**

Abstract

In settings where labeled verifiable training data is the binding constraint, each checked example should be allocated to the model and reward density where it is most informative. We identify a reward-density principle that governs this allocation: sparse sequence-level reward is most useful on models that can explore and discover better behavior, while dense token-level teacher supervision is better suited for compressing that behavior into a smaller deployment model. The principle yields a simple allocation rule: use scarce labeled data upstream on the strongest available teacher, then transfer the reward-shaped behavior downstream as dense supervision.

We evaluate this rule through a four-stage workflow—teacher RL, forward-KL warmup, on-policy distillation, optional post-bridge student RL—on verifiable math with Qwen3 and Llama models. At fixed Qwen3-1.7B deployment-student size, an RL-improved 8B teacher distilled through the dense bridge outperforms direct GRPO on the same student (79.3% vs. 75.9% on MATH; 25.2% vs. 19.8% on AIME 2024, avg@16), while transfer from the same teacher *before* RL underperforms. A component ablation confirms that each stage is load-bearing: replacing the RL-improved teacher with a raw teacher costs 7.8 MATH points, removing the forward-KL warmup costs 1.7, and removing on-policy distillation costs 3.3. The teacher-quality ordering—raw-teacher transfer $<$ direct GRPO $<$ RL-teacher transfer—replicates on Llama-3.1-8B-Instruct with a Llama-3.3-70B-Instruct teacher. The operational lesson is to avoid spending scarce labeled data on the least prepared policy: use sparse reward for teacher-side discovery, dense transfer for student compression, and student-side sparse reward only after the bridge.

1 Introduction

Labeled training data is the bottleneck of LLM post-training. Pretraining text and teacher rollouts can scale with compute; labeled data for verifiable tasks does not scale so easily. Each example needs a problem with a checkable answer and a grader whose errors will not corrupt the reward. The practical question is therefore not which post-training algorithm is best in isolation, but *which model should receive each scarce labeled example, with which density of signal, and in what order*.

The default approach is to train the deployment model directly: if a 1.7B model is the deployment target for MATH, run GRPO on it directly. This paper argues for a different allocation, and for the simple reward-density principle behind it.

*Equal contribution, order decided by a coin flip.
†Correspondence to yuanda@math.princeton.edu

Preprint.

The reward-density principle. Sparse reward first defines a reward-shaped target distribution. Direct GRPO asks the deployment student to discover this target from its own sparse rollouts. The teacher-first route instead uses the labeled examples to train a stronger teacher, then treats that teacher as a dense, autoregressive proxy for the target. The bridge is the projection step: forward KL moves the student onto teacher support; OPD transfers the teacher under student occupancy. Sparse reward is used where exploration is productive, and dense teacher supervision is used where the goal is compression.

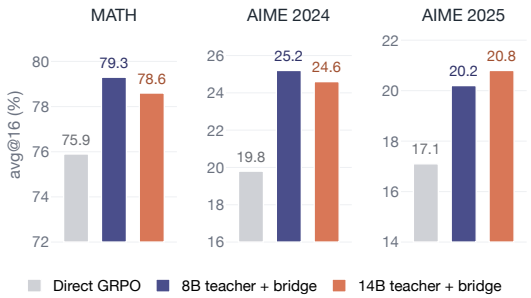

Figure 1: Headline contrast on the same Qwen3-1.7B deployment student (avg@16, %; details in Table 6). Each panel uses its own zoomed accuracy axis so the three runs stay visually comparable across MATH and AIME. Allocating the same labeled training data to teacher RL plus the two-stage bridge outperforms direct student GRPO at every benchmark.

What this changes in practice. The standard post-training pipeline—SFT, then RL on the deployment model—places the scarce labeled data in the least effective position first. The teacher-first view prescribes a different order: allocate the labeled training data to a model large enough to use it, run a two-stage dense bridge into the deployment model, and only then decide whether any held-out labeled data remains worth using on the student. Figure 2 summarizes the resulting pipeline.

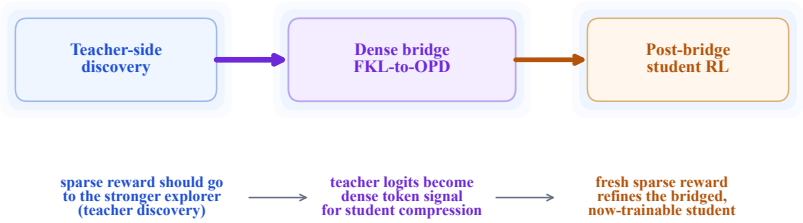

Figure 2: Where labeled training data should be allocated. The teacher-side path (Stage 1: teacher RL) discovers reward-shaped behavior; the two-stage dense bridge (Stage 2a: FKL warmup, Stage 2b: OPD) converts it into token-level supervision for the deployment student; the optional post-bridge student RL stage (Stage 3) uses any remaining labeled data on a now-trainable student.

Contributions. We evaluate the reward-density principle on verifiable math and make three contributions:

1. **Teacher-first allocation** (Section 4). At fixed deployment-student size, a fixed pool of labeled training data yields a stronger student when it is allocated to teacher RL plus dense transfer than when it is allocated to direct student RL. The gain requires a reward-shaped teacher: transferring the same base teacher before teacher-side RL underperforms direct GRPO, so scale alone is not the cause.
2. **A two-stage dense bridge** (Sections 3 and 4.2). In the pre-Stage-3 transfer comparison, a forward-KL warmup on teacher rollouts followed by OPD on student rollouts outperforms both teacher-sample SFT and OPD-only transfer. The warmup reduces the cold student's coverage mismatch so that the subsequent OPD stage—which is a local trust-region update under a dense teacher-induced implicit reward—is better conditioned.

3. **Post-bridge student RL** (Section 4). The bridge changes student trainability: sparse-reward GRPO that is weak on a cold student lifts the bridge endpoint above both direct GRPO and a matched replay control that reuses bridge data.

Scope. The evidence is on verifiable math (MATH-500, AIME 2024, AIME 2025) with two student-teacher families: Qwen3-family models [Yang et al., 2025] as the main study, and Llama-family models [Grattafiori et al., 2024] as a cross-family replication (Section 4.3). The Qwen deployment student is Qwen3-1.7B, paired with Qwen3-8B and Qwen3-14B teachers in two variants each—off-the-shelf ("raw") and RL-trained. The Llama student is Llama-3.1-8B-Instruct with a Llama-3.3-70B-Instruct teacher. On-policy distillation requires a shared tokenizer; we therefore run the recipe separately within each family. An additional non-RL teacher control (SFT-trained teacher) is reported in Appendix B.

Terminology. A *sparse reward* is a sequence-level task reward $R(x, y)$ available only at the end of a trajectory. A *reward-shaped target* π_R^* is the KL-regularized policy induced by that reward. A *dense teacher signal* is the token-level teacher log-probability $r_T(s_t, y_t) = \beta \log \pi_T(y_t \mid s_t)$ supplied by a teacher that approximates π_R^*. *OPD* is reverse-KL distillation on student rollouts. The *two-stage bridge* (or FKL-to-OPD) is forward-KL on teacher rollouts followed by OPD on student rollouts.

2 The Workflow

We now spell out the operational route implied by the allocation principle. The workflow takes a fixed pool of labeled training data $\mathcal{D}$, a deployment student π_θ, and a larger teacher π_T that shares the student's tokenizer. It produces a post-trained student through four stages; the theoretical justification is deferred to Section 3.

Stage 1: Teacher-side sparse-reward RL. Run GRPO (or a comparable verifier-based RL algorithm) on the teacher using $\mathcal{D}$. This produces a reward-shaped teacher π_T whose distribution concentrates on high-reward trajectories. Standard recipes either skip this stage or replace it with supervised fine-tuning on $\mathcal{D}$; we argue in Section 3 that an RL-shaped teacher is qualitatively different from an SFT-shaped one as a source of dense supervision.

Stage 2a: Forward-KL warmup on teacher rollouts. Sample K rollouts from π_T on prompts in $\mathcal{D}$ and train the student to match the teacher's next-token distribution on those rollouts:

$$\mathcal{L}_{\mathrm{F}}(\theta) = \mathbb{E}_{s \sim d_{\pi_T}} \, \mathrm{KL}(\pi_T(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)) \, .$$

This is supervised next-token training under teacher occupancy. It is mode-covering, off-policy with respect to the student, and well-conditioned at cold start because it trains directly on teacher-supported states.

Stage 2b: On-policy distillation under student rollouts. Sample rollouts from the current student π_θ and minimize the reverse-KL to the teacher on those rollouts:

$$\mathcal{L}_{\mathrm{R}}(\theta) = \mathbb{E}_{s \sim d_{\pi_\theta}} \, \mathrm{KL}(\pi_\theta(\cdot \mid s) \,\|\, \pi_T(\cdot \mid s)) \, .$$

The teacher's log-probabilities are queried on-policy at the student's sampled prefixes; the teacher itself is frozen. This stage corrects the student on its own state distribution.

Stage 3 (optional): Post-bridge student-side sparse-reward RL. If any labeled data remains held out from Stages 1–2, run GRPO on the bridged student using that held-out data. Section 4 shows this stage adds value over (a) skipping it and (b) reusing already-seen bridge data for more student updates.

The full workflow at a glance

Stage 1. Sparse-reward RL on teacher, using labeled data $\mathcal{D}$.
Stage 2a. Forward-KL distillation on teacher rollouts.
Stage 2b. On-policy reverse-KL distillation on student rollouts.
Stage 3. (Optional.) Sparse-reward RL on student, using held-out labeled data.

Stages 1–2 are the core bridge; Stage 3 applies only when labeled data has been split between teacher and student.

What standard recipes drop. The SFT-then-RL recipe on the deployment model corresponds to "replace Stage 1 with SFT on $\mathcal{D}$, drop Stage 2 entirely, run Stage 3 on the full $\mathcal{D}$." DeepSeek-R1-style teacher distillation [Guo et al., 2025] corresponds to "run Stage 1, keep only the off-policy half of Stage 2 (teacher-sample SFT), drop Stage 2b and Stage 3." Standalone on-policy distillation [Agarwal et al., 2024, Lu and Thinking Machines Lab, 2025] corresponds to "run Stage 2b alone, dropping Stage 2a." The component ablation in Section 4 measures each of these omissions.

3 Theory: A Reward-Density Principle

We justify the workflow through a trust-region reading of on-policy distillation that mirrors recent self-distillation analyses [Shenfeld et al., 2026]. The full derivation is in Appendix A; here we keep the load-bearing equations and arguments.

Reward-shaped target. Let x be a prompt, $y = (y_1, \ldots, y_T)$ a response, and $s_t = (x, y_{<t})$ the autoregressive state. A sequence-level verifier reward $R(x, y)$ induces the KL-regularized target

$$\pi_R^*(y \mid x) = \tfrac{1}{Z_R(x)} \pi_{\mathrm{ref}}(y \mid x) \exp(R(x, y)/\beta), \tag{1}$$

the optimum of $\mathbb{E}_{y\sim\pi}[R(x, y)] - \beta\,\mathrm{KL}(\pi\|\pi_{\mathrm{ref}})$ [Rafailov et al., 2023, Shenfeld et al., 2026]. Direct student RL tries to recover π_R^* from the student's own sparse rollouts. The workflow instead uses sparse reward on a larger model and treats the resulting teacher as a proxy

$$\pi_T(\cdot \mid x) \approx \pi_R^*(\cdot \mid x). \tag{2}$$

OPD as a local implicit-reward update. Let $\pi_k = \pi_{\theta_k}$ be the current student. The negative reverse-KL gradient at π_k equals the policy gradient of a dense per-token implicit reward $\widetilde{R}_T^k$:

$$-\beta\,\nabla_\theta\,\mathrm{KL}(\pi_\theta\|\pi_T)\big|_{\theta_k} = \mathbb{E}_{y\sim\pi_k}\Big[\widetilde{R}_T^k(x, y)\,\nabla_\theta \log \pi_\theta(y \mid x)\Big]_{\theta_k}, \quad \widetilde{R}_T^k(x, y) = \sum_{t=1}^{T} \beta \log \tfrac{\pi_T(y_t|s_t)}{\pi_k(y_t|s_t)}. \tag{3}$$

Each OPD step is therefore the local trust-region update that maximizes $\mathbb{E}_\pi[\widetilde{R}_T^k] - \beta\,\mathrm{KL}(\pi\|\pi_k)$. The signal is a dense, on-policy, teacher-induced surrogate for the sparse verifier; its value depends on two conditions on the proxy.

(C1) Optimality. The teacher must achieve near-maximal reward on the verifier: $\mathbb{E}_{y\sim\pi_T}[R] \approx \mathbb{E}_{y\sim\pi_R^*}[R]$. If π_T has not been shaped by sparse reward, then the implicit reward in Eq. 3 pushes the student toward an unshaped distribution—larger than the student, but not reward-aware. *Stage 1 is the device that enforces C1.*

(C2) Minimal deviation. The teacher must lie within a small KL of the student in the relevant state distribution: $\mathrm{KL}(\pi_T\|\pi_\theta)$ small at the current π_k. When teacher and student have little coverage overlap—e.g., a cold 1.7B student and a post-RL 8B teacher—the per-token implicit reward $\beta \log(\pi_T/\pi_k)$ has high variance: it takes large magnitudes on student-sampled tokens the teacher considers unlikely, while teacher-favored tokens are rarely sampled. The OPD gradient is then dominated by a few outlier terms and updates are unstable [Li et al., 2026, Hou et al., 2026]. The forward-KL warmup is supervised next-token training under teacher occupancy: it moves the student onto teacher-supported tokens without student-side discovery, so that the post-warmup anchor π_k is closer to π_T and (C2) is more plausible. *Stage 2a is the device that enforces C2.* Shenfeld et al. [2026] satisfy (C2) by construction because their teacher is the same model conditioned on a demonstration; our cross-scale setting must construct it explicitly.

Three falsifiable predictions. The reward-density principle yields three predictions for the experiments in Section 4.

- **(P1, C1):** If Stage 1 is removed—a raw teacher in its place—the bridge endpoint should not exceed direct student GRPO, regardless of teacher scale.

Table 1: Direct GRPO across Qwen3 scales on MATH-500, AIME 2024, AIME 2025 (avg@16, %). The Qwen3-1.7B row is the deployment-student baseline that the workflow must beat.

Model	MATH	AIME 2024	AIME 2025
Qwen3-1.7B	75.9 ± 0.9	19.8 ± 1.4	17.1 ± 0.9
Qwen3-8B	88.4 ± 0.8	47.7 ± 1.5	36.7 ± 1.2
Qwen3-14B	89.5 ± 0.7	47.1 ± 1.2	39.0 ± 0.9

- **(P2, C2):** If Stage 2a is removed—OPD without the warmup—the endpoint should fall below the full bridge, because Eq. 3 is ill-conditioned at cold start. A purely off-policy bridge (no Stage 2b) should also fall below, because the student never receives feedback on its own states.
- **(P3, post-bridge trainability):** Once Stages 1–2 place the student in a useful neighborhood of π_T, fresh labeled examples in Stage 3 should produce sparse-reward gradients beyond what extra updates on bridge data can extract.

4 Experiments

We test the three predictions through one baseline table (Section 4.1) and one component-ablation table (Section 4.2) on the Qwen3 family—Qwen3-1.7B as the deployment student, with raw and RL-trained Qwen3-8B and Qwen3-14B as teachers—then replicate the central teacher-quality contrast in the Llama family (Section 4.3). The Qwen experiments follow the workflow order: allocation first, bridge second, and post-bridge student RL third, but fold these questions into a single ablation table. Accuracies are avg@16 (each problem is scored by mean correctness over 16 independent samples), with $\pm$ standard error across evaluation problems. The training stack builds on verl/HybridFlow [Sheng et al., 2024]; full hyperparameters and data splits are in Appendix F.

4.1 Direct GRPO baseline across scales

Table 1 establishes that, across Qwen3 models, larger models reach substantially higher endpoints under the same algorithm. The 1.7B's lower endpoint is not evidence of a broken optimizer; it reflects two compounding factors: the model's smaller intrinsic capacity, and the known inefficiency of sparse-reward RL on a policy with near-zero base pass rate on hard problems. The workflow targets only the second factor—the first is a hard ceiling no post-training procedure can move—and the recoverable portion is what subsequent sections quantify.

4.2 Component ablation: each stage of the workflow is load-bearing

Table 2 is the central experimental result. It asks three questions in order: whether sparse reward is better placed on the teacher than on the student, whether the bridge needs both the forward-KL warmup and OPD, and when student-side RL becomes useful again. Each row removes one stage of the workflow while holding all other stages and the labeled-data pool fixed. The top block isolates Stages 1, 2a, and 2b in the full-DAPO setting without Stage 3. The bottom block isolates Stage 3 in the half-split setting, where the first half of DAPO is used for the bridge and the second half is held out for Stage 3 or for the replay control.

The table confirms all three predictions in Section 3.

(P1, Stage 1 enforces C1.) Removing teacher-side RL collapses the deployment student at both teacher sizes. Raw 8B and 14B teachers distilled through the same bridge yield only 71.5% and 72.8% MATH—7.8 and 5.8 points below their respective full-workflow endpoints, and both well below cold GRPO. The dense implicit reward in Eq. 3 carries useful gradient information only when its source distribution has itself been shaped by sparse reward; scale alone is not the cause—a larger raw teacher is if anything worse than direct student RL. A non-RL counterfactual (SFT-trained teacher) sits between the raw and RL-trained endpoints and is reported in Appendix B.

(P2, Stage 2a enforces C2 and Stage 2b uses the local IRL identity.) Removing the forward-KL warmup degrades MATH by 1.7 points at the 8B teacher ($79.3 \rightarrow 77.6$) and by 1.5 points at the 14B teacher ($78.6 \rightarrow 77.1$): the cold student's coverage mismatch with the post-RL teacher makes

Table 2: Component ablation of the workflow at both teacher sizes. Qwen3-1.7B deployment student; avg@16 (%). **Top:** Stages 1/2a/2b, evaluated pre-Stage-3 on full DAPO. **Bottom:** Stage 3, evaluated in the half-split setting (1H for bridge, 2H for Stage 3 or replay). Each row removes one stage of the workflow.

Teacher	Configuration	MATH	AIME 2024	AIME 2025
Stages 1, 2a, 2b (full DAPO; no Stage 3)				
RL'd Qwen3-8B	**Full bridge**: Stage 1 + 2a + 2b	$\mathbf{79.3 \pm 0.7}$	$\mathbf{25.2 \pm 1.6}$	20.2 ± 1.3
	− Stage 1 *(raw teacher)*	71.5 ± 0.9	15.0 ± 1.5	10.6 ± 1.2
	− Stage 2a *(no FKL warmup, OPD only)*	77.6 ± 0.8	23.0 ± 1.4	18.9 ± 1.4
	− Stage 2b *(no OPD, teacher-sample SFT only)*	76.0 ± 0.9	22.4 ± 1.5	19.4 ± 1.4
RL'd Qwen3-14B	**Full bridge**: Stage 1 + 2a + 2b	78.6 ± 0.9	24.6 ± 1.5	$\mathbf{20.8 \pm 1.5}$
	− Stage 1 *(raw teacher)*	72.8 ± 0.8	16.7 ± 1.4	13.5 ± 1.3
	− Stage 2a *(no FKL warmup, OPD only)*	77.1 ± 1.0	22.8 ± 1.5	18.6 ± 1.7
	− Stage 2b *(no OPD, teacher-sample SFT only)*	76.5 ± 1.1	21.5 ± 1.5	17.0 ± 1.1
—	− Entire pipeline *(cold GRPO baseline)*	75.9 ± 0.9	19.8 ± 1.4	17.1 ± 0.9
Stage 3 (half-split: 1H bridge, 2H held out)				
RL'd Qwen3-8B	**Full workflow**: bridge (1H) + Stage 3 (2H)	$\mathbf{78.5 \pm 0.9}$	$\mathbf{23.7 \pm 1.5}$	18.5 ± 1.2
	− Stage 3 *(bridge only on 1H)*	75.4 ± 0.8	22.0 ± 1.6	16.7 ± 1.4
	∘ Replay control *(Stage 3 reuses 1H data)*	75.7 ± 0.7	21.6 ± 1.3	17.0 ± 1.2
RL'd Qwen3-14B	**Full workflow**: bridge (1H) + Stage 3 (2H)	$\mathbf{78.7 \pm 1.1}$	23.1 ± 1.7	$\mathbf{19.2 \pm 1.3}$
	− Stage 3 *(bridge only on 1H)*	76.3 ± 1.1	22.7 ± 1.7	17.3 ± 1.2
	∘ Replay control *(Stage 3 reuses 1H data)*	75.6 ± 1.0	22.4 ± 1.5	17.6 ± 1.0

the implicit reward high-variance and the OPD update poorly conditioned. Removing OPD and keeping only the off-policy stage degrades further—to 76.0% and 76.5% respectively—because teacher-sample SFT never gives feedback on student-only states. Both one-stage variants sit below the two-stage bridge at both teacher sizes on MATH and AIME 2024, as the trust-region analysis predicts.

(P3, Stage 3 adds value after the bridge.) Once Stages 1–2 place the student inside the teacher's neighborhood, sparse-reward RL on the held-out half lifts MATH from 75.4% to 78.5% at the 8B teacher and from 76.3% to 78.7% at the 14B teacher ($+3.1$ and $+2.4$ points). The replay controls run the same number of student updates on already-seen bridge data and never improve by more than 0.3 points, so the gains are attributable to new labeled examples rather than to extra updating. Cold direct GRPO without the bridge reaches only 75.9%, confirming that the bridge changes student trainability rather than merely providing a better initialization that subsequent RL can match from scratch.

Held-out-half allocation. A residual allocation question—whether the held-out half of the labeled data is better spent upstream on the teacher or downstream on Stage 3—is examined in Appendix C. The teacher-side route slightly outperforms the student-side route (0.8 MATH points at the 8B teacher), but the gap is much smaller than the stage-ablation gaps in Table 2.

4.3 Cross-family replication on Llama

We test whether the teacher-quality ordering generalizes beyond the Qwen3 family by repeating the central P1 contrast in the Llama family: a Llama-3.1-8B-Instruct deployment student paired with a Llama-3.3-70B-Instruct teacher (Table 3). The same ordering reproduces—raw-teacher transfer $<$ direct GRPO $<$ RL-teacher transfer. A $9\times$ larger raw teacher is still worse than direct GRPO on the student (55.4% vs. 59.8% MATH), confirming that scale alone does not satisfy C1; the same teacher after Stage 1 RL is the best source (62.1%, $+2.3$ over cold GRPO; AIME 2024 14.9% vs. 12.5%). The deployment-student gain is smaller in absolute terms than in the Qwen block because the 8B Llama student has a higher base capacity than the 1.7B Qwen3 student and therefore less recoverable headroom. The full Llama component ablation (Stage 2a, Stage 2b, half-split, replay, SFT-teacher controls) remains future work.

Table 3: Llama cross-family replication. Student = Llama-3.1-8B-Instruct, Teacher = Llama-3.3-70B-Instruct; avg@16 (%). Each row uses the same protocol as the corresponding Qwen3 row in Table 2.

Configuration	MATH	AIME 2024	AIME 2025
Direct GRPO (cold student)	59.8 ± 0.9	12.5 ± 1.2	7.2 ± 1.1
Two-stage bridge ← raw 70B	55.4 ± 0.8	8.8 ± 1.5	3.1 ± 1.2
Two-stage bridge ← RL'd 70B	$\mathbf{62.1 \pm 0.8}$	$\mathbf{14.9 \pm 1.8}$	$\mathbf{9.2 \pm 1.4}$

5 Discussion

What changes operationally. The standard reading of the post-training literature is a menu of competing methods: SFT, RL, distillation. The reward-density principle turns that menu into an allocation problem. Once OPD is viewed as a local implicit-reward update (Eq. 3), the design choice is not only which method to run, but which model should receive which density of signal, and in what order. Direct sparse-reward RL on the deployment model is inefficient placement on both axes: sparse reward is given to the policy least prepared to use it.

Implication for model-family training. The practical recipe is clearest when a lab trains or maintains a model family rather than a single deployment checkpoint. A larger teacher and a smaller deployment student can be pretrained on the same data distribution, preferably with a shared tokenizer, and kept as parallel post-training targets. The reward-density principle then says that labeled post-training data should be allocated preferentially to the larger model first, because it can convert sparse reward into a better reward-shaped distribution. The smaller model should receive that distribution through the dense FKL-to-OPD bridge, with student-side sparse RL reserved for held-out labeled data after the bridge.

Why student-side reward still matters. The post-bridge student-RL result keeps the recipe from becoming a rigid "never train the student" rule. After the bridge, sparse reward on the student gives a real 3.1-point lift on MATH and is strictly better than running more updates on bridge data. The right framing is teacher-first with post-bridge student RL; the weaker framing is either "RL the student" or "never RL the student."

Limitations. The evidence is on verifiable math with two student-teacher families at relatively small deployment scale (1.7B and 8B students, with teachers up to 14B and 70B). Whether the teacher-first advantage persists, grows, or shrinks at larger scales—for example, a 70B student with a 400B+ teacher—remains open. The reward-density argument predicts persistence, but the marginal value of sparse reward on a stronger student may shift the allocation balance. The bridge requires a shared tokenizer between teacher and student. Code, instruction following, and open-ended tasks would need their own verifier-density experiments, and we make no claim about an optimal way to blend sparse and dense rewards beyond the staged version of Figure 2.

6 Related Work

Post-training reshapes LLM behavior through sparse-reward RL and teacher transfer: RLHF uses sparse preference or outcome rewards [Ouyang et al., 2022, Stiennon et al., 2020, Bai et al., 2022], while distillation transfers teacher behavior through dense supervised signals [Hinton et al., 2015]. We position this paper along both axes. Appendix D gives the per-paper detail.

Sparse-reward post-training. PPO, GRPO, and SFT-warmup-then-PPO recipes apply sparse reward directly to the deployment model [Schulman et al., 2017, Shao et al., 2024, Luong et al., 2024]. Verifier-filtered SFT uses the reward only as a data filter [Zelikman et al., 2022, Singh et al., 2024]. Recent work increases reward density through self-distillation [He et al., 2026, Yang et al., 2026] or reference-guided trajectories [Wu et al., 2026a]. These methods differ in how they use reward, but they still train the deployment model on the labeled data. Our workflow uses the same data upstream on a teacher and then densifies it through the bridge.

Distillation and OPD. Knowledge distillation transfers teacher behavior into smaller models [Hinton et al., 2015]; teacher-sample SFT is its off-policy form [Guo et al., 2025]. OPD corrects the student on its own rollouts [Agarwal et al., 2024] and has been framed as dense on-policy teacher-logprob reward [Lu and Thinking Machines Lab, 2025]. Self-distillation work gives a related trust-region/implicit-reward derivation for demonstration-conditioned teachers [Shenfeld et al., 2026]. Related work connects distillation to entropy-regularized or RL-aware objectives [Liu et al., 2025, Zhang et al., 2026b] and extends OPD through KL scheduling, token importance, chain compression, and offline caching [Xu et al., 2026a,b, Sang et al., 2026, Wu et al., 2026b]. Our Eq. 3 uses the same connection prescriptively: the dense implicit reward is only as good as the teacher proxy, so sparse reward should first improve the teacher.

Reasoning teachers and data allocation. DeepSeek-R1 showed that RL-improved models can teach smaller ones via SFT [Guo et al., 2025]; MiMo-V2-Flash extends this with multi-teacher OPD that integrates domain specialists through on-policy token-level rewards [Xiaomi LLM-Core Team, 2026]. Our focus is different: not whether an RL-improved model can teach, but where a fixed pool of labeled training data should be allocated—teacher-side or student-side—and what bridge connects the two. Appendix E classifies representative methods along this axis.

7 Conclusion

We presented a four-stage post-training workflow—teacher RL, forward-KL warmup, on-policy distillation, optional post-bridge student RL—that improves a Qwen3-1.7B deployment student from 75.9% to 79.3% on MATH and from 19.8% to 25.2% on AIME 2024 at fixed labeled-data budget. The workflow is justified by a reward-density principle: each on-policy distillation step is a local trust-region update under a dense teacher-induced implicit reward, informative only when the teacher is reward-shaped (enforced by Stage 1) and lies within a trust region of the student (enforced by Stage 2a). A single component-ablation table confirms that each stage is load-bearing: removing any one stage costs 1.7–7.8 MATH points. The broader lesson is not to avoid student RL, but to apply it after dense transfer has made the deployment policy trainable.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Zhiqi Bai, Ken Deng, Jinyang Guo, Congnan Liu, Jiaheng Liu, Jie Liu, Lin Qu, Haoran Que, Wenbo Su, Jiakai Wang, Jiamang Wang, Yanan Wu, Chenchen Zhang, Ge Zhang, Yuanxing Zhang, and Bo Zheng. DDK: Distilling domain knowledge for efficient large language models. In *Advances in Neural Information Processing Systems*, 2024.

Junfeng Fang, Zhepei Hong, Mao Zheng, Mingyang Song, Gengsheng Li, Houcheng Jiang, Dan Zhang, Haiyun Guo, Xiang Wang, and Tat-Seng Chua. Rubric-based on-policy distillation. *arXiv preprint arXiv:2605.07396*, 2026a.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, and Feng Zhao. Flow-OPD: On-policy distillation for flow matching models. *arXiv preprint arXiv:2605.08063*, 2026b.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning. *arXiv preprint arXiv:2301.12726*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, et al. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026.

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, et al. Uni-OPD: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

Hyunseok Lee, Soheil Abbasloo, Jihoon Tack, and Jinwoo Shin. Beyond correctness: Learning robust reasoning via transfer. *arXiv preprint arXiv:2602.08489*, 2026.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xin Xie. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*, 2022.

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

Kun Liang, Clive Bai, Xin Xu, Chenming Tang, Sanwoo Lee, Weijie Liu, Saiyong Yang, and Yunfang Wu. ORBIT: On-policy exploration-exploitation for controllable multi-budget reasoning. *arXiv preprint arXiv:2601.08310*, 2026.

Guanlin Liu, Anand Ramachandran, Tanmay Gangwani, Yan Fu, and Abhinav Sethy. Knowledge distillation with training wheels. *arXiv preprint arXiv:2502.17717*, 2025.

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

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. ReFT: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

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

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

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

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents. *arXiv preprint arXiv:2604.10674*, 2026a.

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. TCOD: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents. *arXiv preprint arXiv:2604.24005*, 2026b.

Yangzhen Wu, Shanda Li, Zixin Wen, Xin Zhou, Ameet Talwalkar, Yiming Yang, Wenhao Huang, and Tianle Cai. Learn hard problems during RL with reference guided fine-tuning. *arXiv preprint arXiv:2603.01223*, 2026a.

Yecheng Wu, Song Han, and Hai Cai. Lightning OPD: Efficient post-training for large reasoning models with offline on-policy distillation. *arXiv preprint arXiv:2604.13010*, 2026b.

Xiaomi LLM-Core Team. MiMo-V2-Flash technical report, 2026. URL `https://arxiv.org/abs/2601.02780`.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. PACED: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*, 2026a.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026b.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengyuan Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, 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, et al. Self-distilled RLVR. *arXiv preprint arXiv:2604.03128*, 2026.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025.

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

Qiying Yu, Zheng Sun, Xiang Shen, Liang Gao, Ziyi Pan, et al. DAPO: An open-source llm reinforcement learning system. *arXiv preprint arXiv:2503.14476*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35, 2022.

Jiaxin Zhang, Xiangyu Peng, Qinglin Chen, Qinyuan Ye, Caiming Xiong, and Chien-Sheng Wu. The illusion of certainty: Decoupling capability and calibration in on-policy distillation. *arXiv preprint arXiv:2604.16830*, 2026a.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for LLM reasoning. *arXiv preprint arXiv:2602.22495*, 2026b.

A OPD as a Local Implicit-Reward Update

This appendix expands Eq. 3 of the main text and the surrounding trust-region argument.

Local trust-region identity. For a fixed prompt x, reverse-KL OPD minimizes

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathrm{KL}(\pi_\theta \| \pi_T) = \mathbb{E}_{y\sim\pi_\theta}\left[\log \pi_\theta(y \mid x) - \log \pi_T(y \mid x)\right]. \tag{4}$$

Let $\pi_k = \pi_{\theta_k}$ be the current student, and define the per-step fixed implicit reward

$$\widetilde{R}_T^k(x, y) = \beta\left[\log \pi_T(y \mid x) - \log \pi_k(y \mid x)\right]. \tag{5}$$

The policy-gradient update that maximizes this reward at θ_k has gradient

$$\nabla_\theta \mathbb{E}_{y\sim\pi_\theta}[\widetilde{R}_T^k(x, y)]\big|_{\theta_k} = \mathbb{E}_{y\sim\pi_k}\left[\widetilde{R}_T^k(x, y)\nabla_\theta \log \pi_\theta(y \mid x)\right]_{\theta_k}. \tag{6}$$

The negative reverse-KL gradient gives the same expression:

$$-\beta\nabla_\theta \mathcal{L}_{\mathrm{OPD}}(\theta)\big|_{\theta_k} = \beta\mathbb{E}_{y\sim\pi_k}\left[\left(\log \pi_T(y \mid x) - \log \pi_k(y \mid x)\right)\nabla_\theta \log \pi_\theta(y \mid x)\right]_{\theta_k}, \tag{7}$$

where the $+1$ term from differentiating $\log \pi_\theta$ has zero expectation against the score function. Using the autoregressive factorization $\log \pi(y \mid x) = \sum_t \log \pi(y_t \mid s_t)$, the sequence reward decomposes into

$$\widetilde{R}_T^k(x, y) = \sum_t \beta \log \frac{\pi_T(y_t \mid s_t)}{\pi_k(y_t \mid s_t)}. \tag{8}$$

Thus each OPD gradient step can be interpreted locally as policy-gradient learning with a dense teacher-student likelihood-ratio reward. This is a local equivalence at π_k, not a claim that OPD globally optimizes the original sparse task reward.

Two levels of abstraction. Eq. 1 defines a reward-shaped fixed point π_R^* from a fixed base π_{ref}; we use it as bookkeeping for what teacher-side sparse RL approximates. Locally, each OPD step is the per-step trust-region IRL update above, anchored at the current π_k with π_T playing the role of the one-step optimum. The local identity holds at any θ_k regardless of which global π_{ref} was used to define π_R^*—it is purely a fact about the linearization at π_k.

Why FKL closes the (C2) gap. Eq. 3 is mathematically valid for any π_T and π_k, but it is *informative* only when (C2) approximately holds at π_k. When teacher and student have little coverage overlap, the implicit reward $\beta \log(\pi_T/\pi_k)$ takes large magnitudes on student-sampled tokens that the teacher considers unlikely, while teacher-favored tokens are rarely sampled and contribute little. The OPD gradient is then dominated by a few outlier terms; updates are unstable. The forward-KL stage is the explicit device that moves the local anchor toward the trust region of π_T: on teacher rollouts, $\mathcal{L}_{\mathrm{F}} = \mathbb{E}_{s\sim d_{\pi_T}} \mathrm{KL}(\pi_T \| \pi_\theta)$ is mode-covering supervised next-token training, off-policy with respect to the student, and well-conditioned at cold start because it trains directly on teacher-supported tokens. FKL does not invoke the local IRL identity itself; its role is to move the anchor. After warmup, π_θ has substantial mass on teacher-supported tokens and the subsequent OPD stage applies the local identity at a post-FKL anchor where (C2) is more plausible. Shenfeld et al. [2026] satisfy (C2) by construction because their teacher is the same model conditioned on a task-specific demonstration; the two-stage bridge is the cross-scale construction that obtains the same trust-region property explicitly.

B Half-Split Experiments: SFT-Teacher and Bridge-Protocol Controls

This appendix gives the SFT-teacher control referenced in Section 4.2 and the bridge-protocol controls within the half-split setting. The half-split construction itself is described in the bottom block of Table 2: the DAPO-Math-17K training set [Yu et al., 2025] is split into two random halves 1H and 2H; Stage 1 and Stage 2 are run on 1H, and Stage 3 is run on either the held-out 2H (full workflow) or on 1H (replay control).

SFT-teacher protocol. The SFT-trained teacher checkpoints used in Tables 4 and 6 are obtained by supervised fine-tuning Qwen3-8B and Qwen3-14B on responses generated by gpt-oss-120B on the DAPO-Math-17K prompts. They serve as a non-RL counterfactual to the RL-trained teachers: same starting checkpoint and same prompt set, but improved through supervised next-token training on a stronger model's traces rather than through sparse-reward RL on the verifier. The intermediate ordering (raw $<$ SFT $<$ RL) in Table 6 confirms C1: supervised teacher improvement helps but does not replace teacher-side sparse-reward shaping.

Table 4: The same Stage 3 pattern holds when the teacher is SFT-trained instead of RL-trained, but with lower MATH and AIME 2025 endpoints, consistent with C1: an unshaped teacher gives a weaker bridge. Qwen3-1.7B student; avg@16 (%).

Teacher	Student stage	MATH	AIME 2024	AIME 2025
SFT'd Qwen3-8B	After two-stage bridge (1H)	74.3 ± 1.0	21.8 ± 1.5	14.5 ± 1.2
	+ GRPO on held-out 2H	77.2 ± 1.0	22.9 ± 1.3	18.4 ± 1.0
	+ GRPO replay on 1H	74.0 ± 0.8	22.1 ± 1.1	14.2 ± 1.0
SFT'd Qwen3-14B	After two-stage bridge (1H)	75.8 ± 0.9	22.0 ± 1.5	15.1 ± 1.4
	+ GRPO on held-out 2H	76.9 ± 0.8	23.2 ± 1.3	18.6 ± 1.1
	+ GRPO replay on 1H	75.6 ± 0.7	22.3 ± 1.2	14.9 ± 1.2

Table 5: Bridge controls under the half-split setting. Each row uses the same RL-trained Qwen3 teacher and the same Stage 3 GRPO data; only the transfer protocol differs. The two-stage bridge remains the strongest pre-Stage-3 starting point for student-side GRPO. Qwen3-1.7B student; avg@16 (%).

Teacher	Transfer protocol	MATH	AIME 2024	AIME 2025
RL'd Qwen3-8B	two-stage bridge	78.5 ± 0.9	23.7 ± 1.5	18.5 ± 1.2
	OPD only	77.8 ± 0.8	22.8 ± 1.2	16.6 ± 1.3
	teacher-sample SFT	77.3 ± 0.8	22.5 ± 1.4	16.9 ± 1.4
RL'd Qwen3-14B	two-stage bridge	78.7 ± 1.1	23.1 ± 1.7	19.2 ± 1.3
	OPD only	77.5 ± 0.9	21.9 ± 1.5	19.5 ± 1.2
	teacher-sample SFT	77.2 ± 1.2	21.6 ± 1.8	19.0 ± 1.4

C Where Should the Held-Out Half Go?

A residual allocation question after Section 4.2: given a fixed labeled-data pool, where should the second half (2H) go—into Stage 1 or into Stage 3?

The *teacher-side* placement uses both 1H and 2H upstream: the full DAPO set trains the teacher and the bridge, with no Stage 3. This is the top block of Table 2; the RL'd 8B teacher endpoint is 79.3% MATH. The *student-side* placement uses only 1H upstream and applies 2H as Stage 3 on the bridged student: this is the bottom block of Table 2 (78.5% MATH). Both use the same total labeled data; only the placement of 2H changes.

The teacher-side placement wins by 0.8 MATH points (AIME points are within standard error). The margin is small relative to the gaps in Table 2: when teacher-side compute is the binding constraint, the student-side route remains a competitive lower-cost alternative. The full transfer-only grid (raw, SFT, and RL'd teachers at 1.7B, 8B, 14B, with one-stage transfer controls) is in Table 6.

Table 6: Transfer-only endpoints at fixed deployment student (Qwen3-1.7B), without Stage 3. Raw and SFT rows test C1 by using the same transfer protocol without teacher-side sparse RL. The 1.7B RL'd-teacher rows are a same-size control that isolates the dense-reward effect from teacher scale.

Teacher checkpoint	Transfer protocol	MATH	AIME 2024	AIME 2025
—	Direct GRPO (cold student)	75.9 ± 0.9	19.8 ± 1.4	17.1 ± 0.9
raw Qwen3-8B	two-stage bridge	71.5 ± 0.9	15.0 ± 1.5	10.6 ± 1.2
raw Qwen3-14B	two-stage bridge	72.8 ± 0.8	16.7 ± 1.4	13.5 ± 1.3
SFT'd Qwen3-8B	two-stage bridge	76.9 ± 0.9	22.1 ± 1.7	17.6 ± 1.4
SFT'd Qwen3-14B	two-stage bridge	77.6 ± 0.8	23.2 ± 1.6	18.4 ± 1.5
RL'd Qwen3-1.7B	two-stage bridge	76.5 ± 0.8	20.6 ± 1.5	17.1 ± 1.4
RL'd Qwen3-8B	two-stage bridge	$\mathbf{79.3 \pm 0.7}$	$\mathbf{25.2 \pm 1.6}$	20.2 ± 1.3
RL'd Qwen3-14B	two-stage bridge	78.6 ± 0.9	24.6 ± 1.5	$\mathbf{20.8 \pm 1.5}$
RL'd Qwen3-1.7B	OPD only	75.2 ± 0.9	19.1 ± 1.5	12.4 ± 1.2
RL'd Qwen3-8B	OPD only	77.6 ± 0.8	23.0 ± 1.4	18.9 ± 1.4
RL'd Qwen3-14B	OPD only	77.1 ± 1.0	22.8 ± 1.5	18.6 ± 1.7
RL'd Qwen3-1.7B	teacher-sample SFT	73.6 ± 0.9	16.7 ± 1.4	11.4 ± 1.0
RL'd Qwen3-8B	teacher-sample SFT	76.0 ± 0.9	22.4 ± 1.5	19.4 ± 1.4
RL'd Qwen3-14B	teacher-sample SFT	76.5 ± 1.1	21.5 ± 1.5	17.0 ± 1.1

D Extended Related Work

This appendix provides the per-paper detail that the shorter related-work section omits.

Sparse-reward post-training. In sparse-reward policy optimization, the reward directly updates the policy through PPO, GRPO, or SFT-warmup-then-PPO recipes such as ReFT [Schulman et al., 2017, Shao et al., 2024, Luong et al., 2024]. Systems work such as verl/HybridFlow makes these RLHF dataflows practical by combining flexible algorithm representation with efficient distributed execution [Sheng et al., 2024]. In verifier-filtered SFT, the reward is a data-construction rule: sample candidate traces, keep correct ones, and then run supervised imitation [Zelikman et al., 2022, Singh et al., 2024, Yang et al., 2024]. DPO and related derivations make explicit the link between reward optimization and KL-regularized policy targets [Rafailov et al., 2023]. Recent RLVR work moves beyond final-answer correctness by training on more informative intermediate reasoning behavior [Lee et al., 2026]. A related line uses self-distillation to convert sparse binary RLVR rewards into dense token-level supervision [He et al., 2026, Yang et al., 2026]. Reference-guided fine-tuning targets the zero-reward hard-problem regime: partial human reference solutions elicit model-generated positive trajectories before DAPO-style RL, raising the density of rewarding samples on problems the base model cannot initially solve [Wu et al., 2026a].

Distillation and OPD. Knowledge distillation transfers behavior from stronger models into smaller models [Hinton et al., 2015]. Reasoning-distillation work shows that intermediate traces can be more useful than final answers alone [Fu et al., 2023, Li et al., 2022, Magister et al., 2023, Hsieh et al., 2023]. Domain-aware distillation methods adapt transfer to domain knowledge and teacher-student capability gaps [Bai et al., 2024]. Teacher-sample SFT is the off-policy form of this idea: imitate teacher-generated traces, including the DeepSeek-R1 distilled models [Guo et al., 2025]. OPD instead corrects the student on its own rollout distribution rather than only on teacher-generated states [Agarwal et al., 2024]; related variants extend this idea to context distillation and black-box teacher access [Ye et al., 2026, 2025]. Rubric-based OPD pushes the black-box direction further by inducing prompt-specific rubrics from teacher-student contrasts and using weighted rubric pass rates as on-policy rewards [Fang et al., 2026a]. Recent practitioner evidence frames OPD as dense on-policy teacher-logprob reward and reports large compute-efficiency gains over sparse RL and extended off-policy distillation [Lu and Thinking Machines Lab, 2025]. Liu et al. [2025] formulate KD as entropy-regularized value optimization with on-policy and off-policy demonstrations, while Zhang et al. [2026b] propose RL-aware distillation through advantage-aware selective imitation during PPO/GRPO-style updates. Further OPD work studies a forward-then-reverse KL schedule [Xu et al., 2026a], analyzes which student-state tokens carry the strongest learning signal [Xu et al., 2026b], applies on-policy self-distillation to compress overlong reasoning chains [Sang et al., 2026],

introduces temporal curricula and skill-conditioned self-distillation for multi-turn agents [Wang et al., 2026b,a], and explores offline OPD through precomputed teacher log-probabilities [Wu et al., 2026b]. Concurrent analyses dissect when OPD succeeds or fails and propose unified recipes across LLM and MLLM settings [Li et al., 2026, Hou et al., 2026], while Flow-OPD adapts OPD-style dense multi-teacher supervision to flow-matching text-to-image alignment [Fang et al., 2026b]. Zhang et al. [2026a] highlight that OPD can systematically miscalibrate confidence even when accuracy improves; for a taxonomy of OPD feedback signals, teacher access regimes, and loss granularity, see Song and Zheng [2026].

Reasoning teachers and data allocation. DeepSeek-R1 showed that large-scale RL can elicit strong reasoning behavior and that smaller models can inherit it through supervised fine-tuning on DeepSeek-R1-generated traces [Guo et al., 2025]. ORBIT studies a different control dimension: it uses multi-stage RL under context-length constraints to discover Pareto-frontier reasoning-effort policies, then fuses those policies by OPD into one controllable model [Liang et al., 2026]. Our allocation question is different: where should scarce labeled training data enter the post-training pipeline? MiMo-V2-Flash makes the OPD connection explicit through Multi-Teacher On-Policy Distillation (MOPD) [Xiaomi LLM-Core Team, 2026]. Its post-training pipeline first runs SFT, then trains domain-specialized teachers through RL or SFT, and finally integrates those teachers by having the student sample from its own on-policy distribution while receiving token-level reverse-KL rewards from the teacher selected for each prompt domain. The formulation is aligned with our reward-density principle: the teacher log-probability ratio becomes a dense per-token advantage. In our taxonomy, MOPD is a scalable multi-teacher OPD mechanism for capability integration, while our paper studies how scarce labeled training data should be allocated before and after such dense transfer.

E Method Classification

Table 7: Representative methods classified by where sparse reward enters and what signal is used for transfer.

Method / reference	Sparse reward use	Transfer signal	Student endpoint
InstructGPT [Ouyang et al., 2022]	preference reward	none	RLHF policy
GRPO / DeepSeekMath [Shao et al., 2024]	answer RL	none	RL policy
ReFT / PPO after SFT [Luong et al., 2024]	answer RL after SFT	none	RL policy
Verifier-filtered SFT [Zelikman et al., 2022, Singh et al., 2024]	verifier as filter	accepted traces	SFT policy
Step-by-step distillation [Fu et al., 2023]	none	rationales	distilled policy
GKD / OPD [Agarwal et al., 2024]	teacher-dependent	teacher logits	OPD policy
DeepSeek-R1 distilled models [Guo et al., 2025]	teacher-side RL	SFT on teacher-generated traces	distilled policy
MiMo-V2-Flash / MOPD [Xiaomi LLM-Core Team, 2026]	specialized teacher RL/SFT plus optional outcome reward	multi-teacher OPD logits	unified post-trained model
This work	teacher RL plus optional post-bridge student RL	FKL-to-OPD bridge	workflow-trained student

F Implementation Details

All route comparisons keep the deployment-student size fixed. In the Qwen3 block, the student is Qwen3-1.7B and the teacher checkpoints are the raw, SFT-trained, and RL-trained Qwen3 checkpoints listed in Tables 1–5. In the Llama block, the student is Llama-3.1-8B-Instruct and the teacher is Llama-3.3-70B-Instruct. OPD is only run within a model family, because the token-level KL in Stage 2b requires a shared tokenizer and vocabulary.

Data splits. The Qwen allocation experiment uses a fixed random split of the DAPO-Math-17K training set [Yu et al., 2025] into two equal halves. The first half (1H) is the teacher-RL and bridge data pool for the half-split rows, and also the replay data pool for the replay control. The second half (2H) is held out from teacher RL and transfer, then used for Stage 3 GRPO in the full workflow. The full-workflow and replay rows therefore start from the same bridge checkpoint and use the same Stage 3 data count and update count; they differ only in whether Stage 3 uses new labeled examples from 2H or replay examples from 1H. The pre-Stage 3 ablation rows of Table 2 instead train teacher RL and the bridge on the full DAPO set. All rows are evaluated on MATH-500, AIME 2024, and AIME 2025, not on either DAPO training half. The SFT-teacher rows in Appendix B use the same first-half/second-half construction, replacing only the source teacher.

Matched training protocol. Direct GRPO, Stage 3 GRPO, and replay GRPO use the same verifier reward, advantage normalization, optimizer family, batch size, rollout count per prompt, length limit, learning-rate schedule, KL settings, and update count within each matched contrast. The full-workflow and replay rows are matched in checkpoint initialization, data count, rollout count, update count, and sequence-length limit. Teacher-sample SFT and OPD-only ablations keep the RL-trained teacher and Stage 3 GRPO fixed but replace the bridge protocol.

Bridge protocol. The two-stage bridge runs a forward-KL warmup on cached teacher rollouts followed by OPD on student rollouts. The forward stage uses cached teacher rollouts and teacher next-token distributions on those rollouts. The OPD stage queries the frozen teacher on the student's sampled prefixes, so the teacher signal is computed on-policy with respect to the current student distribution. Implementation caches may store these logits for audit and replay, but the teacher checkpoint is not updated. Unless otherwise stated, the forward and OPD stages use the same maximum sequence length and tokenizer as the corresponding student/teacher family, and all teacher-logit temperatures and KL coefficients are fixed across rows inside a contrast.

Evaluation and error bars. All reported accuracies are avg@16. For each evaluation problem, the model samples 16 independent completions under the same decoding configuration; the problem score is the mean correctness over those completions, and the table entry is the mean over problems. The reported $\pm$ values are standard errors over evaluation problems, not standard deviations across independently retrained checkpoints. Data-split seeds, decoding seeds, training seeds, rollout counts, learning rates, KL coefficients, OPD temperatures, maximum prompt and response lengths, and exact checkpoint identifiers are recorded with the run configuration for each table row, so the route contrasts can be reproduced without changing non-ablation hyperparameters.

Table 8: Key GRPO training hyperparameters for direct-RL and Stage 3 student-RL runs.

Group	Parameter	Value
Algorithm	Framework	VERL
Algorithm	Estimator	GRPO
Optimizer	Optimizer	AdamW
Optimizer	Learning rate	1×10^{-6}
Update	GRPO epochs	10
Update	Mini-batch size	8
Update	Micro-batch per GPU	1
Length	Max prompt tokens	3072
Length	Max response tokens	16384
Loss	Clip ratio	0.2
Loss	Gradient clip	1.0
KL	KL coefficient	5×10^{-4}
Data	Validation sets	MATH-500, AIME 2024, AIME 2025
Compute	Precision	bfloat16
Compute	GPUs	$2 \times 8\times$ NVIDIA H200
Compute	Rollout engine	sglang
Compute	Tensor parallel size	16

Table 9: Key OPD/transfer-stage hyperparameters.

Group	Parameter	Value
Algorithm	Framework	VERL
Algorithm	Estimator	GRPO-style actor rollout
Optimizer	Optimizer	AdamW
Optimizer	Learning rate	1×10^{-6}
Training setup	Rollouts per prompt	8
Update	Epochs	1
Update	Mini-batch size	64
Update	Micro-batch per GPU	4
Length	Max prompt tokens	2048
Length	Max response tokens	16384
Sampling	Temperature / top-p / top-k	1.0 / 1.0 / -1
Data	Validation sets	MATH-500, AIME 2024, AIME 2025
Compute	Precision	bfloat16
Compute	GPUs	$2 \times 8\times$ NVIDIA H200
Compute	Rollout engine / TP size	sglang / 2