CONSENSUS AS PRIVILEGED CONTEXT FOR LABEL-FREE SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sampling multiple solutions and returning the majority answer is among the most reliable ways to improve the reasoning accuracy of large language models without labels, and a growing family of methods converts this consensus signal into training supervision. However, existing approaches use consensus only in restricted forms: as a filter that selects solutions for fine-tuning, as a preference between answers, or as a scalar reward for reinforcement learning, discarding most of the information that the agreeing solutions contain. We present CANON (**C**onsensus-**AN**chored self-distillati**ON**), a label-free training method that turns consensus into dense, token-level supervision. For each unlabeled prompt, CANON samples multiple solutions, extracts the majority answer, and conditions a frozen snapshot of the model on a solution that reaches it; this consensus-anchored teacher then supervises the model on its own rollouts at every token. Experiments on mathematical and scientific reasoning benchmarks show that CANON improves pass@1 by up to 15 points, outperforming label-free reinforcement learning by 6 points at 15% of its compute and approaching a teacher conditioned on gold solutions. Trained on unlabeled data, it transfers to held-out benchmarks, matching training methods that use gold labels. Analysis attributes most of the improvement to the consensus being distilled into the weights and finds no significant narrowing of coverage. On the mathematics benchmarks, CANON's pass@k advantage over the base model persists at every sampling budget measured, reaching 512 samples per prompt, and its majority vote itself becomes more accurate.

1 INTRODUCTION

Reinforcement learning with verifiable rewards has become the standard recipe for improving the reasoning abilities of large language models (Guo et al., 2025; Shao et al., 2024), but it presupposes gold answers to verify against. Without labels, agreement across a model's own samples can serve as a signal of correctness: sampling several solutions and returning the most frequent answer, known as self-consistency (Wang et al., 2022), reliably outperforms single-pass decoding. Answers supported by many independent samples are intuitively more likely to be correct, so consistency acts as an unsupervised proxy for accuracy (Wang et al., 2022; Prasad et al., 2025). Agreement is available on any unlabeled prompt at the cost of extra samples, and it identifies not only a likely answer but entire solutions that reach it. This paper studies how to turn that signal into supervision.

Existing methods use consensus as a filter that selects solutions for supervised fine-tuning (Huang et al., 2023), as a source of pairwise preferences (Prasad et al., 2025), or, most prominently, as a scalar pseudo-reward for reinforcement learning on unlabeled data (Zuo et al., 2025). In every case, the agreeing solutions are reduced to a selection or a single bit of feedback, extracting little signal per sample. The reinforcement learning variants additionally require hundreds of optimization steps, with fresh generation at every step. In parallel, on-policy self-distillation has emerged as a dense alternative to reinforcement learning in the labeled regime: a model conditioned on label-derived privileged information (e.g., the gold solution or a derivation for a gold answer) serves as a token-level teacher for its own unconditioned policy (Zhao et al., 2026a; Hübotter et al., 2026). The difficulty is that this privileged context depends on the gold label and is therefore unavailable in the unsupervised setting. This raises the question: *can the model's own consensus play the role of this privileged context, enabling dense self-distillation without gold solutions?*

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

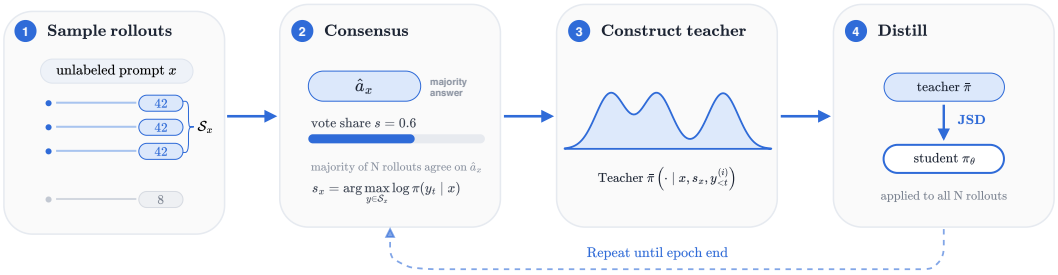


Figure 1: **CANON training framework.** Given an unlabeled prompt, the model samples N solutions and extracts the majority answer. A frozen snapshot of the model, conditioned on a consensus solution, serves as a dense per-token teacher; the student is trained on its own rollouts to match the teacher's distributions, without seeing the consensus solution. Gradients never flow into the teacher. A single generation pass provides both the consensus and the distillation substrate.

We propose CANON (**C**onsensus-**AN**chored self-distillati**ON**), which turns the model's own agreement into a dense training signal. For each unlabeled prompt, CANON samples N solutions, extracts the majority answer, and selects a consensus solution from among the rollouts that reach it. A frozen snapshot of the model, conditioned on this consensus solution, then acts as a teacher: the student minimizes a full-vocabulary Jensen-Shannon divergence toward the teacher's next-token distributions over its own sampled tokens. The rollouts that produce the consensus are reused as the distillation substrate. Thus, CANON needs no generation beyond what self-consistency decoding already performs. Empirically, a single epoch of training suffices (§6).

We evaluate CANON in two regimes. In the *transductive* regime, we train directly on the unlabeled prompts of a test benchmark and evaluate on the same prompts, the setting popularized by test-time reinforcement learning (Zuo et al., 2025); labels are used exclusively for evaluation. In the *inductive* regime, we train on one unlabeled prompt set and evaluate on disjoint prompts, drawn from other benchmarks, from harder problems in the same source, and from a different domain, testing how far the improvements carry beyond the training prompts. Across both, we compare against label-free alternatives, including self-consistency fine-tuning (Huang et al., 2023) and test-time reinforcement learning (Zuo et al., 2025), and against label-full ones: rejection fine-tuning (Yuan et al., 2024), GRPO with gold rewards (Shao et al., 2024), and an oracle variant of our own method whose teacher is conditioned on gold-verified solutions, mirroring the privileged context assumed by gold-conditioned self-distillation (Zhao et al., 2026a; Hübotter et al., 2026). Our evaluation includes competition mathematics and graduate-level multiple-choice science: the method assumes only an extractable final answer and a consensus that is more often correct than individual samples (§5.2).

Our findings are consistent across models and benchmarks. Transductively, CANON is the strongest of the label-free methods we evaluate, improving pass@1 by up to 15 points on competition mathematics and graduate-level science questions while training in about five GPU-hours, whereas the reinforcement learning alternatives require 25 to 40. It also approaches the oracle-conditioned teacher, trailing by half a point on AMC, three on AIME 2024, and eight on GPQA: the model's own consensus captures much of the value of a gold solution as privileged context. Inductively, the improvements transfer to held-out and harder benchmarks, where CANON matches gold-supervised training on the same prompts. Finally, coverage and majority-vote accuracy rise alongside pass@1, concentrated on problems the base model rarely or never solved.

The key contributions of our work are as follows. **(i)** CANON, a label-free post-training method that distills a consensus-anchored teacher into the model over its own rollouts, turning majority votes into dense token-level supervision. **(ii)** A controlled comparison of supervision signals for unlabeled test-time training, showing that CANON is both more accurate and cheaper than the label-free alternatives we evaluate, and recovers most of the gains of gold-conditioned teachers. **(iii)** Evidence that the improvements transfer: gains persist on harder and cross-domain held-out benchmarks, where CANON matches training on gold labels. **(iv)** An analysis of when consensus supervision helps: per-prompt gains track the state of the base model's consensus, largest where the majority is correct but unconfident, near zero where it is saturated, and negative where it is confidently wrong.

2 Related Work

Test-time scaling and test-time training. Sampling multiple solutions and aggregating them by majority vote improves reasoning accuracy without any training (Wang et al., 2022), at the cost of multiplying inference compute on every query. Test-time *training* moves that compute into parameter updates, usually paid per test instance, on data taken from the input via self-supervision (Sun et al., 2020), retrieved for relevance (Hardt & Sun, 2024) or selected to be informative (Hübotter et al., 2025), synthesized from it (Akyürek et al., 2024; Zweiger et al., 2026), or folded into the forward pass (Sun et al., 2025). Training once over the whole unlabeled prompt set amortizes this cost; that is the setting of TTRL (Zuo et al., 2025) and of this paper. CANON also uses self-generated data consisting of the model's own rollouts on unlabeled prompts, while a teacher conditioned on a consensus solution provides token-level supervision.

Consensus as a training signal. Using agreement to exploit unlabeled data predates large language models. Co-training lets two learners supervise each other through their predictions (Blum & Mitchell, 1998; Collins & Singer, 1999). With a single model, agreement among sampled rollouts has served as a filter selecting self-generated solutions for supervised fine-tuning (Huang et al., 2023) and as a preference ranking them for DPO-style training (Prasad et al., 2025), with label-full analogs filtering by gold correctness instead (Zelikman et al., 2022; Singh et al., 2024; Yuan et al., 2024). TTRL (Zuo et al., 2025) turns the majority-vote answer into a pseudo-reward for RL with verifiable rewards, descendants refine or stabilize the pseudo-reward (e.g., Yan et al., 2026; Lin et al., 2026), and related methods replace the vote with intrinsic signals such as semantic entropy, self-certainty, or token-level entropy (Zhang et al., 2025; Zhao et al., 2026b; Agarwal et al., 2025), echoing entropy-based test-time adaptation in vision (Wang et al., 2021). We evaluate TTRL, two of its descendants, and EMPO as baselines, alongside an entropy-minimization control. The vote-based methods reduce agreement to sequence-level targets on selected solutions or to a scalar reward per trajectory; CANON keeps every rollout and supervises it with the teacher's full next-token distributions, obtained from a single generation pass. We show that this difference matters empirically (§6).

Self-distillation with privileged context. Distilling the effect of a context into the weights goes back to context distillation (Snell et al., 2023; Askell et al., 2021), and on-policy distillation trains on the student's own rollouts under a teacher's dense token-level feedback (Agarwal et al., 2024; Song & Zheng, 2026). Recent work instantiates the teacher as the same model conditioned on privileged information, typically the gold solution or rich environment feedback (Zhao et al., 2026a; Hübotter et al., 2026). This can be viewed as generalized distillation (Lopez-Paz et al., 2015), the distillation form of learning using privileged information (Vapnik & Vashist, 2009), with the privileged context assumed given. Closest to us, GATES (Stein et al., 2026) studies label-free self-distillation in document-grounded QA, training off-policy under a live tutor that shares the student's weights, with an external document as the privileged context and consensus among tutor traces gating the signal. CANON instead makes the privileged context itself model-generated, distills on-policy from a frozen snapshot, and needs no document, verifier, or label; our oracle results quantify the cost of substituting consensus for gold solutions.

3 Methodology

3.1 Setting

We are given a policy π_θ, initialized from a pretrained model π_0, and a set of *unlabeled* prompts $\mathcal{D} = \{x_1, \ldots, x_M\}$ with extractable final answers (boxed mathematical answers or multiple-choice letters). No gold answers, reward models, or verifiers are available. We consider two regimes. In the **transductive** regime, $\mathcal{D}$ is the test set itself: we adapt on its prompts and evaluate on the same prompts, with gold labels used only for scoring. In the **inductive** regime, we train on $\mathcal{D}$ and evaluate on disjoint prompt sets, measuring whether label-free training produces improvements that transfer.

3.2 CANON

CANON runs in three steps, illustrated in Figure 1: it extracts a consensus from sampled solutions, constructs a teacher anchored to that consensus, and distills the teacher back into the model on the model's own rollouts.

For each prompt x, we sample N rollouts $y^{(1)}, \ldots, y^{(N)} \sim \pi_0(\cdot \mid x)$ and extract their final answers $a_i = \mathrm{ans}(y^{(i)})$. The majority answer is

$$\hat{a}_x = \arg\max_a \sum_{i=1}^{N} \mathbf{1}\big[a_i = a\big], \tag{1}$$

and the *consensus solution* is the most confident member of the majority set $\mathcal{S}_x = \{y^{(i)} : a_i = \hat{a}_x\}$, namely the rollout with the highest mean token log-probability, $s_x = \arg\max_{y \in \mathcal{S}_x} \frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi_0(y_t \mid x, y_{<t})$. Prompts for which no rollout yields an extractable answer are skipped.

The teacher is a frozen snapshot $\bar{\pi}$ of the policy, taken at the start of the epoch; with a single training epoch, as in our recipe, this snapshot coincides with the initial policy π_0, the same weights that generated the rollouts. The teacher never generates; it only scores. To score a rollout $y^{(i)}$, the consensus solution is inserted into the teacher's context through a short template that presents it as a reference solution for the prompt, the standard construction of privileged-context distillation (Appendix C), and the teacher computes, in a single teacher-forced pass over the fixed rollout tokens, a next-token distribution at every position, $\bar{\pi}(\cdot \mid x, s_x, y^{(i)}_{<t})$. Because the teacher reads a complete solution that reaches the consensus answer before scoring, these distributions are anchored to that line of reasoning. The student computes distributions over the same rollout tokens from the prompt alone and is trained to match the teacher's, by minimizing a full-vocabulary Jensen-Shannon divergence:

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim \mathcal{D}} \, \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|y^{(i)}|} \sum_{t=1}^{|y^{(i)}|} D_{\mathrm{JS}}\Big(\pi_\theta\big(\cdot \mid x,\, y^{(i)}_{<t}\big) \,\|\, \bar{\pi}\big(\cdot \mid x,\, s_x,\, y^{(i)}_{<t}\big)\Big), \tag{2}$$

where $D_{\mathrm{JS}}(p \,\|\, q) = \frac{1}{2} D_{\mathrm{KL}}(p \,\|\, m) + \frac{1}{2} D_{\mathrm{KL}}(q \,\|\, m)$ with $m = \frac{1}{2}(p+q)$; the divergence was selected on a development slice and is discussed in Appendix F. Each rollout contributes its token-mean divergence, so rollouts count equally in the objective regardless of their length. Gradients flow only through the student; the teacher is a stop-gradient snapshot. The N rollouts that formed the consensus double as the distillation substrate, and training runs for a single epoch; additional epochs are studied in §6.

Three properties distinguish CANON from prior uses of consensus. First, the supervision is dense: where a majority-vote reward assigns one bit per trajectory, the consensus-anchored teacher provides a full next-token distribution at every position of every rollout, including rollouts that disagree with the consensus. Second, the teacher is frozen: conditioning a snapshot rather than the live policy prevents the degenerate co-adaptation we observe when teacher and student share parameters (Appendix F). Third, the method is cheap: no reward model, no repeated generation, no preference pairs; the marginal cost over self-consistency decoding is one epoch of distillation. What the teacher can teach is bounded by the consensus. A confidently wrong majority anchors it to a wrong solution and a saturated consensus leaves little to distill (§5).

4 EXPERIMENTS

Our experiments address four research questions. **RQ1** (§4.2): How does CANON compare to existing label-free methods and to a gold-conditioned teacher, in accuracy and in compute? **RQ2** (§4.3): Do the gains hold across models, families, and benchmarks? **RQ3** (§4.4): Do the improvements transfer inductively to held-out and harder benchmarks? **RQ4** (§5): Does CANON sharpen at the expense of coverage, narrowing what the model can solve?

4.1 EXPERIMENTAL SETUP

Most experiments use Qwen3-4B-Instruct-2507; Section 4.3 extends the evaluation to six further models spanning the Qwen family from 2B to 9B, three other model families, and a mixture-of-experts architecture. Hybrid-thinking models are run in non-thinking mode throughout. Small instruct models are the natural setting for this problem: they exhibit reliable consensus on the benchmarks we study while leaving a large gap between single-sample and majority-vote accuracy, which is the headroom CANON targets (§5.2).

Table 1: **Method and supervision comparison** (Qwen3-4B-Instruct-2507, avg@32). All trained arms use the same 83 unlabeled AMC prompts; AMC is transductive, while the remaining columns evaluate the same checkpoints on benchmarks they never trained on. GPU-hours aggregate wall-clock time across the training GPUs and include generation. Best label-free result per column in **bold**; shaded rows use gold labels. Seed variance and significance tests are reported in Appendices H and E.

Method	Labels	GPU-h	AMC (trans.)	AIME24	AIME25	GPQA	AVG
Base model	—	0	66.0	32.5	30.1	43.9	43.1
ScPO	none	4.0	66.4	32.9	30.6	44.3	43.6
LMSI	none	5.6	66.5	32.5	29.3	44.2	43.1
TTRL	none	32.4	70.3	36.7	31.3	45.0	45.8
EMPO	none	25.6	69.6	36.9	31.7	45.0	45.8
SCRL	none	39.2	71.0	37.3	31.4	45.0	46.2
TTRL-Guard	none	37.6	70.5	34.9	**33.2**	45.7	46.1
EM-RL	none	24.8	69.7	37.3	31.4	45.9	46.1
CANON (ours)	none	4.8	**76.5**	**39.5**	32.6	**48.9**	**49.4**
RFT	gold	3.6	70.0	37.0	31.2	44.9	45.8
GRPO (60 steps)	gold	25.6	71.7	38.4	31.8	45.4	46.8
GRPO (180 steps, best)	gold	76.8	79.6	39.7	30.4	47.6	49.3
Oracle distillation	gold	4.8	76.9	40.4	31.6	49.9	49.7

We evaluate on AMC 2023 (83 prompts), AIME 2024 and 2025 (30 each), GPQA-Diamond in multiple-choice format (198) (Rein et al., 2024), and MATH500 (500) (Lightman et al., 2024). For inductive experiments, we additionally construct a pooled unlabeled training set of 339 mixed mathematics prompts (composition in Appendix D) and use OmniMath difficulty slices (Gao et al., 2025). All development choices and hyperparameters are tuned on disjoint OmniMath development slices, with a single CANON configuration used across all models and datasets (Appendix B).

We compare methods matched on training prompts and grouped by supervision. The label-free baselines are LMSI, self-consistency-filtered supervised fine-tuning (Huang et al., 2023); ScPO, self-consistency preference optimization (Prasad et al., 2025); TTRL, reinforcement learning with a majority-vote pseudo-reward (Zuo et al., 2025); two of its descendants, SCRL (Yan et al., 2026) and TTRL-Guard (Lin et al., 2026), which gate and reweight that pseudo-reward; EMPO, which minimizes semantic entropy over answer clusters (Zhang et al., 2025); and EM-RL, whose only reward is low token entropy and which serves as a pure sharpening control (Agarwal et al., 2025). The label-full baselines are RFT, rejection fine-tuning on gold-correct rollouts (Yuan et al., 2024); GRPO, the TTRL pipeline with the reward verified against gold answers (Shao et al., 2024); and an oracle variant of CANON whose teacher is conditioned on a deep-resampled gold-verified solution, so that it differs from CANON only in which solution anchors the teacher. Full descriptions and the deep-resampling protocol are presented in Appendix B.

All evaluations use a single frozen configuration, following TTRL's evaluation protocol: temperature 0.6, top-p 0.95, n=32 samples per prompt with a fixed seed, and one greedy sample. We report avg@32, the mean per-prompt fraction of correct samples (an unbiased estimate of pass@1 at the sampling temperature), maj@32 (majority-vote accuracy), and pass@32. Answers are extracted from `\boxed{}` and checked with math-verify for mathematics, and by letter extraction for multiple choice. Uncertainty for every comparison in the paper, from paired bootstraps over prompts (B=10,000, percentile intervals), is reported in Appendix E.

4.2 TRANSDUCTIVE TEST-TIME TRAINING (RQ1)

Among label-free methods, CANON is the strongest by a wide margin, raising AMC avg@32 from 66.0 to 76.5. Table 1 compares all methods trained on the same 83 unlabeled AMC prompts, evaluated on AMC in the transductive setting and, without further training, on three held-out benchmarks in the inductive one. In the transductive AMC column, the five RL-based alternatives land within 1.4 points of one another and 5 to 7 points below CANON, despite rewards that range from the raw majority vote to consensus-gated labels and pure token entropy, and the two fine-tuning methods barely move the model. What separates CANON is how much of the consensus signal is used to provide supervision, namely a full next-token distribution at every position of every rollout.

Table 2: **Transductive CANON across models and benchmarks** (avg@32; base / CANON, change in parentheses). Each cell trains only on that benchmark's unlabeled prompts, with one fixed recipe per model (Appendix B). †Mixture-of-experts, 1B active parameters.

	AMC	AIME24	AIME25	GPQA	MATH500
Qwen3-4B-Instruct	66.0 / **76.5** (+10.5)	32.5 / **44.6** (+12.1)	30.1 / **35.8** (+5.7)	43.9 / **54.6** (+10.7)	89.3 / **91.2** (+1.8)
Qwen3.5-2B	36.8 / **44.4** (+7.6)	8.2 / **14.7** (+6.5)	11.8 / **14.7** (+2.9)	27.1 / **35.2** (+8.1)	68.8 / **75.3** (+6.6)
Qwen3.5-4B	67.6 / **75.7** (+8.1)	37.4 / **43.4** (+6.0)	26.2 / **33.3** (+7.2)	52.5 / **61.6** (+9.2)	88.6 / **93.9** (+5.3)
Qwen3.5-9B	74.5 / **79.4** (+4.9)	43.1 / **49.4** (+6.2)	34.3 / **37.4** (+3.1)	59.2 / **68.2** (+9.0)	91.9 / **95.2** (+3.3)
SmolLM3-3B	44.4 / **49.0** (+4.7)	7.2 / **11.5** (+4.3)	10.3 / **11.9** (+1.6)	27.8 / **29.5** (+1.7)	72.2 / **77.8** (+5.5)
LFM2.5-8B-A1B†	58.6 / **61.0** (+2.4)	28.8 / **31.9** (+3.1)	23.3 / **23.4** (+0.1)	23.0 / **30.6** (+7.5)	**83.1** / 80.0 (−3.1)
Gemma-4-E4B-it	63.7 / **72.1** (+8.4)	28.4 / **43.9** (+15.4)	25.3 / **28.3** (+3.0)	45.9 / **54.1** (+8.2)	85.4 / **91.1** (+5.8)

Figure 2 measures the accuracy-compute frontier. Gold-reward reinforcement learning eventually surpasses CANON transductively, but at an order of magnitude more compute, and its extra gains do not transfer. GRPO reaches 79.6 at step 180, requiring 76.8 GPU-hours against 4.8 and gold labels throughout; even so, its remaining advantage of 3.1 points over CANON is not statistically significant under a paired bootstrap (see Appendix E). On the held-out columns the picture inverts: the step-180 GRPO checkpoint transfers no better than CANON on AIME 2024, and worse on AIME 2025 and GPQA.

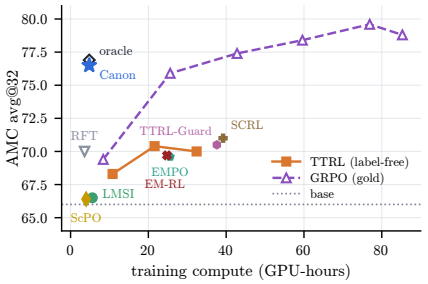


Figure 2: **Transductive AMC, accuracy against compute.** CANON reaches within half a point of the gold-conditioned oracle in 4.8 GPU-hours.

Consensus conditioning recovers most of the value of gold conditioning. The oracle version, identical to CANON except that its teacher is anchored to deep-resampled gold-verified solutions, reaches 76.9 against CANON's 76.5, a difference within evaluation noise. Trained transductively per benchmark, CANON recovers 97% of the oracle's gain on AMC, 78% on AIME 2024, and 58% on GPQA, the one benchmark where gold conditioning keeps a clear advantage (Appendix E).

Adaptation also changes what a single sample achieves. After CANON, a single greedy pass on AMC scores 76.7 averaged over three evaluation seeds, up from 67.1 for the base model and approaching the base model's 32-sample majority vote of 83.1. Much of the benefit of ensembling is amortized into the weights: the one-time cost is a voting pass plus a dozen optimizer steps, and every later query needs one sample instead of 32.

4.3 BREADTH ACROSS BENCHMARKS, MODELS, AND FAMILIES (RQ2)

CANON improves 34 of the 35 reported model-benchmark cells. Table 2 applies CANON across five benchmarks and seven models spanning four families and scales from 2B to 9B, with one fixed recipe per model; the four hard benchmarks gain $+5.7$ to $+12.1$ for Qwen3-4B-Instruct-2507 and $+3.1$ to $+9.0$ for Qwen3.5-9B, the largest model. MATH500 shows that saturation is a property of the model-benchmark pair rather than the benchmark: Qwen3-4B-Instruct-2507 already answers 89% of samples correctly and votes at 95%, leaving little room between expressed and consensus accuracy, and gains $+1.8$, while the unsaturated models gain $+3.3$ to $+6.6$ on the same benchmark (§5.2). AIME 2025 shows the smallest hard-benchmark gain for Qwen3-4B-Instruct-2507; its base consensus is also the weakest of the four (43.3 maj@32), again tracking consensus quality.

Evidence from a single model family can be weak on its own. Shao et al. (2026) show that reinforcement learning on Qwen models can produce large mathematical gains even from random or deliberately incorrect rewards, by amplifying reasoning behaviors already acquired during pretraining, that the same signals largely fail on other families, and accordingly recommend validating any proposed training signal across model families and against uninformative-signal controls. Our experiments address both recommendations. Within the Qwen family, the content-free conditioning control of Section 6 plays the role of the uninformative signal and yields no gain, so the improvements require the consensus content. Across families, SmolLM3-3B improves on all five benchmarks ($+1.6$ to $+5.5$), LFM2.5-8B-A1B, a mixture-of-experts model, on four of five, and Gemma-4-E4B-it, from a fourth family, on all five, with the largest single-cell gain in the table on AIME 2024 ($+15.4$ points).

4.4 Inductive Transfer and Self-Improvement (RQ3)

The transductive results showcase CANON's performance on the prompts it trains on. The inductive question is how much the improvement carries beyond them. We therefore train label-free on one prompt set and evaluate on disjoint benchmarks, in two configurations: a single-source configuration, where the AMC-trained checkpoints of Table 1 are evaluated elsewhere, and a pooled configuration, where training uses a larger mixed-mathematics pool.

The held-out columns of Table 1 already contain the single-source result: AMC-trained CANON improves AIME 2024 by $+7.0$ points over base, AIME 2025 by $+2.5$, and GPQA by $+5.0$, the last a transfer from competition mathematics to graduate-level science questions. The AIME 2025 gain is small on its own; on the combined AIME 2024+2025 set (n=60), AMC-trained CANON improves by $+4.7$ points against $+2.7$ for TTRL and $+3.8$ for GRPO-180 (Appendix E). Qwen3.5-4B shows the same pattern, including transfer from easy to hard problems within OmniMath (Appendix G).

Table 3: **Pooled label-free training, held-out evaluation.** Qwen3-4B-Instruct-2507 trained on a 339-prompt unlabeled pool (Appendix D) and evaluated on AIME 2024, which shares no prompts with the pool. Trajectory methods (TTRL, GRPO) report their final checkpoint, since selecting an intermediate checkpoint would itself require labeled held-out evaluation; full trajectories in Figure 3. Shaded rows use gold labels.

Method	Labels	GPU-h	avg	maj
Base model	—	0	32.5	56.7
TTRL-pool	none	138	39.4	63.3
CANON-pool	none	16.4	**41.7**	53.3
GRPO-pool	gold	142.4	39.0	60.0
Oracle-pool	gold	16.4	42.4	56.7

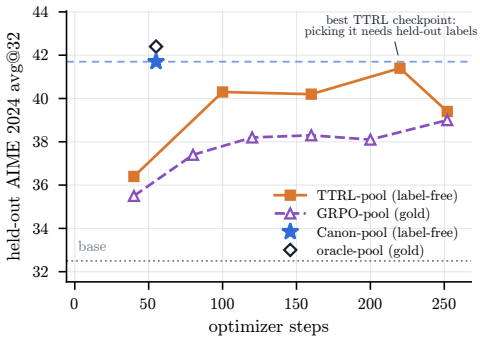


Figure 3: **Held-out AIME 2024 during pooled label-free training.** One CANON-pool epoch is roughly 55 optimizer steps at this batch size; the axis counts the RL methods' steps.

Table 3 scales the source data: all methods train on the 339-prompt unlabeled pool and are evaluated on held-out AIME 2024. CANON reaches 41.7 avg@32 in a single epoch, above the final checkpoints of both TTRL-pool (39.4 after 138 GPU-hours) and gold-reward GRPO-pool (39.0), and within a point of gold-conditioned oracle distillation (42.4). TTRL does pass through a stronger intermediate checkpoint (41.4 at step 220, Figure 3) before regressing, but identifying that checkpoint requires labeled held-out evaluation that the label-free setting does not provide; CANON needs no such selection, since it trains for exactly one epoch. That a label-free method matches gold-supervised training on held-out data mirrors the single-source finding above: dense consensus distillation appears to produce more transferable improvement per unit of supervision than sparse gold rewards, at about 12% of the compute. Two caveats apply: the evaluation set has 30 prompts, so one prompt moves maj@32 by 3.3 points, and CANON-pool's majority-vote accuracy lands below base here while GRPO-pool's does not. Section 5.1 examines consensus-level effects where the statistics are adequate.

Transfer also degrades gracefully with distance from the training distribution. Training Qwen3.5-4B on one unlabeled OmniMath slice and evaluating the same checkpoint at increasing distance yields gains of $+9.2$ points held-in, $+4.9$ on harder problems from the same source, $+0.6$ on a different mathematics benchmark, and $+3.9$ cross-domain on GPQA (Appendix G). Label-free self-improvement is strongest near the training distribution but also transfers to other datasets.

5 Analysis

5.1 Sharpening and Coverage (RQ4)

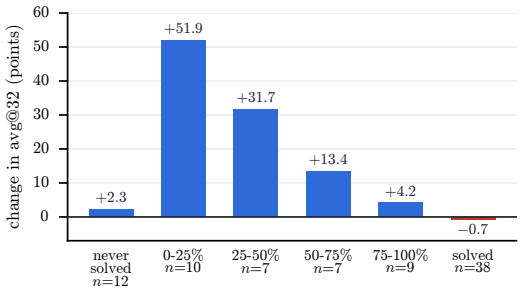
A natural concern with any self-training signal is that it sharpens the output distribution, concentrating probability on answers the model already produces, rather than improving what the model can solve (Huang et al., 2025; Lin et al., 2026). In its simplest form, sharpening makes each prompt's answer distribution peakier without reordering it, so the dominant answer stays dominant and wrong

majorities should not flip to right ones except through sampling noise. Its effect on coverage depends on the reweighting, and token or sequence-level concentration can reorder answer groups, so we measure both empirically. CANON does concentrate the distribution, as expected: mean vote share on AMC rises from 0.68 to 0.81. The EM-RL baseline of Table 1 calibrates what concentration alone buys on the same prompts: rewarding only low token entropy, with no consensus signal, moves AMC avg@32 from 66.0 to 69.7 and leaves maj@32 at the base value of 83.1. The question is whether concentration is all that happens with CANON.

Table 4: **Consensus and coverage effects** of transductive CANON (Qwen3-4B-Instruct-2507). Order-preserving concentration toward existing answers provides no mechanism for flipping a wrong majority to a right one.

	maj@32		pass@32	
	base	CANON	base	CANON
AMC	83.1	**85.5**	85.5	**90.4**
AIME24	56.7	**60.0**	56.7	**66.7**
AIME25	43.3	**46.7**	46.7	**63.3**
GPQA	62.1	**64.7**	**74.2**	73.7
MATH500	**95.4**	95.2	**97.2**	96.8

Figure 4: **Where the gains land.** Change in avg@32 by base-model difficulty band (AMC, transductive). Gains concentrate on the hardest prompts.



Concentration is not all of it. On fresh draws of up to 512 samples per prompt, CANON's pass@k stays above the base model's at every k on the mathematics benchmarks, and the gap persists at every sampling budget measured, from $+5.4$ points at k=32 to $+7.2$ at k=512 on AMC, the opposite of the crossover reported for RL-trained models (Yue et al., 2025) (Appendix J). At that budget CANON solves 12 of the 29 AMC and AIME prompts the base model never solves, and loses none that it solves. At the standard budget, Table 4 shows majority-vote accuracy improving on all four hard benchmarks, by $+2.4$ to $+3.3$ points, and pass@32 improving substantially on the mathematics benchmarks ($+4.9$ on AMC, $+10.0$ on AIME 2024, $+16.7$ on AIME 2025); wrong-to-right majority flips outnumber the reverse on every benchmark, though the per-benchmark counts are small (Appendix J).

Stratifying prompts by base difficulty locates the gains: they are largest in the hardest solved bands and near zero on already-solved prompts (Figure 4). Assigning the bands from one evaluation draw and computing the baseline from an independent second draw changes little, with the hardest solved band still gaining $+39.8$ points (Appendix J). The consensus solution is selected per prompt, but every prompt trains the same weights, so prompts with a correct consensus can improve prompts whose consensus is wrong or missing. This is the transfer of §4.4 inside the training set, and it would explain both the newly solved prompts and the average gain on weakly held wrong majorities (§5.2). MATH500 shows flat coverage because that model-benchmark pair is nearly saturated (§5.2). GPQA is multiple choice, and at large k, pass@k rewards spreading answers across the four options. The base model spreads more and closes the gap by k=128. Majority voting gains nothing from spreading answers, and there CANON stays above the base at every k (Appendix J).

These results argue against sharpening as the complete explanation. They do not establish new capability. Zero successes in a finite draw bound the success probability but do not eliminate it, the entropy baseline does not exhaust sharpening mechanisms, and we do not demonstrate causation in the cross-prompt transfer. What does not depend on the mechanism is the outcome. At every budget we measure, coverage and consensus accuracy rise with the gains instead of being traded for them.

5.2 WHEN DOES CONSENSUS SUPERVISION HELP?

The supervision CANON constructs is only as good as the consensus behind it, which predicts where the gains should concentrate. To test this, we pool per-prompt gains across twelve reported transductive cells of Tables 1 and 2, giving 1,493 prompt-level observations across three models (Qwen3-4B-Instruct-2507, Qwen3.5-4B, and Qwen3.5-9B), and classify each prompt by the state of the base model's consensus: whether the majority answer is correct, and whether it is held confidently (vote share of at least 0.5). The prediction holds. Gains concentrate where the majority is correct but unsure ($+21.5$ points, 239 prompts); they are moderate where it is correct and confident

(+4.3, 875 prompts), small where it is wrong but weakly held (+2.6, 338), and negative where the majority answer is confidently wrong (−3.0, 41), suggesting that training can reinforce these errors. Confidently wrong consensus is also rare, which is why the aggregate effect is strongly positive.

The same pattern holds at a finer grain across vote-share bands (Table 20, Appendix J). Gains are largest through the middle of the range where the consensus is informative but not yet expressed in single samples, moderate at the bottom because it is often too weak to anchor reliably and near zero at the top since there is nothing left to distill. This yields a practical selection rule. CANON is well suited when base consensus accuracy meaningfully exceeds base pass@1, which is exactly the regime of small instruct models on competition mathematics and multiple-choice science, and less likely to help where that gap has closed or the consensus is unreliable.

6 ABLATIONS

The gains of §4 could come from several sources: from the fine-tuning procedure itself, from conditioning on generic solution-shaped text, from conditioning on the consensus answer alone, or from the content of the prompt's own consensus solution. To separate these, we hold the entire recipe fixed and vary only what the teacher is conditioned on, from nothing at all up to the gold solution.

The ladder separates these sources. The content-free placeholder leaves the model at base level (66.6 against 66.0 on AMC), so the fine-tuning procedure alone contributes nothing. Conditioning on the consensus final answer buys 3.5 points, and another prompt's consensus solution, solution-shaped text with the wrong content, buys 3.9. The full consensus solution buys 10.5, within half a point of the gold solution on AMC. The prompt's own consensus solution carries most of the signal, beyond its final answer and beyond solution-shaped text from another prompt.

Table 5: **Conditioning ladder** (Qwen3-4B-Instruct-2507, transductive; avg@32 / maj@32). All arms are identical to CANON except the teacher's context. The shaded oracle row uses deep-resampled gold solutions (Appendix B).

	AMC		GPQA	
Teacher conditioned on	avg	maj	avg	maj
— (base, no training)	66.0	83.1	43.9	62.1
Content-free placeholder	66.6	81.9	44.8	62.1
Consensus final answer only	69.5	84.3	47.2	62.6
Another prompt's consensus	69.9	83.1	47.7	64.1
Consensus solution (CANON)	**76.5**	**85.5**	**54.6**	**64.7**
Gold-verified solution (oracle)	76.9	86.8	62.5	73.7

The remaining design choices, and a set of mechanism arms that vary one component of the supervision at a time, are summarized and detailed in Appendix F (Table 13). Three findings carry the section. First, the teacher must be a frozen snapshot: a live shared-parameter teacher collapses training entirely, with generation entropy and the training loss reaching zero within a handful of steps (Appendix F). Second, the supervision must be dense and must cover all rollouts: sequence-level SFT toward the same consensus solution gains $+1.4$ points where distillation gains $+10.5$, masking out minority-answer rollouts costs 3.5 points, and supervising only the minority costs more than a size-matched random subset (Appendix F). Hard targets and gates discard exactly the low-confidence prompts from which a conditioned teacher's full distribution still extracts signal, which is where CANON's gains concentrate (§5.2). Third, modest sampling and training budgets are sufficient: performance improves monotonically as the rollout count increases up to $N{=}32$ and plateaus at 64, with even $N{=}4$ recovering two-thirds of the gain. Additional epochs with regenerated rollouts add at most a point before performance degrades (Appendix F).

7 CONCLUSION

We showed that a model's own consensus can serve as privileged context for self-distillation, turning the signal behind majority voting into dense, token-level, label-free supervision. Instantiated with a frozen snapshot teacher and a single training epoch, CANON outperforms label-free reinforcement learning and consensus fine-tuning on transductive test-time training at a fraction of their compute, approaches gold-conditioned teachers, and transfers to held-out benchmarks, where it matches training methods that use gold labels. The results suggest that CANON learns from consensus while also improving coverage: majority-vote accuracy and coverage rise alongside pass@1, with the pass@k advantage on mathematics persisting through 512 samples. Limitations of the method and of the evaluation are discussed in Appendix K.

USE OF GENERATIVE AI

AI coding assistants helped implement and refactor the training, evaluation, and analysis code and auxiliary scripts. All AI-written code was reviewed by the authors, and the training and evaluation pipelines were verified end-to-end against reference runs and the released artifacts. AI assistants also checked statistical arguments, derivations, and the consistency of reported numbers; every analysis was specified, run, and verified by the authors. AI assistants helped search for and identify related work; every reference was retrieved and read by the authors before citation, and bibliographic details were checked against the published versions. AI assistants helped draft and edit portions of the manuscript under author direction; all text was reviewed and revised by the authors. The research idea, the method, and the experimental design are the authors' own and were not AI-generated. Beyond the model rollouts that constitute the method under study, generative AI was not used to construct datasets. We have reviewed all AI-assisted content and take full responsibility for the text, claims, and artifacts of this work.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=UfFTBEsLgI.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. *arXiv preprint arXiv:2411.07279*, 2024.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment, 2021. URL https://arxiv.org/abs/2112.00861.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998.

Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999. URL https://aclanthology.org/W99-0613/.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yaqPf0KAlN.

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.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=CNL2bku4ra.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International*

Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=WJaUkwci9o.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1051–1068, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.67. URL https://aclanthology.org/2023.emnlp-main.67/.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. Efficiently learning at test-time: Active fine-tuning of LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NS1G1Uhny3.

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.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Hongxiang Lin, Zhirui Kuai, Erpeng Xue, and Lei Wang. When the majority votes wrong, the intervention timing for test-time reinforcement learning hides in the extinction window. *arXiv preprint arXiv:2605.19444*, 2026.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

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

Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason E Weston, and Jane Yu. Self-consistency preference optimization. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 49737–49751. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/prasad25a.html.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in RLVR. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tqTNOpkP5j.

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

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

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura A Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=lNAyUngGFK. Expert Certification.

Charlie Victor Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, 2023. URL https://openreview.net/forum?id=am22IukDiKf.

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

Alex Stein, Furong Huang, and Tom Goldstein. Gates: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=wXfuOj9C7L.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Dong Yan, Jian Liang, Yanbo Wang, Shuo Lu, Ran He, and Tieniu Tan. What if consensus lies? selective-complementary reinforcement learning at test time. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28957–28970, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1337. URL https://aclanthology.org/2026.acl-long.1337/.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2024. URL https://openreview.net/forum?id=cijO0f8u35.

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

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised LLM reasoning incentivization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=k8Mim6RI5O.

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. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=Jpxfof0EaS.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=OU9nFEYR2M.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=VuVhgEiu20.

Adam Zweiger, Jyo Pari, Han Guo, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. *Advances in Neural Information Processing Systems*, 38:74084–74115, 2026.

A Formal Setup, Objective, and Training

This appendix restates the method of Section 3 with full notation.

Setup. Let $\mathcal{V}$ be the vocabulary and $\mathcal{Y} = \mathcal{V}^*$ the space of token sequences. A policy $\pi_\theta(y \mid x) = \prod_{t=1}^{|y|} \pi_\theta(y_t \mid x, y_{<t})$ is initialized from a pretrained model π_0 and adapted through a low-rank parameterization of θ. We are given unlabeled prompts $\mathcal{D} = \{x_1, \ldots, x_M\}$ and a deterministic answer-extraction map $\mathrm{ans} : \mathcal{Y} \to \mathcal{A} \cup \{\perp\}$, where $\mathcal{A}$ is the answer space (a canonical form for boxed expressions, or a choice letter) and $\perp$ marks sequences without an extractable answer. No reward function, verifier, or gold answer is available to the learner.

Consensus. For each prompt x, draw N rollouts $y^{(1)}, \ldots, y^{(N)} \overset{\text{iid}}{\sim} \pi_0(\cdot \mid x)$ and let $a_i = \mathrm{ans}(y^{(i)})$. The majority answer is the plug-in mode of the induced answer distribution,

$$\hat{a}_x = \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{N} \mathbf{1}\big[a_i = a\big], \tag{3}$$

with size ties resolved to the first-formed group and prompts whose rollouts are all unparseable removed from $\mathcal{D}$. Writing $\mathcal{S}_x = \{y^{(i)} : a_i = \hat{a}_x\}$ for the majority set, the consensus solution is its most confident member under the generating policy,

$$s_x = \arg\max_{y \in \mathcal{S}_x} \frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi_0\big(y_t \mid x, y_{<t}\big). \tag{4}$$

Both selections are deterministic given the rollouts.

Teacher. Let $c(x, s)$ denote the augmented context that embeds a solution s into the prompt for x through the fixed template of Appendix C. The teacher is the conditional next-token distribution of a frozen snapshot $\bar{\pi}$ of the policy under this context, $\bar{\pi}(\cdot \mid c(x, s_x), y_{<t})$. The snapshot is taken at the start of the epoch; in the single-epoch recipe, $\bar{\pi} = \pi_0$. The teacher is never sampled from: it is evaluated by teacher forcing on externally supplied token sequences, and no gradient flows through it.

Objective. Training minimizes the empirical token-mean Jensen-Shannon divergence between student and teacher next-token distributions over the sampled rollouts,

$$\hat{\mathcal{L}}(\theta) = \frac{1}{NM} \sum_{x \in \mathcal{D}} \sum_{i=1}^{N} \frac{1}{|y^{(i)}|} \sum_{t=1}^{|y^{(i)}|} D_{\mathrm{JS}}\Big(\pi_\theta\big(\cdot \mid x,\, y^{(i)}_{<t}\big) \,\|\, \bar{\pi}\big(\cdot \mid c(x, s_x),\, y^{(i)}_{<t}\big)\Big), \tag{5}$$

that is, each rollout contributes its token-mean divergence with equal per-sequence weight, so longer rollouts do not carry proportionally more gradient; $D_{\mathrm{JS}}(p \,\|\, q) = \frac{1}{2} D_{\mathrm{KL}}(p \,\|\, m) + \frac{1}{2} D_{\mathrm{KL}}(q \,\|\, m)$ with $m = \frac{1}{2}(p + q)$, and both arguments are full distributions over $\mathcal{V}$, computed in the implementation as fp32 log-softmax over bf16 logits with no temperature. The divergence is symmetric, finite for any pair of distributions, and bounded by $\log 2$, which makes the per-token loss bounded regardless of how far teacher and student disagree; this contrasts with forward or reverse KL, either of which is unbounded under support mismatch. Treating the teacher as constant, the per-token gradient takes the simple form

$$\nabla_\theta \, D_{\mathrm{JS}}(p_\theta \,\|\, q) = \frac{1}{2} \sum_{v \in \mathcal{V}} \nabla_\theta \, p_\theta(v) \, \log \frac{p_\theta(v)}{m(v)}, \tag{6}$$

which vanishes at the distributional optimum $p_\theta = q$. The log-ratio $\log\left(\frac{p_\theta(v)}{m(v)}\right)$ is bounded above by $\log 2$.

Training. One epoch of stochastic optimization over the fixed set of $\sum_x N$ sequences is performed with AdamW (Loshchilov & Hutter, 2019) under the configuration of Appendix B, with the teacher held fixed throughout. The rollouts serving as the distillation substrate are the same N samples that defined the consensus, drawn from the policy at the start of the epoch; their staleness relative to the evolving student is therefore bounded by a single epoch of optimization. In the multi-epoch generalization, rollouts are regenerated and the teacher re-snapshotted at each epoch boundary, but all results in this paper use one epoch.

B Implementation Details

Training configuration. CANON trains LoRA (Hu et al., 2022) adapters (rank 64, α=128, zero dropout, applied to all linear layers) with AdamW (β_1=0.9, β_2=0.999, weight decay 0.01), a constant learning rate of 10^{-5} with no warmup, gradient clipping at 1.0, and micro-batch size 1, for a single epoch over the $N \times M$ sampled sequences; on an 83-prompt benchmark with N=32 this is 2,656 sequences and roughly 13 optimizer steps. Rollout generation for training uses vLLM at temperature 1.0; the evaluation configuration of Section 4.1 (temperature 0.6) is never used to produce training rollouts. Training runs under FSDP via verl (Sheng et al., 2024) on four-GPU nodes; wall-clock figures are reported for a single reference node with four RTX 6000 Ada GPUs (48 GB). Each model's generation mode is fixed across training and evaluation (Table 6), and answers are extracted from `\boxed{}` with math-verify equivalence checking for mathematics and by letter extraction for multiple choice (Appendix C).

Consensus mechanics. In the consensus computation, vote share is measured over all N rollouts including unparseable ones, and size ties between answer groups resolve to the first-formed group; the selection of the consensus solution by mean token log-probability is deterministic.

Software. Training uses verl 0.9.0.dev0 with vLLM 0.22.1, transformers 5.12.1, torch 2.11.0, and math-verify 0.9.0; TTRL and GRPO instead run unmodified in TTRL's own released stack (verl 0.4.1, vLLM 0.8.4, transformers 4.52.4), so the faithful baseline keeps its reference environment.

Hyperparameter selection. Because transductive experiments offer no validation set, hyperparameter selection must happen elsewhere. We therefore swept the learning rate once per model on a 128-prompt OmniMath development slice (difficulty 2 to 3.5), disjoint from every evaluation benchmark, and froze the selected value for all experiments in the paper. Table 7 reports the sweep; 10^{-5} wins on avg@32 for both models and is used throughout, so no benchmark result in the paper reflects benchmark-specific tuning. The remaining models of Table 2 follow the same protocol, with the learning rate and generation mode selected once per model on dedicated development slices and then frozen; Table 6 lists the selected values.

Table 6: Per-model recipe values (all other recipe components identical across models; everything selected on development slices).

Model	LR	Mode
Qwen3-4B-Instruct-2507	10^{-5}	non-thinking
Qwen3.5-4B / Qwen3.5-2B	10^{-5}	non-thinking
Qwen3.5-9B	10^{-5}	non-thinking
SmolLM3-3B	$3 \cdot 10^{-5}$	non-thinking
LFM2.5-8B-A1B	$5 \cdot 10^{-6}$	non-thinking
Gemma-4-E4B-it	10^{-5}	non-thinking

Table 7: Development learning-rate sweep (OmniMath dev slice, avg@32 / maj@32). The selected value, in bold, is frozen for all reported experiments.

Model	$\mathbf{10^{-5}}$	$5 \cdot 10^{-6}$	$2 \cdot 10^{-6}$
Qwen3.5-4B	**73.4 / 81.3**	71.6 / 82.8	70.3 / 82.8
Qwen3.5-2B	**40.0 / 60.9**	36.9 / 59.4	33.9 / 60.9

Baselines. The baselines follow their reference implementations wherever one exists. TTRL is run faithfully from the released codebase and configuration (learning rate $5 \cdot 10^{-7}$, GRPO advantage estimator, majority-vote pseudo-reward with 64 votes, 32 samples per prompt, no KL penalty), with training and validation both set to the unlabeled target prompts. GRPO is identical except that the reward verifies against gold answers, which makes the TTRL/GRPO pair a controlled comparison of pseudo-reward against gold reward under the same optimizer. LMSI follows the self-consistency-filtered SFT recipe (Huang et al., 2023): 32 rollouts per prompt are sampled at temperature 1.0, prompts whose majority vote share reaches 0.5 are kept (56 of 83 on AMC), and all majority-group rollouts become SFT targets (1,661 sequences), trained for one epoch at learning rate

10^{-5} with the same LoRA configuration; the reported row is a faithful retrain of this configuration whose transductive result matches the original run to 0.1 points, and its measured train pipeline (generation, consensus filtering, and SFT) takes 84 minutes end to end. RFT samples 16 rollouts per prompt at temperature 0.6, keeps prompts with at least one gold-correct rollout (70 of 83 on AMC), and fine-tunes on the shortest correct rollout per prompt (LoRA rank 64, learning rate 10^{-5}, 3 epochs, response-masked loss). ScPO is implemented from the paper, as no official code is released: 32 rollouts per prompt are clustered by final-answer equivalence with the repository's extraction machinery; the chosen response is a random member of the highest-vote cluster and the rejected response a random member of the least-voted cluster; prompts are kept only if the majority vote reaches τN with τ=0.5, and each pair is weighted by the vote margin. The objective is the paper's weighted DPO plus a weighted length-normalized NLL term (β=0.5, α=1, learning rate $5 \cdot 10^{-6}$), with the frozen base model as reference via adapter disabling. Logged deviations from the paper: a single iteration and epoch, LoRA rank 64 rather than full fine-tuning, and a fresh seeded rollout draw. On the 83 AMC prompts the construction yields 12 pairs (44 prompts are unanimous and 27 fail the vote filter), which is itself an observation about consensus-preference methods at test-time scale. Strengthening these baselines beyond their papers' protocols does not change the picture: LMSI run for three epochs with fresh generation and re-filtering each epoch reaches 68.3, and a second ScPO iteration at the paper's iteration-two threshold (τ=0.7) retains 3 pairs and reaches 66.0.

EMPO, SCRL, TTRL-Guard, and EM-RL. All four run from their official releases on the same 83 prompts at the TTRL arm's budget. EMPO minimizes semantic entropy over answer clusters and serves as the confidence-sharpening representative (Zhang et al., 2025); SCRL grants the pseudo-reward only under strict consensus criteria and adds entropy-gated negative pseudo-labels (Yan et al., 2026); TTRL-Guard reweights the pseudo-reward by vote stability (Lin et al., 2026), though its released code reduces to the confidence-reweighting component in this setting, a caveat we do not repair; EM-RL rewards only low token entropy (Agarwal et al., 2025) and runs with the paper's configuration in the TTRL stack (learning rate 10^{-6} with cosine decay, KL coefficient 0.001, full fine-tuning). Table 8 reports their transductive AMC trajectories: all four end between 69.6 and 71.0, inside the TTRL band and well below CANON, and none exceeds the base model's majority-vote accuracy at its endpoint. Their held-out results appear in Table 1. The same paper's two other variants do not help on this cell: EM-FT, which fine-tunes directly on self-sampled rollouts to minimize token entropy, collapses under the TTRL-budget protocol, and EM-INF, inference-time logit adjustment with no parameter update, is within evaluation noise of the base model.

Table 8: Additional label-free RL baselines: transductive AMC trajectories (avg@32 at optimizer steps 20/40/60). Endpoints match Table 1, where held-out results are reported.

Method	step 20	step 40	step 60
EMPO	68.2	70.0	69.6
SCRL	69.1	70.4	71.0
TTRL-Guard	68.6	71.6	70.5
EM-RL	70.1	70.5	69.7

Oracle construction. Oracle distillation is identical to CANON except that the teacher is conditioned on a gold-correct rollout, ranked by the same mean-token-log-probability statistic that selects the consensus solution. On AMC and AIME 2024 the conditioning solutions come from a dedicated deep-resampling pass rather than the training rollouts: up to 256 samples per prompt at temperature 1.0 under one seed, a second 256-sample pass at a second seed for prompts still unsolved, and 512 for the remainder, retaining the most confident gold-correct rollout per prompt. This yields gold conditioning for 75 of 83 AMC prompts, 23 of 30 AIME 2024 prompts, and 179 of 198 GPQA prompts, with per-prompt coverage records; the uncovered prompts (0 correct in 256 to 512 samples) fall back to consensus conditioning, and restricting the oracle comparisons to covered prompts changes no conclusion, since the fallback prompts are ones the base model never solves. The pooled oracle arm uses trainer-internal verification over the N=32 training rollouts instead. The resampling pass is a one-off cost excluded from the oracle's GPU-hours in Table 1.

C Prompt Templates and Answer Extraction

Every rollout, in training and evaluation, is generated from a single-turn chat prompt with no system message: the user message is the problem followed by the standard suffix used by TTRL, wrapped with the tokenizer's chat template (generation prompt appended; thinking disabled for hybrid-thinking models). The student computes its distributions under this same plain prompt.

Rollout and evaluation prompt (user message)

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

The teacher scores the identical rollout tokens under an augmented context: the consensus solution is inserted into a single user message that presents it as a reference solution, followed by the same assistant header, after which the rollout tokens are appended. The full teacher-side user message is the following (long lines wrapped for display; the reference solution sits inside the context, before the assistant turn, and the student never sees it):

Teacher conditioning prompt (user message)

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

A correct solution to this problem is given below for
your reference:
<solution>
{consensus solution}
</solution>

Guided by the reference solution, write your own
step-by-step solution, and put your final answer
within \boxed{}.
```

Final answers are read from the last `\boxed{...}` occurrence in a response, with brace-depth matching for nested braces. For multiple choice, a cascade applies: a boxed letter (allowing `\text` and `\mathrm` wrappers), then an "answer: X" pattern, then a trailing bare letter, searched within the final 400 characters. Two answers are considered equivalent if they match as strings or if math-verify's symbolic check passes in either direction; responses without an extractable answer never match, and count in the vote-share denominator.

D The Pooled Training Set

The pooled experiments of Section 4.4 are meant to test whether label-free training on a broader unlabeled corpus improves a disjoint benchmark, so the pool must be large enough to matter and verifiably disjoint from the evaluation set. The pool contains 339 mathematics prompts: the 83 AMC 2023 prompts taken whole, a 128-prompt subset of MATH500 drawn uniformly at random (seed 1234, the project-wide seed), and 128 OmniMath problems drawn with the same seeded sampler from the difficulty 2.0 to 3.5 band used for development slices. All prompts enter training unlabeled; gold answers for pool prompts are never read by any label-free method, and the label-full arms (GRPO-pool, oracle-pool) use them only as rewards or for teacher-rollout verification, exactly as in the single-source experiments.

The held-out evaluation set, AIME 2024, shares no prompts with the pool; a 13-gram overlap audit of all 339 pool prompts against every held-out evaluation set finds zero matches, with the AMC-transductive component as a positive control (Appendix I). One overlap inside the training side is disclosed for completeness: 43 of the pool's 128 OmniMath prompts also appear in the 128-prompt OmniMath development slice used for learning-rate selection. Both are training-side sets, so no evaluation benchmark is affected and the protocol claim of Section 4.1 is unchanged.

E SIGNIFICANCE TESTS

The benchmark cells in this paper contain between 30 and 198 prompts, so point differences of a few points can be noise. We quantify uncertainty on the central comparisons with a paired bootstrap over prompts (B=10,000), reported in Table 9 as avg@32 differences in points with 95% confidence intervals.

Table 9: Paired bootstrap comparisons (Qwen3-4B-Instruct-2507; B=10,000; avg@32 differences in points). All comparisons use the locked CANON artifact; transductive oracle comparisons use the deep-resampled oracle arms.

Comparison	**Setting**	Δ	**95% CI**	n
CANON − base	AMC transductive	$+10.5$	$[+5.8, +15.6]$	83
CANON − TTRL60	AMC transductive	$+6.2$	$[+2.9, +9.9]$	83
CANON − LMSI	AMC transductive	$+9.9$	$[+5.5, +14.7]$	83
CANON − GRPO60	AMC transductive	$+4.8$	$[+0.8, +8.9]$	83
CANON − RFT	AMC transductive	$+6.6$	$[+1.8, +11.4]$	83
Oracle − CANON	AMC transductive	$+0.4$	$[-2.3, +3.4]$	83
GRPO180 − CANON	AMC transductive	$+3.1$	$[-0.3, +6.9]$	83
CANON − base	AIME24 transductive	$+12.1$	$[+5.3, +19.8]$	30
Oracle − CANON	AIME24 transductive	$+3.4$	$[-1.4, +9.1]$	30
Oracle − CANON	GPQA transductive	$+7.9$	$[+4.7, +11.2]$	198
CANON-AMC − base	AIME24 held-out	$+7.0$	$[+3.3, +11.0]$	30
CANON-AMC − base	AIME25 held-out	$+2.5$	$[-1.9, +6.8]$	30
CANON-AMC − base	AIME24+25 combined	$+4.7$	$[+2.0, +7.7]$	60
CANON-AMC − base	GPQA held-out	$+5.0$	$[+2.8, +7.2]$	198
CANON-AMC − TTRL-AMC	AIME24 held-out	$+2.8$	$[+0.5, +5.7]$	30
Oracle-AMC − CANON-AMC	AIME24 held-out	$+0.9$	$[-1.8, +3.9]$	30
Oracle-AMC − CANON-AMC	AIME25 held-out	-1.0	$[-3.9, +2.0]$	30
Oracle-AMC − CANON-AMC	GPQA held-out	$+1.1$	$[0.0, +2.2]$	198
CANON-pool − base	AIME24 held-out	$+9.2$	$[+2.8, +16.7]$	30
CANON-pool − TTRL-pool	AIME24 held-out	$+2.3$	$[-2.4, +9.2]$	30
CANON-pool − Oracle-pool	AIME24 held-out	-0.7	$[-3.5, +1.9]$	30

Three conclusions follow. CANON's advantage over the base model and over every label-free and matched-compute gold baseline is significant on the transductive AMC cell and on every held-out comparison with 60 or more prompts. Of the six comparisons against base or label-free arms on 30-prompt sets, four are significant; the remaining two (single-source AIME 2025, and CANON-pool against TTRL-pool) have positive point estimates with intervals that include zero, the resolution one expects when a single prompt moves avg@32 by 3.3 points. Against the oracle the picture splits by domain: on the mathematics benchmarks the difference is not distinguishable from zero in any setting tested, while on GPQA the deep-resampled oracle keeps a significant transductive advantage and the held-out interval sits exactly at the boundary. Finally, GRPO-180's remaining transductive advantage, at sixteen times the compute and with gold labels throughout, is itself not significant.

F DESIGN CHOICES AND VARIANCE

The recipe of Section 3.2 fixes four design choices: a frozen snapshot teacher, rollouts generated at the start of the epoch, a single consensus solution per prompt, and no vote-share gating. This appendix justifies each choice and reports seed variance; every experiment varies one choice at a time under an otherwise identical configuration.

Freezing the teacher. Freezing is the one indispensable choice. Table 10 compares the snapshot teacher against a live teacher that shares parameters with the student, on a 390-prompt OmniMath slice: the live-teacher run collapses entirely, and Table 11 shows the collapse signature from the training diagnostics, with loss and gradient norm reaching exactly zero within five steps. Consensus anchoring is only stable against a fixed reference. Rollout freshness matters for the same on-policy reason, though far less dramatically: reusing stale rollouts from an earlier model version loses one to five points against regenerating at epoch start.

Table 10: **Teacher and rollout freshness** (390-prompt OmniMath slice, avg@32; one change at a time).

Variant	avg@32
frozen snapshot (recipe)	72.1
live-policy teacher	0.0
stale rollouts	66.6 to 70.8

Table 11: **Collapse signature** of the live-teacher run (training diagnostics at step 5) against the snapshot run.

	Snapshot	Live teacher
training loss	≈ 0.15	0.00
generation entropy	≈ 0.28	0.04

Consensus width and gating. The remaining two choices are matters of simplicity rather than necessity (values in Table 13). Using three consensus solutions instead of one, implemented as a uniform log-mixture of three consensus-anchored teacher distributions rather than concatenated text, costs about a point on AMC (76.5 to 75.6), so we keep one. Gating training on prompts with vote share of at least 0.5 costs 4.4 points (76.5 to 72.1), consistent with the vote-share-band analysis of Section 5.2: low-confidence consensus still carries usable signal, so discarding it wastes training data.

Rollout count. N controls both the consensus estimate and the amount of distillation substrate. Table 12 varies N with everything else locked: gains grow monotonically to $N{=}32$ and plateau at 64, and even $N{=}4$ recovers roughly two thirds of the full gain, so the method does not depend on a large sampling budget.

Additional rounds. The recipe trains one epoch. Iterating the full procedure, i.e., drawing fresh rollouts and re-anchoring the teacher, buys little: on AMC a second round reaches 77.4 ($+0.9$, at twice the cost) and a third gives the gain back (76.4), and on the pooled configuration a second round declines on held-out AIME 2024 (38.8 against the single-round 41.7). One round captures the achievable gain.

Table 12: **Rollout-count ablation** (transductive AMC; locked recipe otherwise).

N	4	8	16	**32 (recipe)**	64
avg@32	73.0	73.5	75.3	**76.5**	76.1
maj@32	83.1	83.1	84.3	**85.5**	85.5

Table 13: **Design and mechanism ablations** (transductive AMC, avg@32; one change at a time from the locked recipe). Teacher and rollout-freshness ablations and further detail in Appendix F.

Variant	AMC
none (CANON recipe)	76.5
$K{=}3$ consensus solutions	75.6
vote-share gate ($\tau{=}0.5$)	72.1
majority-only supervision	73.0
minority-only supervision	69.4
random subset, minority-matched count	72.2
SFT on the consensus solution	67.4

Mechanism arms. Two arms vary one component of the supervision at a time (Table 13). Restricting supervision to majority-answer rollouts (masking the minority) costs 3.5 points (76.5 to 73.0), so rollouts that disagree with the consensus carry usable signal, consistent with the gate ablation. The converse restriction is worse. Supervising only the minority-answer rollouts reaches 69.4, a deficit that could come from the rollouts or from training on less data. To separate token starvation from rollout identity, a control supervises a deterministic per-prompt random subset of the parsable rollouts, count-matched to the minority set so that data volume cancels from the comparison. It reaches 72.2, so at equal count the minority rollouts are worse supervision targets than random ones by 2.8 points on AMC and by 7.2 points on the OmniMath development slice (65.5 against 72.7), both beyond the training-seed noise of Table 14. Rollouts for these arms are drawn under the recipe's seed but are not identical across arms, since the inference engine is not run-to-run deterministic; a same-seed replicate of the recipe arm lands within 0.9 points of the reported value. Replacing distillation with

sequence-level SFT toward the same consensus solution, selected by the same highest-confidence rule, yields 67.4 against CANON's 76.5: the dense per-token target, not the choice of solution, carries most of the improvement. The divergence itself (JSD against forward KL) was selected on the OmniMath development slice, where JSD won; on the AMC test cell the divergence choice moves the result by under two points (forward KL 78.2, JSD 76.5), single cells at multi-seed noise of 0.3.

Stability. Table 14 reports three independent CANON training runs on AMC with identical configuration; three identical reruns of the AIME 2024 transductive cell give 46.0, 46.2, and 42.6 against the reported 44.6 (mean 44.8, sd 1.7), the expected larger variance of a 30-prompt set. Three seeds of the Qwen3.5-9B AMC cell give mean 79.2, sd 0.7. Evaluation-seed variance is measured separately in Appendix H.

Table 14: **Training-seed variance** (transductive AMC, three independent runs).

	seed 1	seed 2	seed 3	mean (sd)
avg@32	76.4	75.7	75.8	76.0 (0.3)
maj@32	86.8	84.3	85.5	85.5 (1.0)

G ADDITIONAL TRANSFER RESULTS

The main text establishes inductive transfer on Qwen3-4B-Instruct-2507. This appendix extends the evidence to Qwen3.5-4B and to two controlled notions of distance from the training distribution.

Table 15: **Additional inductive transfer on Qwen3.5-4B** (avg@32). Left: AMC-trained and pool-trained checkpoints evaluated on disjoint benchmarks. Right: one checkpoint trained label-free on a 128-prompt OmniMath slice, evaluated at increasing distance from the training distribution.

(a) Cross-benchmark transfer.

Held-out	**Base**	**AMC-tr.**	**Pool-tr.**
AIME 2024	37.4	40.9	39.3
AIME 2025	26.2	28.1	28.9
GPQA	52.5	55.2	55.9

(b) Distance ladder.

Evaluation set	**Base**	**Trained**	Δ
Held-in slice (transductive)	65.0	74.2	$+9.2$
Harder, same source	59.6	64.5	$+4.9$
Different benchmark (AMC)	67.6	68.2	$+0.6$
Cross-domain (GPQA)	52.5	56.4	$+3.9$

Table 15a shows the same qualitative picture as Qwen3-4B-Instruct-2507: label-free training on AMC or on the pool improves every held-out benchmark tested, by two to three and a half points. Majority-vote accuracy on the 30-prompt AIME sets is noisy at this scale; it rises on AIME 2024 (from 60.0 to 63.3 for the AMC-trained checkpoint) and falls on AIME 2025 (from 43.3 to 36.7, a two-prompt movement), so we do not read consensus-level conclusions from these cells. Transfer also crosses difficulty within a single source: training on an easy OmniMath slice (difficulty 2 to 3.5) and evaluating on a separate difficulty 4.5 to 5.5 subset improves avg@32 from 29.6 to 31.3 with maj@32 unchanged.

Table 15b evaluates one checkpoint at increasing distance. Gains decay with distance but remain non-negative everywhere; majority-vote accuracy on the far transfers is noisier, and on AMC it falls from 86.8 to 79.5 while avg@32 still improves. The pattern supports the reading of Section 4.4: label-free self-improvement is strongest near the training distribution and attenuates, rather than inverts, away from it.

H EVALUATION ROBUSTNESS

The benchmark cells in this paper are single evaluations under the frozen configuration, so we quantify the sampling noise a reader needs to interpret point differences.

Table 16 re-evaluates fixed checkpoints under two additional vLLM sampling seeds. The standard deviation over three seeds is 0.2 to 0.8 points on the 83-prompt AMC cells and 0.7 to 1.2 points on the 30-prompt AIME cells; differences of a point or less between arms on a 30-prompt set are therefore within evaluation noise, which is how we read them throughout. Training-seed variance for CANON is of the same order (0.3 points over three independent runs, Appendix F).

Table 16: **Evaluation-seed variance** (avg@32 under three vLLM sampling seeds; fixed checkpoints, frozen configuration otherwise).

Checkpoint	Eval set	1234	5678	9012	sd
base	AMC	66.0	67.2	65.7	0.8
CANON	AMC	76.5	76.9	76.7	0.2
base	AIME 2024	32.5	33.5	34.0	0.8
CANON	AIME 2024	44.6	46.5	44.2	1.2
CANON-pool	AIME 2024	41.7	40.3	41.0	0.7

I CONTAMINATION AUDITS AND A POST-CUTOFF EVALUATION

This appendix addresses the possibility that inductive transfer reflects overlap between the training pool and the held-out benchmarks.

At the prompt level, an exact-string and normalized 13-gram overlap audit between all 339 pool prompts and each held-out evaluation set (AIME 2024, AIME 2025, GPQA) finds zero matches; the audit's positive control, AMC prompts against the pool's AMC component, recovers all 83 with near-duplicate pairs. A sentence-embedding audit (all-MiniLM-L6-v2 (Wang et al., 2020)) over all pool-eval pairs finds no pair with cosine similarity of 0.70 or above (maximum 0.687, median top-1 similarity 0.51).

Finally, we constructed a post-cutoff evaluation set from the 2026 AIME, which post-dates the training data of every model in this paper. The set contains 29 of the 30 problems: every answer was cross-checked against at least two independent sources, and one problem was excluded because its published statement is genuinely disputed. The set has zero 13-gram overlap with the training pool. Table 17 evaluates the transfer checkpoints on it. At n=29, single-sample accuracy is within evaluation noise for every arm; majority-vote accuracy improves under the AMC-trained checkpoint ($+6.9$ points) and the oracle-pool checkpoint ($+6.9$). We report this cell primarily as a contamination control: a benchmark that cannot appear in any training corpus shows the same qualitative consensus-level movement as the pre-cutoff held-out sets.

Table 17: **Post-cutoff evaluation: AIME 2026** (n=29; avg@32 / maj@32). Checkpoints as in Section 4.4.

Arm	avg@32	maj@32
Base model	36.1	51.7
CANON-AMC	37.1	58.6
CANON-pool	35.9	55.2
Oracle-pool	36.9	58.6

J PER-PROMPT ANALYSIS DETAILS

This appendix defines the per-prompt quantities used in Section 5 and reports the statistics that did not fit in the main text. For each prompt we compare base and trained accuracy over the 32 evaluation samples and classify the movement. Coverage transitions count prompts that move between zero and at least one correct sample (cov^+ for gained coverage, cov^- for lost); sharpening transitions count accuracy movement within already-covered prompts (sharp^+ and sharp^-); and majority flips compare the correctness of the per-prompt majority answer before and after training.

Table 18: **Per-prompt transitions** under transductive CANON (Qwen3-4B-Instruct-2507). Counts of prompts per category; the full per-cell table over all reported checkpoints is released with the evaluation artifacts.

						maj flips to	
	cov^+	cov^-	sharp^+	sharp^-	flat	right	wrong
AMC (83)	4	0	26	9	44	3	1
AIME 2024 (30)	4	1	11	2	12	3	2
GPQA (198)	15	16	78	14	75	13	8

Table 18 shows the transition counts on the three transductive benchmarks. On AMC the picture is clean: coverage only grows, sharpening improvements outnumber regressions roughly three to one, and majority flips run three to the right against one to the wrong side. AIME 2024 shows the same asymmetry at its smaller scale. On GPQA the aggregate is similar but individual prompts are noisier, with coverage gains and losses nearly balanced and majority flips at 13 right against 8 wrong; this matches the flat pass@32 and positive maj@32 of Table 4.

A regression-to-the-mean control for the difficulty analysis. The difficulty bands of Figure 4 are assigned from the base model's observed accuracy over 32 samples, and the same finite-sample estimate ordinarily serves as the baseline when computing gains. Prompts that enter the hardest bands partly through an unlucky draw would then show spurious gains under any re-evaluation, and the never-solved band can only move up. To control for this, we separate the roles: bands are assigned from one base evaluation draw, the baseline is computed from an independent second base draw, and the post-training accuracy comes from the trained checkpoint (the two additional evaluation seeds of Appendix H). Table 19 reports both estimates. The regression component is small in every band except one, where it works against the naive estimate, and the aggregate gain is slightly larger under the control ($+11.2$ against $+9.7$ points). On the never-solved band, the independent draw also scores zero. Band membership and magnitudes differ from Figure 4 because the draws differ.

Table 19: **Regression-to-the-mean control** (AMC, transductive CANON, avg@32 in percent). Bands assigned from base draw A; Δnaive uses draw A as the baseline, Δclean uses the independent base draw B.

Band (by draw A)	n	**base A**	**base B**	**CANON**	**Δnaive**	**Δclean**
never solved	10	0.0	0.0	2.5	$+2.5$	$+2.5$
(0, 0.25]	11	11.6	11.4	51.1	$+39.5$	$+39.8$
(0.25, 0.5]	7	40.2	40.6	70.1	$+29.9$	$+29.5$
(0.5, 0.75]	7	62.9	50.9	85.3	$+22.3$	$+34.4$
(0.75, 1]	48	98.6	97.7	98.1	-0.5	$+0.5$
all prompts	83	67.2	65.7	76.9	$+9.7$	$+11.2$

Table 20: **Gains by base vote-share band**: change in avg@32 after transductive training, in fixed-width bins of the base model's vote share over its 32 evaluation samples, pooled over the twelve reported transductive cells.

Base vote share	0 to 0.2	0.2 to 0.4	0.4 to 0.6	0.6 to 0.8	0.8 to 1
Δavg@32 (points)	$+8.0$	$+17.6$	$+16.0$	$+15.6$	$+1.0$
n (prompts)	413	107	122	116	735

Coverage at a larger sampling budget. The coverage claims of Section 5.1 rest on 32 samples per prompt. pass@k analyses of RL-trained models report a crossover: the trained model wins at small k but falls below the base model at large k, indicating a narrowing of what the model can solve (Yue et al., 2025). Tables 21 and 22 extend the budget with fresh draws for the base model and the corresponding transductive CANON checkpoint of each benchmark: 512 samples per prompt on AMC (two pooled 256-sample draws), 256 on the AIME sets, 128 on GPQA, and 64 on MATH500. On the mathematics benchmarks there is no crossover at any budget, and the gap does not narrow: on AMC the pass@k difference is $+5.4$ points at k=32 and $+7.2$ at k=512, and on AIME 2025 it grows from $+9.8$ to $+13.3$. The coverage asymmetry is one-sided at the full budget: across the three mathematics benchmarks, CANON solves 12 of the 29 prompts the base model never solves in its full draw, while no prompt solved by the base is lost. On saturated MATH500 the two arms are indistinguishable at every k, both reaching 62 of 63 covered prompts by k=16. GPQA behaves differently for a structural reason: with four answer options, k independent guesses alone reach $1 - 0.75^k$, so pass@k at large k rewards spreading mass over letters rather than reasoning. The base model spreads over more letters per prompt (2.2 against 2.0 at k=128) and closes the gap by k=128 (-1.0 points, 95% CI $[-6.1, +4.0]$), while majority-vote accuracy, which the guessing floor does not inflate, stays above the base at every k (68.2 against 63.1 at k=128).

Table 21: **pass@k and maj@k at large budgets on AMC** (83 prompts; two pooled 256-sample draws, 512 samples per prompt, so values differ slightly from the fixed-seed harness). The paired difference at k=512 is $+7.2$ points (95% CI $[+2.4, +13.3]$), and at no k does the interval favor the base model.

	k=1	8	32	64	128	256	512
Base pass@k	66.4	81.7	85.9	86.8	87.5	88.3	89.2
CANON pass@k	76.7	87.8	91.3	93.0	94.6	95.9	96.4
Base maj@k	68.7	80.7	84.3	83.1	83.1	83.1	83.1
CANON maj@k	77.1	83.1	86.8	86.8	86.8	86.8	86.8

Table 22: **pass@k at the largest measured budget, per benchmark** (base against the benchmark's transductive CANON checkpoint; paired bootstrap intervals; maj@k at the same k). *Paired on the 63 base-completed prompts; over all 500 prompts CANON reaches pass@64 97.4.

Benchmark	max k	base	CANON	Δ [95% CI]	crossover	maj@k base / CANON
AMC 2023	512	89.2	96.4	$+7.2$ $[+2.4, +13.3]$	none	83.1 / 86.8
AIME 2024	256	73.3	80.0	$+6.7$ $[0.0, +16.7]$	none	63.3 / 66.7
AIME 2025	256	60.0	73.3	$+13.3$ $[+3.3, +26.7]$	none	53.3 / 56.7
GPQA-Diamond	128	84.9	83.8	-1.0 $[-6.1, +4.0]$	k=128	63.1 / 68.2
MATH500*	64	98.4	98.4	0.0	tie (saturated)	96.8 / 95.2

K LIMITATIONS

CANON requires an extractable final answer to form a consensus, restricting it to verifiable-output domains; extending consensus extraction to free-form generation would require replacing exact-match voting with a selection rule over sampled solutions, such as minimum-Bayes-risk or medoid selection under an answer-equivalence or embedding similarity, which we leave to future work. The method distills the gap between expressed and consensus accuracy, so it offers little on saturated model-benchmark pairs and can reinforce errors where the samples collude on a wrong answer, a measurable but rare failure mode (41 of 1,493 pooled prompt-level observations, §5.2). We study a single round of self-distillation; in preliminary experiments, iterating the procedure yields at most one additional productive round before returns turn negative. Finally, the transductive regime assumes access to test prompts before answering, which is natural for batch workloads but not for interactive use; the inductive results suggest the assumption can be relaxed.