RELIABILITY-GUIDED OBJECTIVE ALLOCATION FOR TEST-TIME REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation can improve large language model reasoning without additional supervision, yet existing approaches typically apply the same optimization objective to every input. We argue that this uniform treatment ignores a fundamental asymmetry in test-time learning: reliable solutions should be consolidated, whereas uncertain ones should be explored. We introduce DiSCTT, which uses agreement among independently sampled reasoning trajectories as a label-free estimate of instance-level reliability and dynamically allocates the adaptation objective accordingly. High-consensus inputs are routed to supervised fine-tuning using modal-answer-supported pseudo-targets, while low-consensus inputs are routed to reinforcement learning with a modal-consistency-gated, population-aware reward. The routing partition is periodically recomputed as the model adapts, yielding a self-evolving curriculum between consolidation and exploration. Across diverse reasoning benchmarks and models ranging from 0.5B to 32B parameters, DiSCTT consistently improves over uniform reinforcement-learning adaptation, including gains of 9.6 and 8.7 percentage points on the three-task average for 20B and 32B models, respectively. Controlled comparisons against random and static routing, sequential SFT+RL, a label-free comparator based on SRFT (LF-SRFT),and an external-verifier variant support consensus-conditioned objective allocation under the evaluated configurations. DiSCTT also substantially reduces adaptation compute relative to full-dataset reinforcement learning and extends to open-ended generation through semantic consensus. These results suggest that test-time adaptation should treat the choice of optimization objective itself as an instance-level decision.

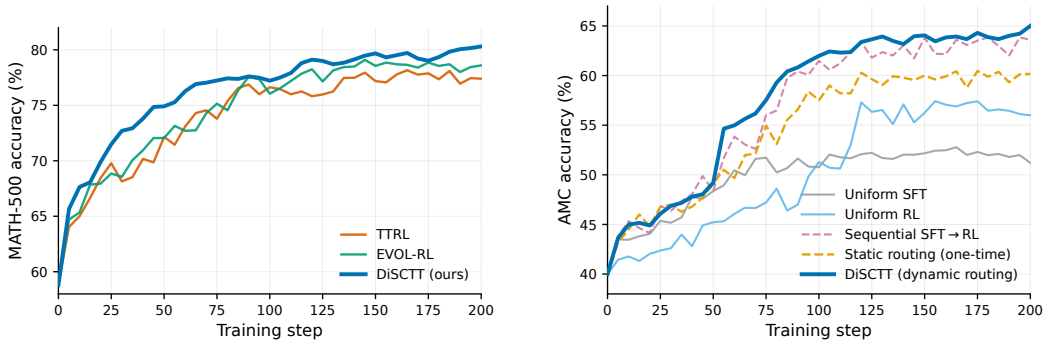
1 INTRODUCTION

Large language models (LLMs) have achieved strong performance on mathematical, scientific, and multi-hop reasoning tasks through supervised fine-tuning, chain-of-thought prompting, and reinforcement learning (Wei et al., 2022). Yet inference-time behavior remains largely static: once deployed, models apply a fixed optimization or decoding strategy to all inputs, regardless of whether a problem is trivial, ambiguous, or genuinely difficult. This mismatch has motivated growing interest in test-time adaptation, where models improve behavior during inference using only unlabeled inputs (Sun et al., 2020; Goyal et al., 2022; Hardt & Sun, 2024; Moradi et al., 2025).

Existing test-time adaptation methods typically apply a single optimization objective uniformly across all inputs. This leads to two complementary failure modes: Uniform supervised self-training saturates quickly on easy instances and offers limited benefit on harder ones, while uniform reinforcement learning exhibits slower and more oscillatory training dynamics and continues to incur rollout-based optimization even on already-solved instances, increasing computational cost. Recent methods such as TTRL (Zuo et al., 2025) and EVOL-RL (Zhou et al., 2025) improve the label-free signal but retain this uniform-objective premise. Figure 1 illustrates the resulting dynamics: uniform SFT plateaus, uniform RL converges more slowly and less consistently, and naïvely mixing the two without instance-level routing fails to close the gap.

These limitations motivate treating objective allocation as a first-order test-time decision: reliable outputs may benefit from supervised consolidation, while unresolved inputs retain rollout-based exploration.

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.



(a) Test-time training dynamics on MATH-500.

(b) Adaptation strategies on AMC.

Figure 1: Test-time adaptation dynamics under different strategies. DiSCTT continues to improve where TTRL plateaus and EVOL-RL oscillates, and outperforms the reported static and uniform strategies under the evaluated configurations.

Estimating uncertainty for reasoning problems is itself nontrivial. Token-level confidence or entropy often correlates poorly with multi-step correctness, where errors may emerge only after several intermediate steps (Geng et al., 2024). In contrast, agreement among independently sampled reasoning trajectories provides a trajectory-level signal: strong consensus typically indicates stable reasoning modes, whereas disagreement reflects variability among sampled solutions (Wang et al., 2023; Lin et al., 2024; Knappe et al., 2024). Because it requires no external labels, it is well-suited for online adaptation. However, consensus is not a correctness oracle: a model can be confidently wrong, in which case naïve consolidation onto the modal answer can entrench errors (Yan et al., 2026). DiSCTT therefore uses consensus as a relative uncertainty signal rather than a ground-truth label.

We make four main contributions:

- **Reliability-guided objective allocation.** We formulate label-free test-time adaptation as an instance-level objective-allocation problem rather than applying one update rule uniformly. A fixed-checkpoint diagnostic shows a change in relative branch performance with answer agreement: RL performs better in the observed low-consensus region, whereas SFT performs better in the observed high-consensus region. DiSCTT operationalizes this structure through periodically refreshed consensus-based routing.
- **A modal-consistency-gated, population-aware adaptation objective.** High-consensus inputs are consolidated through SFT on self-generated pseudo-targets, while low-consensus inputs receive GRPO updates with a modal-consistency-gated reward combining population-relative Jensen–Shannon novelty and bounded relevance shaping. The method requires no external labels, demonstrations, or verifier during adaptation.
- **A local theory of conditional objective allocation.** We characterize the excess risk of an arbitrary SFT/RL router through a conditional allocation-regret identity and give explicit sufficient conditions under which the optimal selector admits threshold structure on the finite consensus grid. We further analyze robustness to estimated conditional risks and threshold misspecification, while making explicit that the theory is local and does not derive the numerical routing threshold or claim global optimality of the multi-round procedure.
- **Broad empirical validation across scale, controls, and task regimes.** Across six reasoning benchmarks and models from 0.5B to 32B, DiSCTT consistently improves over uniform-RL adaptation, including 9.6- and 8.7-point average gains over TTRL at 20B and 32B. Random and static routing, sequential SFT+RL, LF-SRFT, majority voting, and external-verifier comparisons support consensus-conditioned allocation under the evaluated configurations. Event/token-based FLOP accounting and wall-clock measurements show substantial savings relative to full-dataset RL, while semantic consensus extends the framework to machine translation and summarization.

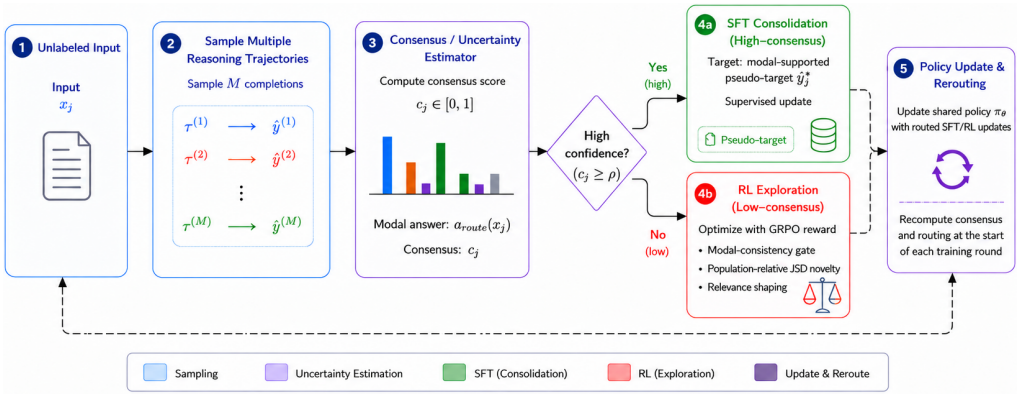


Figure 2: Overview of DiSCTT. Multiple reasoning trajectories are sampled per input and grouped by final answer to estimate consensus. Inputs are routed to supervised fine-tuning (high consensus) or reinforcement learning (low consensus), with routing recomputed at the start of each training round to form a self-evolving adaptation loop.

2 Related Work

Self-consistency and label-free adaptation. Self-consistency aggregates sampled reasoning trajectories without supervision (Wang et al., 2023), while self-improvement methods such as STaR train on model-generated rationales (Zelikman et al., 2022; Huang et al., 2023). TTRL converts majority agreement into a label-free RL reward (Zuo et al., 2025), EVOL-RL adds semantic-novelty and exploration-oriented shaping (Zhou et al., 2025), and SCRL selectively constructs pseudo-label signals according to uncertainty (Yan et al., 2026). These methods adapt how self-supervision is constructed or weighted; DiSCTT instead uses answer agreement to select between supervised consolidation and reinforcement-learning exploration and periodically recomputes that assignment. Consensus can still reinforce systematic errors, paralleling confirmation bias in pseudo-labeling (Arazo et al., 2020; Huang et al., 2025).

Test-time adaptation and adaptive computation. Test-time training adapts models on unlabeled inputs (Sun et al., 2020; Hardt & Sun, 2024; Moradi et al., 2025), while search-based reasoning and test-time scaling allocate inference computation across candidate trajectories (Yao et al., 2023; Besta et al., 2024; Snell et al., 2025). Curriculum and intrinsic-motivation methods similarly vary optimization pressure within a fixed objective (Bengio et al., 2009; Pathak et al., 2017). DiSCTT instead allocates between distinct optimization regimes at the instance level.

Reinforcement learning and SFT/RL integration. RL-based reasoning methods optimize human, learned, or verifiable reward signals (Christiano et al., 2017; Ouyang et al., 2022; Lightman et al., 2024), and we use GRPO (Shao et al., 2024) as the RL backbone. Most closely, SRFT (Fu et al., 2026) combines SFT and RL in a single stage through entropy-dependent weighting, showing that their relative influence can depend on uncertainty. Original SRFT uses external demonstrations and correctness supervision, whereas DiSCTT operates on unlabeled test inputs and assigns one objective per input according to answer agreement, with assignments recomputed as the policy evolves. Appendix I.6 compares against LF-SRFT, our label-free SRFT-based comparator.

3 Method

DiSCTT allocates supervised consolidation and reinforcement-learning exploration according to agreement among sampled answers, with routing periodically refreshed under the evolving policy (Figure 2).

3.1 Problem Setting

Let $\mathcal{D} = \{x_j\}_{j=1}^{N_d}$ be unlabeled reasoning inputs, π_θ the raw model distribution, and q_θ its decoding distribution after temperature and top-p transformations. For each x_j, sample $\mathcal{Y}_j = \{y_{j,i}\}_{i=1}^{M}$ with $y_{j,i} \sim q_\theta(\cdot \mid x_j)$ and $y_{j,i} = (r_{j,i}, a_{j,i})$. Adaptation uses only these self-generated trajectories.

3.2 Agreement-Based Uncertainty Estimation

We use final-answer agreement as an empirical reliability signal. Let $\mathcal{V}_j = \{i \in \{1, \dots, M\} : a_{j,i} \text{ is valid}\}$ index the routing generations with valid canonical answers.

The empirical modal-consensus ratio is

$$c_j = \begin{cases} \dfrac{1}{M} \max\limits_{a} \displaystyle\sum_{i \in \mathcal{V}_j} \mathbf{1}[a_{j,i} = a], & \mathcal{V}_j \neq \varnothing, \\ 0, & \mathcal{V}_j = \varnothing. \end{cases} \qquad c_j \in [0, 1]. \tag{1}$$

Invalid generations remain in the denominator. Modal consensus measures agreement with the most frequent valid answer under the current policy and does not certify correctness.

When $\mathcal{V}_j \neq \varnothing$, the valid routing samples define the routing-time modal answer

$$a_{\text{route}}(x_j) = \text{TieBreak}\left(\arg\max_{a} \sum_{i \in \mathcal{V}_j} \mathbf{1}[a_{j,i} = a] \right). \tag{2}$$

If $\mathcal{V}_j = \varnothing$, set $a_{\text{route}}(x_j) = \varnothing$ and $c_j = 0$, route the input to RL, and construct no SFT target. With $M = 8$, the cutoff $c_j \geq 4/8$ can include a 4–4 tie; we therefore use "modal" rather than "majority" answer. A fixed extractor and deterministic likelihood-based tie breaker are detailed in Appendix B.

Working assumption. For ground-truth answer $a^*(x)$, we assume $\Pr(a_{\text{route}}(x) = a^*(x) \mid c)$ is non-decreasing in c. This is an empirical assumption: systematic errors can yield confident incorrect agreement. It is evaluated post hoc in Appendices I.1 and I.5; ground truth never guides adaptation.

Given a threshold ρ, the consensus partition is

$$\mathbb{D}_{\text{conf}} = \{\, x_j \mid c_j \geq \rho \,\}, \qquad \mathbb{D}_{\text{unc}} = \{\, x_j \mid c_j < \rho \,\}. \tag{3}$$

This partition is policy-dependent: as π_θ improves, instances may migrate between subsets.

Hard routing. Sequential SFT+RL and LF-SRFT controls that apply both objectives to every input are evaluated in Appendix I.6.

3.3 Supervised Consolidation on $\mathbb{D}_{\text{conf}}$

For $x_j \in \mathbb{D}_{\text{conf}}$, let $\mathcal{C}_j = \{y_{j,i} \in \mathcal{Y}_j : a_{j,i} = a_{\text{route}}(x_j)\}$ and sample $y_j^\star \sim \text{Uniform}(\mathcal{C}_j)$. For minibatch $\mathcal{B}$, let $\mathcal{S}(y_j^\star)$ denote active generated-response positions, excluding prompt and padding tokens. The token-weighted SFT loss is

$$\mathcal{L}_{\text{SFT}}^{(\mathcal{B})}(\theta) = -\frac{\displaystyle\sum_{j \in \mathcal{B}} \sum_{t \in \mathcal{S}(y_j^\star)} \log \pi_\theta\big(y_{j,t}^\star \mid x_j, y_{j,<t}^\star\big)}{\displaystyle\sum_{j \in \mathcal{B}} |\mathcal{S}(y_j^\star)|}. \tag{4}$$

The target answer has at least $\lceil \rho M \rceil$ sampled support, although its reasoning trace need not recur; the target is cached for both SFT epochs and refreshed next round.

3.4 Reinforcement Learning on $\mathbb{D}_{\text{unc}}$

For $x_j \in \mathbb{D}_{\text{unc}}$, we apply Group Relative Policy Optimization (GRPO; Shao et al., 2024) using fresh groups of $N = 32$ completions.

The N RL trajectories form a fresh pool distinct from routing. Let $\mathcal{V}_{\mathrm{RL}}(x) = \{i \in \{1, \ldots, N\} : a_i \text{ is valid}\}$ index its valid generations.

When $\mathcal{V}_{\mathrm{RL}}(x) \neq \varnothing$, we define

$$a_{\mathrm{RL}}(x) = \mathrm{TieBreak}\left(\arg\max_{a} \sum_{i \in \mathcal{V}_{\mathrm{RL}}(x)} \mathbf{1}[a_i = a] \right). \tag{5}$$

Only valid answers participate; if $\mathcal{V}_{\mathrm{RL}}(x) = \varnothing$, set $a_{\mathrm{RL}}(x) = \varnothing$ and all task rewards to zero.

Modal-consistency-gated reward. Each completion $y_i = (r_i, a_i)$ receives the multiplicative reward

$$R(y_i) = \underbrace{\mathbf{1}[\, i \in \mathcal{V}_{\mathrm{RL}}(x)\,]\, \mathbf{1}[\, a_i = a_{\mathrm{RL}}(x)\,]}_{\text{validity and modal-consistency gate}} \cdot \underbrace{\big(\alpha + \beta\, \mathrm{JSD}_{\mathrm{nov}}(r_i)\big)}_{\text{population-relative novelty}} \cdot \underbrace{\big(\varepsilon + (1-\varepsilon)\, g_{\mathrm{rel}}(r_i)\big)}_{\text{relevance shaping}} \tag{6}$$

We use $\alpha = \beta = 1$ and $\varepsilon = 0.2$. Invalid or non-modal rollouts receive zero reward; valid modal-answer-matching rollouts satisfy $0.2 \leq R(y_i) \leq 1 + \log 2$. This is a scalar bound, not a correctness or gradient guarantee; an incorrect modal answer can suppress a correct minority.

Population-relative novelty. For a fresh rollout group, let $\mathcal{I}_{\mathrm{RL}}(x) = \{j \in \mathcal{V}_{\mathrm{RL}}(x) : a_j = a_{\mathrm{RL}}(x)\}$ and let T_i contain the valid reasoning-token positions of r_i, excluding prompt, final-answer, padding, EOS, and post-EOS positions. At position t, define $\mathcal{I}_t(x) = \{j \in \mathcal{I}_{\mathrm{RL}}(x) : t \in T_j\}$ and

$$\bar{p}_t(\cdot \mid x) = \frac{1}{|\mathcal{I}_t(x)|} \sum_{j \in \mathcal{I}_t(x)} \pi_{\theta_{\mathrm{roll}}}(\cdot \mid x, r_{j,<t}), \tag{7}$$

using raw rollout-snapshot probabilities. For $i \in \mathcal{I}_{\mathrm{RL}}(x)$ with $|T_i| > 0$,

$$\mathrm{JSD}_{\mathrm{nov}}(r_i) = \frac{1}{|T_i|} \sum_{t \in T_i} \mathrm{JS}(\pi_{\theta_{\mathrm{roll}}}(\cdot \mid x, r_{i,<t}) \,\|\, \bar{p}_t(\cdot \mid x)), \tag{8}$$

and $\mathrm{JSD}_{\mathrm{nov}}(r_i) = 0$ otherwise. The scored trajectory belongs to its own reference, so singleton references contribute zero. Natural-log JSD lies in $[0, \log 2]$; retained distributions are detached and reused without an additional policy forward pass.

Relevance shaping. For reasoning steps $\{s_{i,j}\}_{j=1}^{n_i}$ and fixed encoder $e(\cdot)$, define for $n_i > 0$

$$g_{\mathrm{rel}}(r_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{1 + \cos(e(s_{i,j}), e(x))}{2} \in [0, 1], \tag{9}$$

and set $g_{\mathrm{rel}}(r_i) = 0$ when $n_i = 0$. Thus the relevance factor lies in $[\varepsilon, 1]$ and downweights semantic drift without eliminating it.

GRPO update. Each fresh rollout group is used for one optimizer update. Rewards, novelty distributions, and encoder outputs are detached.

Let $\bar{R}$ and s_R denote the mean and standard deviation of rewards within a rollout group. We compute detached group-relative advantages as

$$A_i = \frac{R_i - \bar{R}}{s_R + \delta_{\mathrm{adv}}}, \tag{10}$$

where $\delta_{\mathrm{adv}} = 10^{-8}$. Equal rewards give zero group-relative advantages; reference regularization can still contribute.

Let $\mathcal{G}_i$ denote generated-response positions used by the policy loss (unlike reasoning-only T_i), and define $r_{i,t}(\theta) = \pi_\theta(y_{i,t} \mid x, y_{i,<t}) / \pi_{\theta_{\mathrm{old}}}(y_{i,t} \mid x, y_{i,<t})$. These ratios use raw model probabilities, whereas trajectory sampling uses q_θ.

The policy component of the GRPO objective uses the clipped surrogate

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{G}_i|}\sum_{t\in\mathcal{G}_i}\min[r_{i,t}(\theta)A_i,\ \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\,A_i] + \beta_{\mathrm{KL}}\,\widehat{\mathcal{K}}(\theta), \tag{11}$$

The policy loss is response-token normalized; we use $\epsilon_{\mathrm{clip}} = 0.2$, $\beta_{\mathrm{KL}} = 0.04$, and frozen $\pi_{\mathrm{ref}} = \pi_{\theta_0}$. Because each fresh group receives one update, $r_{i,t} = 1$ at loss evaluation and ratio clipping is inactive.

The term $\widehat{\mathcal{K}}$ is a nonnegative sampled reference regularizer. At a fixed prefix its expectation under raw π_θ equals $D_{\mathrm{KL}}(\pi_\theta\|\pi_{\mathrm{ref}})$; because rollouts are sampled from q_θ, we do not treat it as an unbiased full-vocabulary KL estimator. Its exact form is given in Appendix D.

3.5 Self-Evolving Curriculum

Adaptation uses five rounds. Each begins with $M = 8$ fresh routing samples per input and fixes the resulting partition for two SFT epochs followed by eight RL epochs. Both phases update the same LoRA adapter. The threshold remains $\rho = 0.45$, equivalent to $c \geq 4/8$ for $M = 8$; assignments and pseudo-targets change with the policy. A final routing refresh is diagnostic only and causes no updates. Algorithm 1 gives the procedure, and Appendix B discusses streaming considerations.

3.6 Local Analysis of Objective Allocation

Fix a checkpoint and the two branch procedures, including their respective budgets. For $c > 0$, let $R_a(c)$ be the finite expected evaluation loss after branch $a \in \{\mathrm{SFT}, \mathrm{RL}\}$, conditional on $C = c$. All-invalid inputs have $C = 0$ and are assigned to RL. Assuming $\Pr(C > 0) > 0$, define the conditional allocation risk

$$\mathcal{R}_+(g) = \mathbb{E}[R_{g(C)}(C) \mid C > 0], \qquad g^*(c) \in \arg\min_{a\in\{\mathrm{SFT},\mathrm{RL}\}} R_a(c). \tag{12}$$

Proposition (conditional allocation regret). Writing $\Delta(c) = R_{\mathrm{SFT}}(c) - R_{\mathrm{RL}}(c)$, every router g satisfies

$$\mathcal{R}_+(g) - \mathcal{R}_+(g^*) = \mathbb{E}[|\Delta(C)|\,\mathbf{1}[g(C) \neq g^*(C)] \mid C > 0]\,. \tag{13}$$

At each consensus value, selecting a non-minimizing branch incurs exactly $|\Delta(c)|$ excess risk; taking expectation proves the identity. Consequently, allocation errors matter most where branch risks differ. On the finite consensus grid, if $\Delta(c)$ is additionally assumed non-increasing, an optimal selector can be chosen as an upper-threshold rule: SFT is weakly preferable exactly when $\Delta(c) \leq 0$. Appendix E gives the full analysis, including risk-estimation and finite-grid bounds. These local results do not determine $\rho = 0.45$ or establish optimality of multi-round DiSCTT.

4 Experiments and Results

Evaluation protocol. Each benchmark is independently adapted for five rounds on unlabeled test questions and evaluated transductively with one fresh generation per question. The six-benchmark and 20B/32B studies use ten independent seeds; the main table reports mean ± sample SD. Ground truth is used only for evaluation and post-hoc diagnostics, with shared hyperparameters and no accuracy-based checkpoint selection. Full protocol details and held-out transfer are in Appendices A and J.2.

4.1 Main Benchmark Results

Table 1 reports the original six-benchmark suite. Across model families and scales from 0.5B to 7B, DiSCTT consistently outperforms the base model and the uniform-RL test-time adaptation baselines TTRL and EVOL-RL, with average gains of $+7.8$ to $+10.8$ points over TTRL. The allocation is data-dependent: easier datasets receive more SFT updates, whereas AIME-2024 retains most of its examples in RL (Appendix I.2). Per-difficulty dynamics further show that SFT supplies fast early gains on easier instances while RL sustains improvement on harder instances (Appendix J.3).

Table 1: Mean final-answer accuracy ($\pm$ sample standard deviation) across six reasoning benchmarks under test-time adaptation. Standard deviations are sample SDs across 10 independent end-to-end runs. "AIME-24" denotes AIME-2024 and "Hotpot" denotes HotpotQA. Average is the unweighted mean across benchmarks.

Model	Method	AMC	MATH-500	AIME-24	GPQA-D	Hotpot	MMLU	Avg.
Qwen-2.5-0.5B	Base	$12.2_{\pm 1.42}$	$27.6_{\pm 0.22}$	$1.3_{\pm 1.72}$	$14.1_{\pm 0.34}$	$1.2_{\pm 0.15}$	$36.5_{\pm 0.13}$	15.5
	TTRL	$20.3_{\pm 1.42}$	$37.2_{\pm 0.25}$	$1.7_{\pm 1.76}$	$16.2_{\pm 0.37}$	$10.3_{\pm 0.17}$	$40.2_{\pm 0.16}$	21.0
	EVOL-RL	$23.0_{\pm 1.58}$	$40.0_{\pm 0.21}$	$2.3_{\pm 2.25}$	$18.2_{\pm 0.29}$	$13.7_{\pm 0.13}$	$43.7_{\pm 0.14}$	23.5
	DiSCTT	$27.7_{\pm 0.79}$	$49.8_{\pm 0.23}$	$3.0_{\pm 2.46}$	$24.2_{\pm 0.34}$	$18.7_{\pm 0.14}$	$49.4_{\pm 0.12}$	**28.8**
LLaMA-3.2-1B	Base	$15.5_{\pm 1.97}$	$14.0_{\pm 0.24}$	$0.3_{\pm 1.05}$	$7.1_{\pm 0.34}$	$11.7_{\pm 0.16}$	$14.9_{\pm 0.10}$	10.6
	TTRL	$22.5_{\pm 1.67}$	$28.0_{\pm 0.21}$	$1.0_{\pm 1.61}$	$10.6_{\pm 0.34}$	$23.3_{\pm 0.18}$	$29.2_{\pm 0.14}$	19.1
	EVOL-RL	$25.5_{\pm 1.58}$	$31.2_{\pm 0.23}$	$1.7_{\pm 1.76}$	$14.1_{\pm 0.34}$	$31.4_{\pm 0.15}$	$36.8_{\pm 0.15}$	23.4
	DiSCTT	$26.5_{\pm 1.29}$	$39.6_{\pm 0.22}$	$2.3_{\pm 1.61}$	$23.8_{\pm 0.32}$	$40.3_{\pm 0.17}$	$47.0_{\pm 0.13}$	**29.9**
Qwen3-1.7B	Base	$19.8_{\pm 1.84}$	$47.0_{\pm 0.13}$	$6.7_{\pm 2.22}$	$23.2_{\pm 0.24}$	$25.4_{\pm 0.12}$	$54.2_{\pm 0.05}$	29.4
	TTRL	$40.2_{\pm 1.42}$	$55.2_{\pm 0.18}$	$16.3_{\pm 1.89}$	$25.8_{\pm 0.29}$	$49.2_{\pm 0.14}$	$62.7_{\pm 0.06}$	41.6
	EVOL-RL	$44.8_{\pm 1.84}$	$53.8_{\pm 0.11}$	$23.3_{\pm 1.57}$	$28.8_{\pm 0.24}$	$56.4_{\pm 0.11}$	$67.3_{\pm 0.07}$	45.7
	DiSCTT	$48.8_{\pm 1.32}$	$67.0_{\pm 0.16}$	$26.7_{\pm 1.57}$	$29.3_{\pm 0.29}$	$67.1_{\pm 0.13}$	$68.4_{\pm 0.05}$	**51.2**
LLaMA-3.2-3B	Base	$12.5_{\pm 1.67}$	$35.7_{\pm 0.11}$	$3.7_{\pm 2.46}$	$13.6_{\pm 0.16}$	$39.9_{\pm 0.10}$	$55.4_{\pm 0.07}$	26.8
	TTRL	$27.2_{\pm 1.42}$	$45.4_{\pm 0.13}$	$9.7_{\pm 1.89}$	$16.2_{\pm 0.24}$	$57.3_{\pm 0.12}$	$61.2_{\pm 0.08}$	36.2
	EVOL-RL	$32.2_{\pm 1.42}$	$47.2_{\pm 0.11}$	$10.0_{\pm 1.57}$	$20.7_{\pm 0.24}$	$61.7_{\pm 0.08}$	$64.5_{\pm 0.07}$	39.4
	DiSCTT	$39.2_{\pm 1.21}$	$55.6_{\pm 0.16}$	$16.3_{\pm 1.89}$	$24.3_{\pm 0.21}$	$66.7_{\pm 0.09}$	$69.8_{\pm 0.06}$	**45.3**
Qwen3-4B	Base	$38.5_{\pm 1.29}$	$51.4_{\pm 0.06}$	$10.0_{\pm 1.57}$	$27.3_{\pm 0.16}$	$39.4_{\pm 0.08}$	$69.5_{\pm 0.03}$	39.4
	TTRL	$46.5_{\pm 1.29}$	$60.6_{\pm 0.08}$	$17.0_{\pm 1.89}$	$28.8_{\pm 0.24}$	$61.5_{\pm 0.06}$	$72.4_{\pm 0.04}$	47.8
	EVOL-RL	$51.8_{\pm 1.21}$	$64.6_{\pm 0.11}$	$20.3_{\pm 1.89}$	$30.3_{\pm 0.16}$	$66.3_{\pm 0.03}$	$74.5_{\pm 0.02}$	51.3
	DiSCTT	$57.0_{\pm 1.05}$	$75.2_{\pm 0.06}$	$23.7_{\pm 1.05}$	$38.4_{\pm 0.16}$	$72.9_{\pm 0.03}$	$81.3_{\pm 0.01}$	**58.1**
Qwen-2.5-7B	Base	$39.7_{\pm 1.42}$	$58.8_{\pm 0.06}$	$10.7_{\pm 1.41}$	$27.3_{\pm 1.09}$	$51.2_{\pm 0.03}$	$76.2_{\pm 0.04}$	44.0
	TTRL	$51.0_{\pm 1.29}$	$74.2_{\pm 0.08}$	$20.3_{\pm 1.05}$	$28.8_{\pm 1.12}$	$62.1_{\pm 0.02}$	$76.4_{\pm 0.03}$	52.2
	EVOL-RL	$55.0_{\pm 1.18}$	$73.4_{\pm 0.06}$	$26.3_{\pm 1.05}$	$29.3_{\pm 1.09}$	$66.1_{\pm 0.04}$	$77.9_{\pm 0.05}$	54.7
	DiSCTT	$59.5_{\pm 1.05}$	$82.2_{\pm 0.06}$	$29.7_{\pm 1.05}$	$34.9_{\pm 0.16}$	$73.7_{\pm 0.01}$	$83.3_{\pm 0.02}$	**60.6**

A fixed-checkpoint Qwen3-1.7B-Base/MATH-500 diagnostic supports the routing premise: RL is stronger below the cutoff (35.89% vs. 30.24%), whereas SFT is stronger at or above it (67.86% vs. 65.48%; Appendix E.5). Under the same five-round schedule, random routing does not improve over uniform RL, static consensus routing recovers much of the gain, and dynamic re-routing adds a further $+4.2$ points (Appendix I.3). Sequential SFT+RL trails DiSCTT by 5.3–5.4 points, LF-SRFT by 3.2 points on the six-benchmark average (Appendix I.6), and matched-budget majority voting does not recover the gain (Appendix I.4).

4.2 Scaling to 20B and 32B Models

A central concern is whether routing remains useful once the base model has substantially higher native accuracy and consensus. We therefore evaluate the unchanged DiSCTT configuration on `openai/gpt-oss-20b` and `Qwen/Qwen3-32B` over AMC, AIME-2024, and GPQA-Diamond. We keep the same M, ρ, reward, and adaptation schedule used by the smaller models.

DiSCTT improves over TTRL by 9.6 and 8.7 points on the three-task average at 20B and 32B, respectively, and exceeds EVOL-RL in every reported model–dataset pair. At 32B, the gains over TTRL are $+7.3$ on AMC, $+11.7$ on AIME-2024, and $+7.2$ on GPQA-Diamond; these experiments are not a controlled scaling-law study. Routing also shifts toward SFT with model strength: initial-to-final SFT fractions on AMC are 50%→70%, 63%→80%, and 75%→88%, and on AIME-2024 13.3%→26.7%, 40%→58%, and 43%→67% for 4B, 20B, and 32B, respectively, while roughly one-third of AIME-2024 remains in RL at 32B.

Table 2: Larger-scale evaluation with the DiSCTT configuration unchanged. Values are mean final-answer accuracies (%) across 10 independent end-to-end runs with distinct training and sampling seeds. Each run is evaluated using one fresh generation per question. The final column is the unweighted mean over AMC, AIME-2024, and GPQA-Diamond.

Model	Method	AMC	AIME-24	GPQA-D	Avg.
GPT-OSS-20B	Base	62.0	33.3	44.9	46.7
	TTRL	68.7	41.7	53.2	54.5
	EVOL-RL	73.5	47.3	58.9	59.9
	DiSCTT	**77.8**	**51.7**	**62.7**	**64.1**
Qwen3-32B	Base	77.5	36.7	67.4	60.5
	TTRL	81.7	43.0	79.2	68.0
	EVOL-RL	86.2	49.3	82.9	72.8
	DiSCTT	**89.0**	**54.7**	**86.4**	**76.7**

Table 3: Final state of initially confidently-wrong examples for Qwen3-4B-Base. "Conf. correct" denotes a correct high-consensus modal answer at the end; "low consensus" denotes an example ending in the RL partition.

Benchmark	Initial wrong	Conf. correct	Low consensus	Still wrong
MATH-500	62	47	4	11
AMC	4	1	2	1
AIME-24	2	0	1	1

4.3 Evolution of Consensus Reliability During Adaptation

Consensus is not a correctness oracle, so we separately track high-consensus coverage and pseudo-label precision. For Qwen3-1.7B-Base, precision increases from 66.7% to 95.0% on AMC, 76.2% to 93.4% on MATH-500, 50.0% to 80.0% on AIME-2024, and 68.3% to 89.0% on GPQA-Diamond, while coverage also increases (Appendices I.1 and I.5). We additionally track the final states of initially confidently-wrong examples under shared-policy adaptation:

On MATH-500, 47 of 62 initially confidently-wrong examples (76%) end confidently correct, four (6%) move to low consensus, and 11 (18%) remain confidently wrong. The much smaller AMC and AIME-2024 subsets preclude strong conclusions. These initial-to-final transitions are observational under shared-policy updates and periodic re-routing and therefore do not identify which objective caused an individual correction.

4.4 Extension to Open-Ended Tasks

DiSCTT-OE clusters 32 sampled outputs to replace exact-answer agreement. It improves over the base model, cluster voting, and uniform RL on both tasks and backbones, closing 41–74% of the reference-guided best-of-32–base gap. The comparator is diagnostic and does not bound adapted-policy performance. Full results and implementation details are in Appendix J.1 and Table 16.

4.5 Compute Efficiency

Across three configurations, estimated adaptation FLOPs decrease by 52.8–61.6% relative to TTRL and 52.9–67.6% relative to EVOL-RL (Table 5). The event/token proxy includes routing and auxiliary model events, while excluding final diagnostics and evaluation (Appendix F). Independently measured training time decreases by 40–49% (Appendix G). Pure SFT is cheaper but yields smaller accuracy gains in the reported comparison (Appendix H).

5 LIMITATIONS

Five limitations bound the claims above. (i) Consensus is a useful label-free uncertainty signal, not a correctness oracle. Confidently wrong agreement, finite-M noise, and threshold errors remain possible; the verifier comparison and transition analysis show mitigation rather than elimination. (ii) The core evaluation is transductive. We observe held-out cross-domain transfer (Appendix J.2), but we do not evaluate a non-stationary protocol in which the incoming distribution changes online, such as adapting first on MATH-500 and then switching to MathOlympiad. (iii) Consensus requires M samples *per input*. This does not require a batch of distinct queries, but it increases latency and makes strict single-pass streaming less attractive. (iv) The larger-scale experiments reach 20B and 32B only on AMC, AIME-2024, and GPQA-Diamond; they are not a controlled scaling-law study, and behavior on substantially larger frontier models remains open. (v) We keep ρ fixed. Dynamic re-routing already changes assignments as the model evolves, and sensitivity is broad on the discrete $M = 8$ consensus grid, but an explicitly adaptive ρ_t schedule and a learned soft/three-way router remain untested.

6 CONCLUSION

DiSCTT uses answer agreement to allocate supervised consolidation and reinforcement-learning exploration during label-free test-time adaptation. A local conditional-risk analysis characterizes allocation regret, while routing controls and branch diagnostics support the design empirically. Across the reported models and tasks, DiSCTT improves accuracy and reduces compute relative to uniform-RL adaptation. Semantic consensus extends the approach to open-ended generation; incorrect agreement and the transductive evaluation setting remain limitations.

AI USE STATEMENT

Generative AI tools were used for manuscript-preparation assistance, including language editing and LaTeX formatting. All AI-assisted material incorporated into the manuscript was reviewed by the authors for correctness and consistency with the underlying experiments and analyses. The authors take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

DiSCTT performs unsupervised parameter updates at test time and is designed to improve reasoning performance without external labels. Such adaptation can also reinforce erroneous high-consensus outputs, amplify biases inherited from the base model, or otherwise alter model behavior in unintended ways. The experiments in this work evaluate reasoning accuracy, adaptation behavior, and computational efficiency; they do not establish preservation of safety, alignment, calibration, privacy, or robustness to misuse after adaptation.

The reported reduction in adaptation compute relative to full-dataset RL is an efficiency result and should not be interpreted as a safety benefit. Deployments of test-time adaptation systems should therefore monitor both task performance and behavioral changes introduced by online updates, and should retain appropriate access controls, evaluation, and human oversight for the intended application domain.

REPRODUCIBILITY STATEMENT

We provide exact checkpoint identifiers and hyperparameters in Appendix A, the full DiSCTT procedure in Appendix C, event/token-based FLOP accounting and measured wall-clock results in Appendices F–G, the pure-SFT comparison in Appendix H, and additional routing controls, ablations, sensitivity analyses, and open-ended-task details in the remaining appendices. These sections document the optimization schedule, routing parameters, reward components, and evaluation controls needed to reproduce the reported comparisons.

REFERENCES

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020. doi: 10.1109/IJCNN48605.2020.9207304. URL https://doi.org/10.1109/IJCNN48605.2020.9207304.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of Thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172. Springer, 2013.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv:2402.03216, 2024. URL https://arxiv.org/abs/2402.03216.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

Daniel Deutsch, Eleftheria Briakou, Isaac Rayburn Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. WMT24++: Expanding the language coverage of WMT24 to 55 languages & dialects. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12257–12284. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.634. URL https://aclanthology.org/2025.findings-acl.634/.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. SRFT: A single-stage method with supervised and reinforcement fine-tuning for reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a79699db176ed0efc04a9da171e52112-Abstract-Conference.html. arXiv:2506.19767.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 6577–6595, 2024.

Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and J. Zico Kolter. Test time adaptation via conjugate pseudo-labels. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/28e9eff897f98372409b40ae1ed3ea4c-Abstract-Conference.html.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *International Conference on Learning Representations*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

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.

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 *International Conference on Learning Representations*, 2025.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1051–1068, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Tim Knappe, Ryan Li, Ayush Chauhan, Kaylee Chhua, Kevin Zhu, and Sean O'Brien. Semantic self-consistency: Enhancing language model reasoning via semantic weighting. arXiv:2410.07839, 2024. URL https://arxiv.org/abs/2410.07839.

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 *International Conference on Learning Representations*, 2024.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, 2004.

Lei Lin, Jiayi Fu, Pengli Liu, Qingyang Li, Yan Gong, Junchen Wan, Fuzheng Zhang, Zhongyuan Wang, Di Zhang, and Kun Gai. Just ask one more time! self-agreement improves reasoning of language models in (almost) all scenarios. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3829–3852, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Claudia Malzer and Marcus Baum. A hybrid approach to hierarchical density-based cluster selection. In *2020 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI)*, pp. 223–228. IEEE, 2020.

Mohammad Mahdi Moradi, Hossam Amer, Sudhir Mudur, Weiwei Zhang, Yang Liu, and Walid Ahmed. Continuous self-improvement of large language models by test-time training with verifier-driven sample selection. arXiv:2505.19475, 2025. URL https://arxiv.org/abs/2505.19475.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, 2017.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, 2018. doi: 10.18653/v1/W18-6319.

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.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892, 2020. doi: 10.18653/v1/2020.acl-main.704.

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

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 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 *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28957–28970, 2026. doi: 10.18653/v1/2026.acl-long.1337. URL https://aclanthology.org/2026.acl-long.1337/. arXiv:2603.19880.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.

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

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, 2025. doi: 10.18653/v1/2025.findings-acl.547. URL https://aclanthology.org/2025.findings-acl.547/.

Yujun Zhou, Zhenwen Liang, Haolin Liu, Wenhao Yu, Kishan Panaganti, Linfeng Song, Dian Yu, Xiangliang Zhang, Haitao Mi, and Dong Yu. Evolving language models without labels: Majority drives selection, novelty promotes variation. arXiv:2509.15194, 2025. URL https://arxiv.org/abs/2509.15194. EVOL-RL.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4376. URL https://papers.nips.cc/paper_files/paper/2025/hash/be690ea16f005c174f6c4102a5970e67-Abstract-Conference.html. arXiv:2504.16084.

APPENDIX ROADMAP

Appendix A provides experimental details and exact checkpoint identifiers. Appendix B specifies answer extraction and tie-breaking and discusses streaming. Appendix C gives the full algorithm, and Appendix E develops the conditional-allocation regret identity, sufficient conditions for threshold structure, estimated-risk robustness, and finite-grid threshold misspecification, followed by a fixed-checkpoint empirical comparison of SFT and RL across consensus bins. Appendices F–H contain event/token-based FLOP accounting, independent wall-clock measurements, and the pure-SFT tradeoff. The remaining experimental appendices analyze consensus reliability, routing dynamics, hard-versus-combined objectives, majority voting, external-verifier routing, open-ended generation, OOD transfer, training dynamics, reward ablations, and hyperparameter sensitivity.

A EXPERIMENTAL SETUP

We evaluate DiSCTT on six diverse reasoning benchmarks spanning mathematical, scientific, multi-hop, and knowledge-intensive reasoning. Specifically, we consider three mathematical reasoning datasets (AMC, MATH-500, and AIME-2024), one graduate-level scientific reasoning benchmark (GPQA-Diamond), one multi-hop question answering benchmark (HotpotQA), and one broad knowledge and reasoning benchmark (MMLU). For compactness, tables abbreviate AIME-2024 as "AIME-24" and HotpotQA as "Hotpot."

Transductive test-time adaptation protocol. The main evaluation follows a transductive test-time adaptation protocol. Each benchmark is adapted independently starting from the original pretrained checkpoint; model state is never carried from one benchmark to another. For a given benchmark, its input questions are used as the unlabeled adaptation set, without access to their ground-truth answers. After the final adaptation round, the adapted checkpoint is evaluated on those same benchmark inputs using fresh generations that are independent of the routing and training rollouts. The main benchmark tables therefore measure transductive adaptation performance rather than held-out generalization.

Held-out transfer is evaluated separately in Appendix J.2: there, a model is adapted on a source benchmark and evaluated on disjoint target-task inputs without additional target adaptation.

The original six-benchmark suite uses 10 independent seeds and reports mean $\pm$ sample standard deviation. The 20B and 32B experiments also use 10 independent seeds. Unless explicitly stated otherwise, the auxiliary exact-match controls and sensitivity analyses use the same 10-run final-evaluation protocol, with one fresh generation per question for each independently adapted checkpoint; these auxiliary tables report the resulting mean accuracy as a point estimate without displaying run-to-run standard deviations.

Use of ground-truth labels. Ground-truth answers are never used for routing, pseudo-target construction, reward computation, gradient updates, or early stopping. All reported main results use the checkpoint obtained after the fixed five-round adaptation schedule; we do not select the best intermediate checkpoint according to benchmark accuracy. The routing threshold $\rho = 0.45$, optimization hyperparameters, and training schedule are shared across the reported benchmarks rather than tuned separately on their ground-truth labels. Ground-truth answers are used only after adaptation for final evaluation and for explicitly labeled post-hoc analyses such as consensus precision, conditional branch accuracy, and sensitivity diagnostics. These post-hoc analyses do not alter the reported adapted checkpoints.

Dataset manifest. The main evaluation uses AMC-2023 (40 problems), MATH-500 (Lightman et al., 2024) (500), AIME-24 (30), GPQA-Diamond (Rein et al., 2024) (198), HotpotQA distractor validation (Yang et al., 2018) (7,405), and MMLU test (Hendrycks et al., 2021) (14,042). GPQA-Diamond and MMLU are evaluated using multiple-choice final-answer accuracy, while HotpotQA uses exact match (EM). MMLU is micro-averaged over all test examples across subjects. The six-task average reported in Table 1 is the unweighted mean of the six benchmark-level scores.

The original six-benchmark suite uses six checkpoints:

- `Qwen/Qwen2.5-0.5B-Instruct`

- `meta-llama/Llama-3.2-1B-Instruct`
- `Qwen/Qwen3-1.7B-Base`
- `meta-llama/Llama-3.2-3B-Instruct`
- `Qwen/Qwen3-4B-Base`
- `Qwen/Qwen2.5-7B-Instruct`

This mix deliberately includes both Base and Instruct checkpoints to test whether the routing effect depends on instruction tuning. The larger-scale study in Section 4.2 additionally evaluates `openai/gpt-oss-20b` and `Qwen/Qwen3-32B` on AMC, AIME-2024, and GPQA-Diamond, using the same DiSCTT hyperparameters.

Model-specific generation settings. For `openai/gpt-oss-20b`, we use the model's Harmony response format with reasoning effort set to `low`. This checkpoint is a mixture-of-experts model rather than a dense 20B model; "20B" is therefore used as the model identifier rather than as a dense-parameter-count characterization. For Qwen3 checkpoints, we use thinking mode (`enable_thinking=True`) consistently during routing, adaptation rollouts, and final evaluation. Instruction-tuned checkpoints use their corresponding tokenizer chat templates.

All experiments are performed on a single machine equipped with 8 NVIDIA RTX A6000 GPUs (48 GB GDDR6 each). For controlled comparisons, TTRL, EVOL-RL, and DiSCTT are evaluated under the same backbone checkpoints, adaptation datasets, high-level adaptation schedule, and evaluation protocol unless explicitly stated otherwise. Baseline methods retain their defining objectives and reward mechanisms, while shared implementation-level settings are aligned to facilitate comparison between adaptation strategies. Total adaptation compute is not constrained to be equal and is reported separately in Appendix F.

Adaptation is organized into $R = 5$ sequential training rounds. At the beginning of each round, the current policy generates $M = 8$ independent completions per input, and the consensus-based routing partition is recomputed using the fixed threshold $\rho = 45\%$. Because consensus lies on the $M = 8$ grid, this threshold corresponds to the effective cutoff $c \geq 4/8$ for SFT.

Each training round consists of 2 SFT epochs on $\mathbb{D}_{\text{conf}}$, followed by 8 RL epochs on $\mathbb{D}_{\text{unc}}$. Therefore, a complete adaptation run contains 10 SFT epochs and 40 RL epochs. After Round 5, we perform one additional routing refresh for final diagnostic analyses; because this refresh is performed after training and does not trigger parameter updates, it is not counted as a sixth adaptation round.

Optimization sequences for SFT and RL are capped at 2048 tokens. Generation and optimization lengths are accounted for separately: rollout and routing costs use the token counts actually processed during generation, whereas training costs use the token counts processed by the corresponding optimization forward/backward passes. We therefore do not identify the generation length with the 2048-token optimization limit. LoRA adaptation (Hu et al., 2022) is applied in both the SFT and RL phases, with rank $r_{\text{LoRA}} = 32$ and scaling factor $\alpha_{\text{LoRA}} = 64$. SFT uses batch size 8 and learning rate 1×10^{-5}. RL uses batch size 8, learning rate 1×10^{-6}, sampling temperature 0.9, and $N = 32$ sampled completions per input for each GRPO rollout group. The reward hyperparameters are fixed to $\alpha = 1$, $\beta = 1$, and $\varepsilon = 0.2$ across experiments.

For GRPO, advantages are normalized within each $N = 32$ rollout group using Eq. (10) with $\delta_{\text{adv}} = 10^{-8}$. We use clipping coefficient $\epsilon_{\text{clip}} = 0.2$ and KL coefficient $\beta_{\text{KL}} = 0.04$. Policy losses are length-normalized over the valid generated-response tokens of each trajectory before averaging across the rollout group. Each fresh rollout group is used for one optimization pass before new trajectories are sampled.

SFT and RL update the same LoRA adapter. LoRA is applied to the attention projections $\{\texttt{q_proj}, \texttt{k_proj}, \texttt{v_proj}, \texttt{o_proj}\}$ and MLP projections $\{\texttt{gate_proj}, \texttt{up_proj}, \texttt{down_proj}\}$. Optimization uses AdamW (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon_{\text{Adam}} = 10^{-8}$, zero weight decay, and gradient-norm clipping at 1.0. SFT and RL use learning rates of 10^{-5} and 10^{-6}, respectively. Separate optimizer states are maintained for the SFT and RL phases and preserved across training rounds.

For final evaluation, each adapted checkpoint is evaluated using one independently sampled generation per question with temperature 0.9, top-$p = 0.9$, and a maximum of 2048 new tokens. We report sampled Pass@1 final-answer accuracy. For the original six-benchmark suite, the reported mean and

sample standard deviation are computed across 10 independently adapted checkpoints with distinct training and sampling seeds.

For the relevance shaping factor g_{rel}, and for the semantic embeddings used by DiSCTT-OE, we use the pretrained `BAAI/bge-m3` sentence encoder (Chen et al., 2024). BGE-M3 is a multilingual embedding model, which allows the same encoder to be used for English-only reasoning tasks as well as the cross-lingual English–German comparisons in WMT24++.

The defaults above ($\rho = 45\%$, $M = 8$, $R = 5$, 2 SFT epochs per round, and 8 RL epochs per round) apply to the exact-match reasoning experiments. DiSCTT-OE instead uses $M_{\text{OE}} = 32$ routing samples per input. Its semantic-clustering configuration, including HDBSCAN parameters and fallback behavior, is reported in Appendix J.1. Sensitivity to the routing threshold, consensus sample size, and number of training rounds for the exact-match setting is reported in Appendices K.3–K.5. The results in Section 4 use the corresponding task-specific configurations unless explicitly stated otherwise.

B Routing Implementation Details

B.1 Answer extraction and tie-breaking

Before frequency counting, each generated trajectory $y_{j,i}$ is mapped by a deterministic benchmark-specific answer extractor $\mathcal{A}$ to a canonical final answer $a_{j,i} = \mathcal{A}(y_{j,i})$. The same extraction and canonicalization procedure is used consistently for routing, RL reward construction, and final-answer evaluation.

When multiple canonicalized answers attain the same maximum count, ties are resolved using the policy likelihood of their supporting trajectories. For each trajectory, we compute the length-normalized sequence log-probability

$$\ell_{j,i} = \frac{1}{L_{j,i}} \sum_{t=1}^{L_{j,i}} \log \pi_{\theta_{\text{roll}}}(y_{j,i,t} \mid x_j, y_{j,i,<t}), \tag{14}$$

where $L_{j,i}$ is the number of valid generated-response tokens. Among the tied answers, we select

$$a_{\text{route}}(x_j) = \arg \max_{a \in \mathcal{Z}_{\max}^{\text{route}}(x_j)} \frac{1}{N_a} \sum_{i: a_{j,i} = a} \ell_{j,i}, \tag{15}$$

where $\mathcal{Z}_{\max}^{\text{route}}(x_j)$ is the set of valid canonical answers attaining the maximum frequency among $i \in \mathcal{V}_j$, and N_a is the number of valid routing samples supporting answer a. Any residual exact tie in the mean length-normalized log-probability is resolved by a fixed lexicographic ordering of the canonical answer strings. The likelihood-based tie breaker is used only to obtain a deterministic support cluster and is not treated as additional evidence of answer correctness.

Malformed or unparseable generations are marked as invalid and cannot define or support a modal answer. They nevertheless remain in the sample pool and therefore in the denominator used to compute consensus, so malformed outputs reduce empirical agreement rather than being silently discarded.

Specifically, modal-answer selection is performed only over valid canonical answers. If no valid answer is produced in a routing sample pool, we define $c_j = 0$ and $a_{\text{route}}(x_j) = \varnothing$, and route the input to the RL branch. Invalid outputs never serve as SFT pseudo-targets.

Sequential or streaming inputs. DiSCTT does not require a batch of distinct queries. For an incoming input, the method can sample its M within-input trajectories, compute consensus, route the input, and apply or accumulate the corresponding update before later inputs arrive. A historical buffer of unrelated queries cannot replace these M trajectories because consensus is defined among alternative solutions to the *same* problem. The practical streaming limitation is therefore per-input sampling latency, not a requirement that multiple distinct test examples arrive together.

C ALGORITHM PSEUDOCODE

Algorithm 1 follows the training schedule used in the implementation. Adaptation consists of $R = 5$ training rounds. At the beginning of every round, the current policy generates $M = 8$ completions per input, recomputes consensus, and refreshes the routing partition. The model then performs 2 SFT epochs on $\mathbb{D}_{\text{conf}}$, followed by 8 RL epochs on $\mathbb{D}_{\text{unc}}$. Thus a complete run contains 10 SFT epochs and 40 RL epochs. After Round 5, one additional routing refresh is performed for diagnostic analysis only and does not trigger further parameter updates.

Algorithm 1 DiSCTT: reliability-guided objective allocation at test time

Require: Dataset $\mathcal{D}$, initial policy π_{θ_0}, rounds $R = 5$, consensus samples $M = 8$, GRPO group size $N = 32$, threshold ρ, SFT epochs $E_{\text{SFT}} = 2$, RL epochs $E_{\text{RL}} = 8$

$\theta \leftarrow \theta_0$
$\pi_{\text{ref}} \leftarrow \pi_{\theta_0}$ and keep π_{ref} frozen throughout all R rounds
for $r = 1, \ldots, R$ **do**
Routing refresh
  for each $x_j \in \mathcal{D}$ **do**
    Sample $\{y_{j,i} = (r_{j,i}, a_{j,i})\}_{i=1}^{M} \sim q_\theta(\cdot \mid x_j)$
    Identify the valid-answer index set $\mathcal{V}_j$
    Compute c_j using Eq. (1), excluding invalid answers from modal counting but retaining the full M denominator
    Determine $a_{\text{route}}(x_j)$ using the documented canonicalization, invalid-output, and tie-breaking rules
    if $c_j \geq \rho$ **then**
      Sample $y_j^\star \sim \text{Uniform}(\mathcal{C}_j)$ and cache it for the SFT phase of Round r
    end if
  end for
  $\mathbb{D}_{\text{conf}} \leftarrow \{x_j : c_j \geq \rho\}$
  $\mathbb{D}_{\text{unc}} \leftarrow \mathcal{D} \setminus \mathbb{D}_{\text{conf}}$
SFT phase
  if $\mathbb{D}_{\text{conf}} \neq \varnothing$ **then**
    for $e = 1, \ldots, E_{\text{SFT}}$ **do**
      Update θ over minibatches from $\mathbb{D}_{\text{conf}}$ using the SFT objective in Eq. (4)
    end for
  end if
RL phase
  if $\mathbb{D}_{\text{unc}} \neq \varnothing$ **then**
    for $e = 1, \ldots, E_{\text{RL}}$ **do**
      for each minibatch from $\mathbb{D}_{\text{unc}}$ **do**
        Sample $N = 32$ fresh rollout trajectories per input
        Determine $a_{\text{RL}}(x)$ from the fresh rollout group using the same canonicalization and tie-breaking rules
        Compute detached rewards using Eq. (6)
        Apply the GRPO update relative to π_{ref}
      end for
    end for
  end if
end for
Final diagnostic refresh
for each $x_j \in \mathcal{D}$ **do**
  Generate $M = 8$ fresh completions and recompute c_j and $a_{\text{route}}(x_j)$
end for
Use the final refresh only for post-training analysis
Evaluate π_θ using fresh generations
return θ

D GRPO Implementation Details

Sampled reference regularization. For reference regularization, we use the nonnegative sampled-token penalty

$$\kappa_{i,t}(\theta) = \exp\big(\ell^{\mathrm{ref}}_{i,t} - \ell^{\theta}_{i,t}\big) - \big(\ell^{\mathrm{ref}}_{i,t} - \ell^{\theta}_{i,t}\big) - 1, \tag{16}$$

$$\widehat{\mathcal{K}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{G}_i|}\sum_{t\in\mathcal{G}_i}\kappa_{i,t}(\theta), \tag{17}$$

Here $\ell^{\theta}_{i,t}$ and $\ell^{\mathrm{ref}}_{i,t}$ are raw-model log-probabilities of the sampled token. At a fixed prefix, $\mathbb{E}_{\pi_\theta}[\kappa_{i,t}] = D_{\mathrm{KL}}(\pi_\theta\|\pi_{\mathrm{ref}})$. Under transformed decoding, $\widehat{\mathcal{K}}$ is a nonnegative sampled reference regularizer, not an unbiased full-vocabulary KL estimate.

E Decision-Theoretic Motivation for Reliability-Guided Allocation

The empirical results motivate a basic question: when can routing different inputs to different adaptation objectives outperform applying one objective uniformly? We give a modest decision-theoretic answer. The analysis below deliberately avoids assuming a universal ordering of SFT and RL gradient variances, does not infer gradient statistics from final-accuracy standard deviations, and does not attempt to derive the numerical threshold $\rho = 45\%$.

The analysis is deliberately local. We fix a policy checkpoint, the training history leading to that checkpoint, and the SFT and RL update procedures together with their respective sampling and optimization budgets. The results therefore concern conditional objective allocation at a fixed state; they are not an end-to-end optimality theorem for multi-round DiSCTT.

Scope of the allocation comparison. An all-invalid routing pool has $C = 0$ and supplies no valid SFT target. We therefore compare the two branches conditional on $C > 0$. All probabilities and expectations in the theoretical analysis below refer to this conditional population. To compare full-population routers, extend every rule by $g(0) = \mathrm{RL}$. Writing $p_+ = \Pr_{\mathrm{full}}(C > 0)$, for $p_+ > 0$ and any two such rules,

$$\mathcal{R}_{\mathrm{full}}(g) - \mathcal{R}_{\mathrm{full}}(h) = p_+\big[\mathcal{R}(g) - \mathcal{R}(h)\big].$$

Thus the common zero-consensus contribution cancels. If $p_+ = 0$, all admissible routers use RL and their allocation regret is zero.

E.1 Conditional branch risks

Let C denote the routing statistic, here empirical trajectory consensus. For a specified evaluation loss and target distribution, define

$$R_{\mathrm{SFT}}(c) = \mathbb{E}[\mathcal{L}_{\mathrm{eval}} \,|\, C = c,\ \text{apply the prescribed SFT update}], \tag{18}$$

and

$$R_{\mathrm{RL}}(c) = \mathbb{E}[\mathcal{L}_{\mathrm{eval}} \,|\, C = c,\ \text{apply the prescribed RL update}]. \tag{19}$$

These are defined conditional risks of the two update procedures. We assume throughout that the expectations defining these conditional risks are finite. They are not assumed to equal gradient variances or pseudo-label error rates.

A routing rule

$$g(c) \in \{\mathrm{SFT}, \mathrm{RL}\}$$

has population risk

$$\mathcal{R}(g) = \mathbb{E}\big[R_{g(C)}(C)\big]. \tag{20}$$

Define the pointwise optimal selector

$$g^*(c) \in \arg\min_{a\in\{\mathrm{SFT},\mathrm{RL}\}} R_a(c). \tag{21}$$

Proposition 1 (conditional allocation regret). For any routing rule g,

$$\boxed{\mathcal{R}(g) - \mathcal{R}(g^*) = \mathbb{E}[|R_{\mathrm{SFT}}(C) - R_{\mathrm{RL}}(C)|\,\mathbf{1}[g(C) \neq g^*(C)]]\,.} \tag{22}$$

Proof. Fix any consensus value c. If $g(c)$ selects a risk-minimizing branch, its excess conditional risk is zero. Otherwise, because there are only two branches, the excess risk is exactly

$$|R_{\mathrm{SFT}}(c) - R_{\mathrm{RL}}(c)|\,.$$

Thus

$$R_{g(c)}(c) - R_{g^*(c)}(c) = |R_{\mathrm{SFT}}(c) - R_{\mathrm{RL}}(c)|\,\mathbf{1}[g(c) \neq g^*(c)].$$

Taking expectation over C proves Eq. (22). If the two risks are equal, the gap is zero regardless of tie-breaking. □

An immediate consequence is

$$\mathcal{R}(g^*) \leq \min\{\mathbb{E}[R_{\mathrm{SFT}}(C)], \mathbb{E}[R_{\mathrm{RL}}(C)]\}\,. \tag{23}$$

The inequality is strict relative to both uniform objectives whenever SFT has strictly lower conditional risk on a set of positive probability and RL has strictly lower conditional risk on another set of positive probability. This states precisely the sense in which heterogeneous objective suitability can make conditional allocation useful.

E.2 When does a threshold rule follow?

Proposition 1 does not by itself imply that the optimal selector is a threshold in consensus. A threshold follows only after an additional structural assumption on the conditional risk difference.

Let

$$\Delta(c) = R_{\mathrm{SFT}}(c) - R_{\mathrm{RL}}(c). \tag{24}$$

Proposition 2 (threshold structure under an explicit surrogate assumption). Suppose

$$\Delta(c) = a\,e(c) - b, \qquad a > 0, \quad b > 0, \tag{25}$$

where $e(c)$ is non-increasing in c. Then SFT is optimal whenever

$$e(c) \leq \frac{b}{a}. \tag{26}$$

Consequently, the set of consensus values for which SFT is optimal is upward closed. On the finite consensus grid induced by M samples, the optimal decision therefore admits a threshold form, allowing ties as well as the degenerate all-SFT and all-RL cases.

Proof. SFT is weakly preferable exactly when

$$\Delta(c) \leq 0,$$

which under Eq. (25) is equivalent to Eq. (26). Because $e(c)$ is non-increasing, once this inequality holds at some consensus value c, it also holds at every larger consensus value. Hence the SFT decision region is upward closed. On a finite ordered consensus grid, any upward-closed subset can be represented by a threshold rule. No observed consensus value is required to satisfy an exact equality between the two risks. □

An illustrative affine special case,

$$R_{\mathrm{SFT}}(c) = \lambda_{\mathrm{SFT}} e(c) + v_{\mathrm{SFT}}, \tag{27}$$

$$R_{\mathrm{RL}}(c) = \lambda_{\mathrm{RL}} e(c) + v_{\mathrm{RL}}, \tag{28}$$

is one special case of Eq. (25) when

$$a = \lambda_{\mathrm{SFT}} - \lambda_{\mathrm{RL}} > 0$$

and

$$b = v_{\mathrm{RL}} - v_{\mathrm{SFT}} > 0.$$

Under those explicit assumptions, the algebraic crossover condition is valid. However, we do not derive either inequality from the names of the SFT and RL objectives, and we do not estimate these quantities from final-accuracy standard deviations. In particular, the empirical results in this paper do not establish a universal ordering of SFT and RL gradient variances or identify a numerical crossover value.

The actual DiSCTT implementation also obtains self-generated modal answers from two different sample pools: routing and SFT target construction use the $M = 8$ routing samples, while every fresh $N = 32$ GRPO group defines its own $a_{\mathrm{RL}}(x)$. Therefore, the general result in Proposition 1 does not require the two branches to share a common pseudo-label error function. Equation (25) should be interpreted only as an additional surrogate assumption under which threshold structure follows.

E.3 Robustness to estimated conditional risks

In practice, conditional risks must be estimated from finite data. The following result bounds the cost of uniformly accurate risk estimation.

Let $\widehat{R}_{\mathrm{SFT}}(c)$ and $\widehat{R}_{\mathrm{RL}}(c)$ satisfy

$$\left|\widehat{R}_a(c) - R_a(c)\right| \le \epsilon \qquad \text{for } a \in \{\mathrm{SFT}, \mathrm{RL}\} \tag{29}$$

for every relevant consensus value c. Let

$$\widehat{g}(c) \in \arg\min_a \widehat{R}_a(c). \tag{30}$$

Proposition 3 (estimated-risk robustness). Under the uniform estimation condition above,

$$\boxed{\mathcal{R}(\widehat{g}) - \mathcal{R}(g^*) \le 2\epsilon.} \tag{31}$$

Proof. For every c,

$$R_{\widehat{g}(c)}(c) \le \widehat{R}_{\widehat{g}(c)}(c) + \epsilon \tag{32}$$

$$\le \widehat{R}_{g^*(c)}(c) + \epsilon \tag{33}$$

$$\le R_{g^*(c)}(c) + 2\epsilon. \tag{34}$$

Taking expectation over C yields Eq. (31). □

This result concerns estimation of conditional branch risks; it does not establish that empirical consensus itself is a calibrated estimate of answer correctness.

E.4 Finite-grid threshold misspecification

Conditional on $C > 0$, with $M = 8$, the routing statistic takes values on the discrete grid

$$\mathcal{C}_M = \left\{\frac{1}{8}, \frac{2}{8}, \ldots, 1\right\}.$$

Let

$$p_c = \Pr_{\mathrm{full}}(C = c \mid C > 0)$$

and define the conditional risk gap

$$\Delta(c) = R_{\mathrm{SFT}}(c) - R_{\mathrm{RL}}(c).$$

For any threshold router g_ρ, Proposition 1 gives the exact finite-grid regret

$$\mathcal{R}(g_\rho) - \mathcal{R}(g^*) = \sum_{c \in \mathcal{C}_M} p_c \, |\Delta(c)| \, \mathbf{1}[g_\rho(c) \ne g^*(c)]\,. \tag{35}$$

Thus threshold misspecification matters only on bins assigned to the wrong branch, weighted jointly by their probability mass and conditional risk gap.

This expression also clarifies the interpretation of the empirical threshold sensitivity in Table 18. The observed stability across nearby thresholds is an empirical property of the reported runs, not a theoretical consequence of the existence of a crossover. Moreover, because the routing statistic is discrete, different nominal thresholds can induce exactly the same routing rule; for example, with $M = 8$, $\rho = 45\%$ and $\rho = 50\%$ both correspond to the cutoff $c \ge 4/8$.

Table 4: Fixed-checkpoint comparison of SFT and RL updates conditioned on the initial routing consensus for Qwen3-1.7B-Base on MATH-500. Each consensus bin is treated independently: the SFT and RL variants for that bin are separately initialized from the same fixed checkpoint and trained on the same bin-specific examples. Results across bins therefore come from different updated model copies rather than a single jointly trained model. Δ is SFT accuracy minus RL accuracy in percentage points. Positive values favor SFT and negative values favor RL.

Consensus c	n_c	SFT correct	SFT Acc.	RL correct	RL Acc.	Δ SFT–RL
$1/8$	15	2	13.33	**3**	**20.00**	-6.67
$2/8$	53	11	20.75	**14**	**26.42**	-5.66
$3/8$	180	62	34.44	**72**	**40.00**	-5.56
$4/8$	75	**39**	**52.00**	38	50.67	$+1.33$
$5/8$	65	**40**	**61.54**	38	58.46	$+3.08$
$6/8$	50	**36**	**72.00**	35	70.00	$+2.00$
$7/8$	37	**32**	**86.49**	31	83.78	$+2.70$
$8/8$	25	**24**	**96.00**	23	92.00	$+4.00$

E.5 Empirical Conditional Branch Comparison

We next provide a local empirical diagnostic of whether the relative performance of the prescribed SFT and RL branch procedures changes with the routing statistic. We perform a fixed-checkpoint comparison on Qwen3-1.7B-Base using all 500 MATH-500 examples. Examples are grouped according to the consensus value obtained from the initial $M = 8$ routing samples.

Each occupied consensus bin is treated as a separate fixed-checkpoint intervention. For a given bin, the SFT and RL variants are independently initialized from the same model checkpoint and trained only on the examples belonging to that bin. The updated models for one consensus bin are therefore distinct from those used for every other bin, and the binwise results do not correspond to a single model jointly trained over all consensus levels.

The SFT branch uses the standard two-epoch SFT update, whereas the RL branch uses the standard eight-epoch GRPO update with $N = 32$ fresh rollouts per group, matching the branch-specific procedures and budgets used within one DiSCTT training round. The comparison therefore measures the relative effectiveness of these specified branch procedures at their respective operating budgets; it should not be interpreted as an intrinsic comparison between SFT and RL under equal computational or sampling resources. Both updated variants are evaluated using fresh final-answer generations.

For this empirical comparison, we instantiate the evaluation loss in the conditional-risk definition as binary final-answer error, $\mathcal{L}_{\text{eval}} = \mathbf{1}[\hat{a} \neq a^*]$. Consequently, lower empirical conditional risk is equivalent to higher final-answer accuracy.

This experiment is deliberately local: it compares the prescribed branch procedures from a common initial model state within each consensus bin. It does not reproduce the complete five-round DiSCTT trajectory and therefore should not be interpreted as an end-to-end causal estimate of the full curriculum.

The conditional comparison exhibits a clear change in branch preference across the consensus grid. RL performs better in every low-consensus bin $c \leq 3/8$, with gains of 5.56–6.67 percentage points over SFT. Beginning at $c = 4/8$, the sign reverses: SFT performs better in every observed bin from $4/8$ through $8/8$, with gains ranging from 1.33 to 4.00 points. Thus, the empirically preferred SFT region is upward closed on the observed consensus grid, consistent with the threshold structure in Proposition 2.

Aggregating the bins on either side of the DiSCTT cutoff gives the same qualitative pattern. For the 248 examples with $c \leq 3/8$, RL achieves 35.89% accuracy compared with 30.24% for SFT, a 5.65-point advantage. For the 252 examples with $c \geq 4/8$, SFT achieves 67.86% compared with 65.48% for RL, a 2.38-point advantage.

As a diagnostic summary, a post-hoc selector that takes the RL predictions from the independently updated low-consensus bin models and the SFT predictions from the independently updated high-

consensus bin models gives 52.0% aggregate accuracy. The corresponding composites that take the RL prediction in every bin and the SFT prediction in every bin give 50.8% and 49.2%, respectively. These quantities combine predictions from separately updated bin-specific models; they are not observed accuracies of a single jointly trained routed model.

The aggregate high-consensus difference is modest in absolute terms: SFT answers 171 of 252 examples correctly, compared with 165 of 252 for RL, corresponding to six additional correct examples. These marginal counts alone are insufficient for a paired significance test, because they do not specify how often SFT and RL disagree on the same examples. We therefore do not claim that this diagnostic statistically establishes the routing boundary.

These results provide preliminary local evidence for heterogeneous objective suitability under the prescribed branch procedures, but they should not be interpreted as proving that $4/8$ is a uniquely optimal or statistically established threshold. The comparison uses one model, one dataset, and one fixed checkpoint. Moreover, the difference at $c = 4/8$ corresponds to only one additional correct example for SFT (39 versus 38), and several extreme-consensus bins contain relatively few examples. The experiment therefore supports the qualitative possibility of a sign-changing branch preference across consensus, consistent with Proposition 2 under its stated assumptions, rather than establishing a universal threshold law across models, datasets, or training budgets.

The experiment also illustrates Proposition 1. Uniform SFT and uniform RL each select the less effective prescribed branch procedure in bins where the competing procedure performs better. Conditional allocation can avoid such bin-specific mismatches, with the potential gain determined jointly by the branch-performance gap and the number of examples in each bin. This fixed-checkpoint diagnostic therefore provides preliminary local evidence for the allocation principle, but does not establish global optimality of the full multi-round DiSCTT procedure.

E.6 Scope of the analysis

The propositions above are deliberately local. They hold at the fixed state at which the conditional branch risks are defined. In DiSCTT, shared parameter updates alter future policy states, future consensus values, and the behavior of other examples. The results therefore do not:

- prove that the complete multi-round DiSCTT trajectory is globally optimal;
- establish that $\rho = 45\%$ is theoretically optimal;
- establish a universal SFT-versus-RL gradient-variance ordering;
- imply that confidently-wrong examples are corrected specifically by the RL branch; or
- predict a universal direction in which the optimal threshold must move as the policy changes.

The transition analysis in Section 4.3 should therefore be interpreted as observational evidence that some confidently-wrong states disappear under shared-policy updates and re-routing, not as an intervention that identifies which update objective caused the transition.

A cost-sensitive extension is immediate. If $C_{\mathrm{SFT}}(c)$ and $C_{\mathrm{RL}}(c)$ denote conditional adaptation costs, one may replace each branch risk with

$$\widetilde{R}_a(c) = R_a(c) + \lambda_{\mathrm{cost}} C_a(c). \tag{36}$$

The same allocation-regret identity then applies to the combined loss–compute objective. The pointwise optimal selector is determined by the corresponding combined conditional risks rather than by prediction loss alone. A consensus-threshold representation for this cost-sensitive selector would require an additional structural assumption on the combined risk difference, analogous to the monotonicity assumption used in Proposition 2.

F FLOP Accounting Methodology

We report parameter-count-based FLOP estimates rather than exact hardware FLOPs. Following the standard parameter-count approximation (Kaplan et al., 2020), a model forward pass costs approximately $2N_p$ FLOPs per processed token, where N_p is the model parameter count, while a training forward+backward pass costs approximately $6N_p$ FLOPs per processed token.

The accounting is event based. Inference-time rollout generation does not retain the differentiable computation graph used for optimization; therefore rollout generation and the subsequent policy forward+backward pass are counted separately. We also distinguish routing generation, SFT optimization, reference-policy scoring, and encoder evaluation. Token counts refer to all tokens actually processed by the corresponding model call, including prompt tokens even when their labels are masked.

For training round r, let $T^{(r)}_{\text{route}}$ denote the tokens processed during routing generation, $T^{(r)}_{\text{SFT}}$ the tokens processed by SFT forward/backward, $T^{(r)}_{\text{gen}}$ the tokens processed during RL rollout generation, $T^{(r)}_{\text{train}}$ the tokens processed by the differentiable policy optimization pass, $T^{(r)}_{\text{ref}}$ the tokens processed by a separately executed reference-policy forward pass, and $T^{(r)}_{\text{old}}$ the tokens processed by a separately executed old-policy scoring pass, when such a pass is used.

Routing refresh. At the beginning of each of the $R = 5$ training rounds, DiSCTT generates $M = 8$ completions per input to recompute consensus and routing. The corresponding round-r cost is

$$C^{(r)}_{\text{route}} \approx 2N_p T^{(r)}_{\text{route}}. \tag{37}$$

Supervised fine-tuning. For inputs assigned to $\mathbb{D}^{(r)}_{\text{conf}}$, SFT contributes

$$C^{(r)}_{\text{SFT}} \approx 6N_p T^{(r)}_{\text{SFT}}. \tag{38}$$

Here $T^{(r)}_{\text{SFT}}$ counts all tokens processed by the training forward/backward pass. Loss masking of prompt tokens removes their loss contribution but not the computation required to process them.

Reinforcement learning. For inputs assigned to $\mathbb{D}^{(r)}_{\text{unc}}$, rollout generation and policy optimization are distinct events. The basic GRPO cost is

$$C^{(r)}_{\text{RL,basic}} \approx \underbrace{2N_p T^{(r)}_{\text{gen}}}_{\text{rollout generation}} + \underbrace{6N_p T^{(r)}_{\text{train}}}_{\text{policy forward+backward}}. \tag{39}$$

If every generated token is subsequently optimized once and $T^{(r)}_{\text{gen}} = T^{(r)}_{\text{train}} = T^{(r)}$, the basic coefficient is therefore $8N_p T^{(r)}$.

The frozen reference policy is evaluated separately when required for the reference-regularization term.

$$C^{(r)}_{\text{RL}} \approx C^{(r)}_{\text{RL,basic}} + 2N_p T^{(r)}_{\text{ref}} + 2N_p T^{(r)}_{\text{old}}, \tag{40}$$

where a term is set to zero when the corresponding log-probabilities are reused from another executed model pass. In our implementation, no additional old-policy forward pass is executed separately, so $T^{(r)}_{\text{old}} = 0$. Thus, when rollout generation, policy training, and reference-policy evaluation process the same number of tokens, the corresponding coefficient is approximately $10N_p$ per token-event.

Embedding-encoder cost. DiSCTT uses `BAAI/bge-m3` for its relevance term and for the semantic operations in DiSCTT-OE, while EVOL-RL uses its reported embedding model for semantic novelty computation (Zhou et al., 2025). Encoder size alone does not determine relative overhead: actual cost also depends on the number and length of encoded trajectories, batching, and hardware utilization. We therefore account for encoder computation using the executed encoder calls and processed-token counts. Let $T^{(r)}_{\text{embed}}$ denote the total number of tokens processed by the encoder during round r. We approximate

$$C^{(r)}_{\text{embed}} \approx 2N_e T^{(r)}_{\text{embed}}, \tag{41}$$

where N_e is the parameter count of the encoder actually used by the corresponding method.

Dynamic event accounting. Because the routing partition changes between rounds, DiSCTT is not approximated using a single static $|\mathbb{D}_{\text{conf}}|$ or $|\mathbb{D}_{\text{unc}}|$. Instead, adaptation cost is accumulated over the actual sequence of model events:

$$C_{\text{adapt}} = \sum_{r=1}^{R} \left[C^{(r)}_{\text{route}} + C^{(r)}_{\text{SFT}} + C^{(r)}_{\text{RL}} + C^{(r)}_{\text{embed}} \right]. \tag{42}$$

Table 5: Estimated adaptation FLOPs under matched settings using the event/token accounting in Appendix F. The estimates include routing generation, SFT optimization, RL rollout generation, differentiable policy optimization, reference- and old-policy scoring when separately executed, and encoder calls. Final diagnostic routing and benchmark evaluation are excluded. Values are analytical parameter-count-based estimates rather than measured hardware FLOPs. Percentages use unrounded totals.

Configuration	TTRL	EVOL-RL	DiSCTT	vs. TTRL	vs. EVOL
Llama-3.2-1B / MMLU	406.7e18	480.9e18	**156.0e18**	−61.6%	−67.6%
Qwen3-4B / MATH-500	154.3e18	156.9e18	**68.3e18**	−55.7%	−56.5%
Qwen2.5-7B / AMC	66.3e18	66.5e18	**31.3e18**	−52.8%	−52.9%

The SFT and RL token counts therefore automatically reflect the round-specific partitions $\mathbb{D}_{\text{conf}}^{(r)}$ and $\mathbb{D}_{\text{unc}}^{(r)}$.

For TTRL, the RL events are accumulated over the full dataset and the routing and SFT terms are absent. For EVOL-RL, RL likewise operates on the full dataset, with its embedding-model evaluations included as separate events.

Adaptation versus evaluation. The post-training diagnostic routing refresh and final benchmark evaluation are kept separate from adaptation:

$$C_{\text{end-to-end}} = C_{\text{adapt}} + C_{\text{diag}} + C_{\text{eval}}. \tag{43}$$

Table 5 reports C_{adapt}, so all methods are compared over their adaptation procedures rather than over post-training benchmark evaluation.

Measured quantities and approximation scope. The numbers of routing generations, SFT optimization events, RL rollout groups, optimization passes, separately executed reference-policy passes, and encoder calls are taken from the executed training procedure. Their processed-token counts are accumulated separately. The conversion from these token-event counts to FLOPs using the $2N_p$ and $6N_p$ coefficients is an analytical approximation.

All experiments use LoRA adaptation, but the estimates above are parameter-count-based full-model-equivalent proxies rather than exact LoRA hardware FLOPs. LoRA, activation recomputation, padding, attention cost, frozen-backbone gradient paths, communication, and kernel utilization can affect actual runtime differently across SFT-heavy and RL-heavy methods. We therefore do not assume that these effects cancel exactly in FLOP ratios; measured wall-clock results are reported independently in Appendix G.

F.1 Compute Efficiency

We estimate adaptation cost using event- and token-level accounting (Appendix F). Rollout generation, differentiable policy optimization, routing generation, reference-policy scoring when separately executed, and encoder calls are treated as distinct model events. In particular, inference-time rollout generation is not treated as the differentiable forward pass used for GRPO optimization. The reported FLOPs are therefore parameter-count-based analytical estimates derived from the executed model events and associated token counts; measured wall-clock time is reported independently.

Across the three representative settings in Table 5, DiSCTT reduces the estimated adaptation FLOPs by 52.8–61.6% relative to TTRL and by 52.9–67.6% relative to EVOL-RL. The dominant savings arise because the expensive rollout-based RL component is applied only to the dynamically varying uncertain subset $\mathbb{D}_{\text{unc}}^{(r)}$ rather than to the full dataset. Total DiSCTT cost does not scale solely with $|\mathbb{D}_{\text{unc}}|$: it also includes full-dataset routing refreshes, SFT updates on $\mathbb{D}_{\text{conf}}^{(r)}$, and auxiliary encoder computations.

Measured wall-clock reductions on the same configurations are 40–49% (Appendix G) and are reported as an independent systems-level efficiency measure.

This comparison should not be misread as claiming that DiSCTT is cheaper than pure supervised self-training. Uniform SFT is much cheaper because it avoids rollout-based RL almost entirely, but it also produces substantially smaller gains. Appendix H quantifies this tradeoff on AIME-24 at 4B and 32B. The efficiency claim is therefore a comparison against full-dataset RL adaptation, not a claim of minimum possible adaptation cost.

G Wall-Clock Time Measurements

The analytical FLOP estimates and measured runtime characterize different aspects of efficiency. FLOP estimates are hardware-independent parameter-count-based proxies, whereas elapsed time also depends on batching, communication, memory traffic, kernel efficiency, padding, and other systems effects. We therefore report measured wall-clock time on identical hardware (8× NVIDIA RTX A6000) as a separate empirical efficiency measure rather than as validation of the analytical FLOP proxy.

Table 6: Measured wall-clock training time for DiSCTT and TTRL under matched hardware and software. The measured reductions are 40–49%, while the event/token-based analytical FLOP estimates in Table 5 give reductions of 52.8–61.6% relative to TTRL. The two measures are reported independently.

Model	Method	Data	SFT	RL	Total	Time Ratio	Reduction
Llama-3.2-1B Instruct	DiSCTT	MMLU	134m	121h42m	123h56m	0.51	48.7%
	TTRL	MMLU	0	241h23m	241h23m		
Qwen3-4B Base	DiSCTT	MATH-500	21m	51h48m	52h09m	0.60	40.3%
	TTRL	MATH-500	0	87h23m	87h23m		
Qwen2.5-7B Instruct	DiSCTT	AMC	6m	15h29m	15h35m	0.52	47.9%
	TTRL	AMC	0	29h53m	29h53m		

DiSCTT reduces measured end-to-end training time in all three configurations. The wall-clock reductions are smaller than the analytical FLOP reductions because runtime also contains fixed costs, communication, data movement, padding, kernel-utilization effects, and other systems overheads that are not represented by the parameter-count FLOP proxy. We therefore interpret the FLOP and wall-clock results as complementary efficiency measurements rather than using one to validate the other.

H Pure-SFT Accuracy–Compute Tradeoff

Because RL is substantially more expensive than supervised fine-tuning, a reviewer-relevant comparison is pure SFT rather than only RL-based baselines. Table 7 reports this tradeoff on AIME-2024.

For the uniform-SFT control, modal-answer-supported pseudo-targets are generated once before training using $M = 8$ samples per input and are then held fixed throughout the 10 SFT epochs. Its adaptation cost therefore contains both the pseudo-target generation stage and the SFT optimization stage:

$$C_{\text{pureSFT}} = 2N_pT_{\text{pseudo}} + 6N_pT_{\text{SFT}}. \tag{44}$$

Under the reported AIME-2024 configuration, this gives approximately 0.0187×10^{18} FLOPs for Qwen3-4B and 0.149×10^{18} FLOPs for Qwen3-32B.

Relative to the corresponding base accuracies (10.0 at 4B and 36.7 at 32B), uniform SFT gains only 1.7 and 5.0 points, whereas DiSCTT gains 13.7 and 18.0 points. Thus DiSCTT is 12.0 and 13.0 points more accurate than uniform SFT in these settings, but at substantially higher compute cost. This is why our efficiency claim is scoped to full-dataset RL baselines: DiSCTT improves the accuracy–compute tradeoff relative to uniform RL, while pure SFT occupies a cheaper but lower-accuracy operating point.

Table 7: Uniform SFT versus DiSCTT on AIME-2024. Uniform-SFT FLOPs include the one-time $M = 8$ pseudo-target-generation stage and the subsequent 10 SFT epochs. DiSCTT FLOPs use the event/token accounting described in Appendix F.

Model	Method	Accuracy	Estimated FLOPs	Wall-clock
Qwen3-4B	Uniform SFT (10 ep.)	11.7	0.0187e18	29m
	DiSCTT	**23.7**	**2.63e18**	8h39m
Qwen3-32B	Uniform SFT (10 ep.)	41.7	0.149e18	1h24m
	DiSCTT	**54.7**	**15.45e18**	46h52m

Table 8: Modal-answer correctness conditioned on empirical consensus for Qwen3-1.7B-Base. Each consensus-bin entry reports modal-answer accuracy (%) with the number of observations in parentheses. The final column reports pseudo-label precision within the high-consensus subset $c \geq 4/8$, corresponding to the effective SFT routing threshold. Ground-truth labels are used only for this post-hoc analysis. Unoccupied bins are denoted by "–".

Checkpoint	Dataset	1/8	2/8	3/8	4/8	5/8	6/8	7/8	8/8	Precision ($c \geq 4/8$)
Initial	AMC	–	0.0 (5)	5.0 (20)	33.3 (6)	75.0 (4)	100.0 (3)	100.0 (1)	100.0 (1)	66.7
Final	AMC	–	0.0 (3)	23.5 (17)	85.7 (7)	100.0 (5)	100.0 (4)	100.0 (2)	100.0 (2)	95.0
Initial	MATH-500	6.7 (15)	18.9 (53)	32.8 (180)	61.3 (75)	72.3 (65)	82.0 (50)	91.9 (37)	96.0 (25)	76.2
Final	MATH-500	12.5 (8)	19.0 (42)	29.0 (100)	83.1 (65)	90.7 (75)	95.0 (80)	98.6 (70)	100.0 (60)	93.4
Initial	AIME-2024	0.0 (5)	0.0 (9)	0.0 (12)	33.3 (3)	100.0 (1)	–	–	–	50.0
Final	AIME-2024	0.0 (3)	12.5 (8)	28.6 (14)	50.0 (2)	100.0 (1)	100.0 (1)	100.0 (1)	–	80.0
Initial	GPQA-D	–	0.0 (12)	5.8 (104)	52.8 (36)	65.0 (20)	84.6 (13)	100.0 (8)	100.0 (5)	68.3
Final	GPQA-D	–	0.0 (7)	6.0 (100)	78.1 (32)	91.3 (23)	93.8 (16)	100.0 (12)	100.0 (8)	89.0

I Additional Experimental Results

I.1 Consensus Reliability Analysis

We directly test whether empirical trajectory agreement is informative about modal-answer correctness. For each example, let k denote the number of the $M = 8$ sampled trajectories that agree with the modal answer, so that $c = k/8$. For every occupied consensus bin, we compute

$$\mathrm{Acc}(k/8) = \Pr(a_{\mathrm{route}} = a^* \mid c = k/8).$$

Ground-truth labels are used only for this post-hoc analysis and never for adaptation, routing, or threshold selection.

Table 8 directly supports the use of consensus as a relative reliability signal. Across all four benchmarks, modal-answer correctness increases monotonically across the occupied consensus bins at both the initial and final checkpoints. The trend is particularly well-supported on MATH-500 and GPQA-Diamond, where individual bins contain substantial numbers of examples. On AIME-2024, several high-consensus bins contain only one or two examples and should therefore be interpreted cautiously despite exhibiting the same qualitative trend.

The reliability of the routed high-consensus subset also improves during adaptation: pseudo-label precision increases from 66.7% to 95.0% on AMC, from 76.2% to 93.4% on MATH-500, from 50.0% to 80.0% on AIME-2024, and from 68.3% to 89.0% on GPQA-Diamond. These observations support

consensus as an empirical reliability proxy while not implying that high-consensus predictions are universally correct.

I.2 Routing Distribution Analysis

A central feature of DiSCTT is that the SFT/RL split is not fixed in advance: the consensus partition is recomputed at the beginning of each training round, allowing instances to migrate between $\mathbb{D}_{\text{conf}}$ and $\mathbb{D}_{\text{unc}}$ as the policy evolves. This appendix quantifies routing at the five training-time refreshes, $r = 1, \ldots, 5$. Each refresh occurs at the beginning of a training round and is followed by 2 SFT epochs and 8 RL epochs, so the five rounds contain 10 SFT and 40 RL epochs in total. A separate post-training diagnostic refresh is used for final reliability analysis and is not counted as an adaptation round.

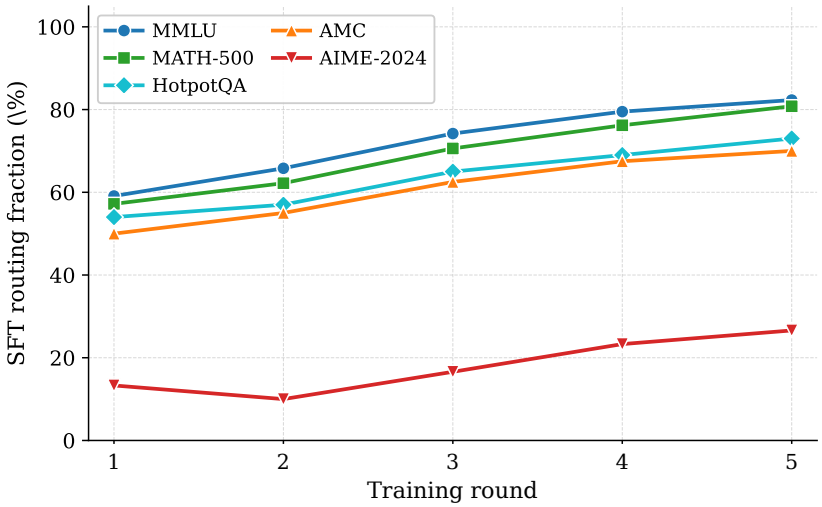


Figure 3: Evolution of the SFT routing fraction during DiSCTT adaptation on Qwen3-4B-Base. The fraction of inputs routed to supervised fine-tuning increases monotonically across training rounds for MMLU, MATH-500, HotpotQA, and AMC, reflecting consensus solidification as the policy improves. AIME-24 shows a non-monotonic dip at Round 2 before recovering, demonstrating that routing assignments can move in both directions rather than monotonically accumulating in the SFT branch. The Round-5 ordering MMLU $>$ MATH-500 $>$ HotpotQA $>$ AMC $>$ AIME-24 mirrors benchmark difficulty.

Aggregate routing distribution. Figure 3 shows the SFT routing fraction at the five training-time routing refreshes, measured immediately before optimization in each corresponding round. Table 9 summarizes these trajectories as the range of SFT and RL fractions observed across the five training rounds. Four of the five benchmarks (MMLU, MATH-500, HotpotQA, AMC) route predominantly to SFT by the end of adaptation, with final SFT fractions in the 70–82% range. AIME-2024 is qualitatively different: it routes the majority of inputs to RL throughout adaptation ($\geq 73\%$ across all training rounds), reflecting its substantial difficulty for the base model.

These routing percentages describe the fraction of *inputs* assigned to each objective, not the fraction of adaptation FLOPs. Because each RL input requires $N = 32$ rollout trajectories and multiple model evaluations, the RL branch can dominate adaptation cost even when it contains a minority of the inputs. The expensive RL component scales with the dynamically varying uncertain subset, whereas total DiSCTT cost additionally includes full-dataset routing refreshes, SFT updates, and auxiliary computations.

Three regimes of routing dynamics. The five trajectories in Figure 3 fall into three distinct regimes that reflect benchmark structure rather than dataset-specific tuning:

Table 9: Routing distributions across datasets, reported as the range of values observed across the five training rounds. Easier datasets exhibit higher SFT fractions, while harder datasets retain more inputs in RL. AIME-24 in particular routes the majority of its inputs to RL throughout adaptation, reflecting its substantial difficulty for the base model.

Dataset	SFT (%)	RL (%)
MMLU	59–82	18–41
MATH-500	57–81	19–43
Hotpot	54–73	27–46
AMC	50–70	30–50
AIME-24	10–27	73–90

- *Saturating regime* (MMLU, MATH-500): the SFT fraction starts above 55%, rises steeply through Round 3, and plateaus near 80% by Round 5. These benchmarks have inputs where the base model already produces a stable solution mode; adaptation reinforces these modes rapidly, and most remaining uncertain inputs solidify into high-consensus by mid-training.
- *Steady-rise regime* (HotpotQA, AMC): the SFT fraction starts near 50%, rises gradually but persistently across all five training rounds, and ends in the 70–73% range. These benchmarks contain a balanced mix of confidently-solved and uncertain inputs at the start; consensus solidification proceeds at a steadier pace because each round's RL phase exposes the model to genuinely novel reasoning patterns rather than refining nearly-known answers.
- *Non-monotonic regime* (AIME-2024): the SFT fraction starts low (13.3%), drops to 10.0% at Round 2, and then increases to 26.7% by Round 5. This pattern shows that the evolving policy can move examples in both directions across the routing threshold. Because these aggregate routing fractions do not track the causal update path of individual examples, we do not attribute the temporary decrease to a specific SFT or RL mechanism. Instead, it demonstrates that the routing signal can evolve non-monotonically as the shared policy changes.

Cross-round interpretation. The routing trajectories show two descriptive properties. First, the fraction of inputs assigned to SFT differs substantially across benchmarks despite shared hyperparameters, demonstrating that the partition is data dependent. Second, routing assignments change as the shared policy is updated: on several benchmarks, progressively more inputs satisfy the high-consensus criterion, while AIME-24 shows a small temporary decrease before increasing later. These observations are consistent with the intended dynamic-curriculum interpretation, but the aggregate routing fractions alone do not establish that increased consensus reflects increased competence or identify which preceding update caused an individual transition.

Connection to the routing ablation. Static routing fixes the Round-1 assignments for the full adaptation run, whereas dynamic DiSCTT recomputes them at every round. The $+4.2$-point difference between dynamic and static routing in Table 10 is consistent with a benefit from recomputing the partition under the evolving policy. Because the two training trajectories subsequently differ, this aggregate comparison does not identify the causal update path of individual examples.

I.3 Routing Strategy Ablation

We compare DiSCTT against four controls to examine objective-allocation strategies under the same experimental settings. All routing controls use the same five-round schedule and optimization hyperparameters. Random routing matches DiSCTT's per-round SFT/RL input counts, whereas static consensus routing freezes the initial per-input branch assignments throughout adaptation. Uniform SFT and uniform RL serve as single-objective endpoint controls.

Ablation baselines.

- **Uniform SFT.** All inputs receive supervised fine-tuning on modal-answer pseudo-labels (no RL).
- **Uniform RL (DiSCTT reward).** All inputs receive reinforcement learning using the full DiSCTT reward in Eq. (6), with no SFT branch. This control follows the same all-RL allocation pattern as

TTRL, but retains DiSCTT's modal-consistency, novelty, and relevance reward shaping. It therefore provides an all-RL comparison while holding DiSCTT's RL objective fixed.

- **Random routing.** At the beginning of each training round, inputs are randomly reassigned to SFT or RL while matching DiSCTT's per-round SFT/RL input counts. The random partition is regenerated independently at every round. All other training settings are unchanged.
- **Static consensus routing.** At the beginning of training, we compute the consensus partition exactly as in DiSCTT using $M = 8$ sampled completions and the same threshold ρ. The resulting per-input assignments are then frozen for the entire adaptation run: an input initially assigned to SFT remains in the SFT branch in every subsequent round, and an input initially assigned to RL remains in the RL branch. In contrast, DiSCTT recomputes the partition under the evolving policy at the beginning of every training round. This control compares fixed and dynamically recomputed consensus routing under the reported configuration.

Table 10: Effect of routing strategy on LLaMA-3.2-3B-Instruct under the same five-round schedule and optimization hyperparameters. Random routing additionally matches DiSCTT's per-round SFT/RL allocation counts. Random routing performs no better than uniform RL under these configurations, while static and dynamic consensus routing perform better. Static consensus routing captures most of the benefit; dynamic re-routing adds further improvement.

Routing Strategy	**MATH-500**	**AIME-2024**	**Avg. Δ vs. Base**
Base	35.7	3.7	0.0
Uniform SFT	43.5	3.7	$+3.9$
Uniform RL	45.4	9.7	$+7.9$
Random routing	44.9	7.3	$+6.4$
Static consensus routing	51.2	12.3	$+12.1$
DiSCTT (dynamic routing)	**55.6**	**16.3**	$+16.3$

Findings. The results decompose the gain hierarchically:

- **Hybridization is not enough.** Random routing performs no better than uniform RL ($+6.4$ vs. $+7.9$), showing that simply combining SFT and RL without uncertainty-aware allocation does not recover the DiSCTT gain.
- **Consensus-based allocation contributes substantially.** Static consensus routing achieves $+12.1$, a $+5.7$ point improvement over random routing under the same training schedule and optimization hyperparameters. Unlike random routing, static routing preserves the initial consensus-based assignment throughout training.
- **Dynamic re-routing provides further improvement.** DiSCTT's dynamic re-routing adds another $+4.2$ points over static routing, providing evidence that periodically recomputing the partition contributes beyond the initial consensus-based allocation.

The hierarchy random $<$ static $<$ dynamic provides controlled evidence that consensus-based allocation and periodic reassignment contribute to the observed gains beyond simply combining SFT and RL. Per-difficulty and held-out transfer results provide complementary evidence that the improvements are not confined to a single benchmark or difficulty regime.

I.4 Control: Majority Voting at Matched Generation Budget

A natural concern is whether DiSCTT's gains stem from the additional sampling that consensus estimation requires, since majority voting (MV) is itself a known test-time scaling method (Wang et al., 2023): aggregating multiple sampled answers and returning the most frequent one improves accuracy without any training. If DiSCTT's improvements were primarily a sampling artifact, matching the sampling budget under MV should recover most of the gain.

Setup. We compare DiSCTT against MV under a matched generation/forward-pass sampling budget. The MV control receives the same forward-generation budget per prompt used in the

corresponding sampling control, but performs no parameter updates. Backward optimization cost is therefore not converted into additional MV samples, and this comparison should not be interpreted as matching total training FLOPs. Both methods use identical decoding settings (temperature, top-p, maximum generation length) and the same chat template per model. This control tests whether additional sampling alone can account for the observed gains.

Table 11: DiSCTT vs. majority voting (MV) under a matched generation/forward-pass sampling budget. Across both models and all three benchmarks, MV yields modest or inconsistent improvements over the base, while DiSCTT yields substantially larger gains. The gap is most pronounced on harder benchmarks (AIME-2024, AMC), where additional sampling alone provides little benefit and the adaptation signal does the work.

Method	AMC	MATH-500	AIME-24
Qwen3-1.7B-Base			
Base	19.8	47.0	6.7
MV	27.5	52.4	6.7
DiSCTT	**48.8**	**67.0**	**26.7**
Qwen-2.5-7B-Instruct			
Base	39.7	58.8	10.7
MV	42.3	64.8	10.0
DiSCTT	**59.5**	**82.2**	**29.7**

Consistency with the routing diagnostic. For Qwen3-1.7B-Base, the majority-vote accuracies in Table 11 are consistent with the initial routing-time modal-answer statistics in Table 8. Aggregating the initial consensus bins in Table 8 gives modal-answer accuracies of 27.5% on AMC, 52.4% on MATH-500, and 6.7% on AIME-2024, exactly matching the corresponding MV results in Table 11.

Findings. DiSCTT outperforms the reported MV configuration in all six comparisons. These results do not establish an equal-total-compute comparison.

I.5 High-Consensus Coverage and Precision

A decrease in confidently-wrong examples does not by itself imply that the retained pseudo-labels have become more reliable: the joint error could also decrease simply because fewer examples are routed to SFT. We therefore separate selection coverage from conditional pseudo-label reliability. For $\mathbb{D}_{\text{conf}} = \{x : c(x) \geq \rho\}$, define

$$\text{Coverage} = \frac{|\mathbb{D}_{\text{conf}}|}{|\mathcal{D}|}, \tag{45}$$

$$\text{Precision} = \frac{|\{x \in \mathbb{D}_{\text{conf}} : a_{\text{route}}(x) \neq a^*(x)\}|}{|\mathbb{D}_{\text{conf}}|}, \tag{46}$$

$$\text{CondErr} = 1 - \text{Precision}, \tag{47}$$

$$\text{JointWrong} = \frac{|\{x \in \mathbb{D}_{\text{conf}} : a_{\text{route}}(x) \neq a^*(x).\}|}{|\mathcal{D}|}. \tag{48}$$

By construction,

$$\text{JointWrong} = \text{Coverage}\,(1 - \text{Precision}).$$

The decomposition shows that the reduction in confidently-wrong assignments is not caused by a shrinking selected set. Coverage increases on every benchmark, while pseudo-label precision also rises substantially: from 66.7% to 95.0% on AMC, 76.2% to 93.4% on MATH-500, 50.0% to 80.0% on AIME-2024, and 68.3% to 89.0% on GPQA-Diamond. Consequently, conditional error decreases even as more examples qualify for supervised consolidation. This provides stronger evidence that the high-consensus subset becomes more reliable during adaptation rather than merely more selective.

Table 12: Reliability decomposition of the high-consensus subset for Qwen3-1.7B-Base. Coverage is the fraction of inputs routed to SFT; precision is modal-answer correctness within that subset; conditional error is $1 - \mathrm{Precision}$; and JointWrong is the fraction of all inputs that are both high-consensus and incorrect. All quantities are computed from the same checkpoint and sample pool.

Dataset	Checkpoint	Coverage (%)	Precision (%)	Cond. Error (%)	Joint Wrong (%)
AMC	Initial	37.5	66.7	33.3	12.5
	Final	50.0	95.0	5.0	2.5
MATH-500	Initial	50.4	76.2	23.8	12.0
	Final	70.0	93.4	6.6	4.6
AIME-2024	Initial	13.3	50.0	50.0	6.7
	Final	16.7	80.0	20.0	3.3
GPQA-D	Initial	41.4	68.3	31.7	13.1
	Final	46.0	89.0	11.0	5.1

I.6 Hard Routing versus Applying Both Objectives

Hard routing is intended to avoid imitation on unresolved examples and unnecessary RL on stable ones. We test whether simply giving every input both objectives recovers the same benefit.

Sequential SFT+RL. The first control applies SFT and then RL uniformly to every input.

Table 13: Sequential SFT+RL versus DiSCTT on three reasoning tasks.

Model	Method	AMC	AIME-24	GPQA-D	Avg.
Qwen3-1.7B	Base	19.8	6.7	23.2	16.6
	Sequential SFT+RL	44.3	18.7	25.6	29.5
	DiSCTT	**48.8**	**26.7**	**29.3**	**34.9**
Qwen3-4B	Base	38.5	10.0	27.3	25.3
	Sequential SFT+RL	53.0	17.3	32.8	34.4
	DiSCTT	**57.0**	**23.7**	**38.4**	**39.7**

DiSCTT exceeds sequential SFT+RL by 5.4 and 5.3 average points, with the largest gaps on AIME-2024, where pseudo-label reliability is lowest.

LF-SRFT: label-free adaptation of SRFT. The original SRFT framework (Fu et al., 2026) jointly optimizes three sources of learning signal within a single training stage: (i) entropy-weighted supervised learning on a demonstration, (ii) off-policy reinforcement learning on that demonstration, with the demonstration included in the augmented group used for group-relative normalization, and (iii) reinforcement learning on self-generated rollouts with entropy-dependent weighting of positive trajectories.

Because our test-time setting provides neither external demonstrations nor ground-truth answers, these sources of supervision cannot be used unchanged. We therefore construct *LF-SRFT*, a label-free adaptation that changes the source of supervision while retaining the SRFT optimization structure.

For every input x, we first generate the same $M = 8$ samples used for label-free pseudo-supervision in DiSCTT and identify the routing-time modal answer $a_{\mathrm{route}}(x)$ using the same answer extraction, canonicalization, invalid-output handling, and tie-breaking rules. One trajectory whose final answer matches $a_{\mathrm{route}}(x)$ is selected as the pseudo-demonstration y^{pseudo}. Unlike DiSCTT, this pseudo-demonstration is constructed for every input because LF-SRFT applies the combined SFT/RL objective uniformly rather than routing examples to exclusive branches.

The external demonstration in the original SRFT supervised term is then replaced by y^{pseudo}. We retain SRFT's original entropy-dependent supervised weight,

$$w_{\text{SFT}} = 0.5 \ \text{sg}\Big(e^{-H(\pi_\theta)}\Big), \tag{49}$$

where sg denotes stop-gradient and $H(\pi_\theta)$ is the policy entropy.

We also retain the demonstration off-policy RL component. The pseudo-demonstration is inserted into the augmented group together with the fresh self-generated rollouts before computing the group-relative reward statistics and advantages. Thus the pseudo-demonstration participates in the same group normalization mechanism that the externally supplied demonstration occupies in original SRFT. The ground-truth correctness reward required by the original method is replaced by the label-free reward in Eq. (6), using the same answer extraction and pseudo-supervision available to DiSCTT.

Finally, we retain SRFT's self-exploration component and its entropy-dependent weighting of positive self-generated trajectories,

$$w_{\text{RL}} = 0.1 \ \text{sg}\Big(e^{H(\pi_\theta)}\Big). \tag{50}$$

Self-generated rollouts are classified as positive or negative using the same label-free reward signal rather than ground-truth correctness.

Accordingly, the LF-SRFT objective preserves the structure of the original SRFT objective,

$$\mathcal{J}_{\text{LF-SRFT}} = w_{\text{SFT}} \, \mathcal{J}_{\text{SFT}}^{\text{pseudo}} + \mathcal{J}_{\text{off}}^{\text{pseudo}} + w_{\text{RL}} \, \mathcal{J}_{\text{pos}} + \mathcal{J}_{\text{neg}}. \tag{51}$$

with the external demonstration and correctness supervision replaced by self-generated counterparts. All four terms update the same LoRA adapter within the same training stage.

LF-SRFT therefore preserves SRFT's distinction between supervised demonstration learning, demonstration-based off-policy RL, and self-generated rollout learning. LF-SRFT is therefore a label-free comparator based on SRFT, adapted to the self-generated supervision available in our test-time setting. This comparator tests whether entropy-aware joint SFT/RL integration can recover the benefit of DiSCTT's consensus-conditioned exclusive objective allocation.

Table 14: LF-SRFT versus DiSCTT under the reported Qwen3-4B training configuration. LF-SRFT is a label-free comparator based on SRFT using self-generated pseudo-supervision.

Benchmark	LF-SRFT	DiSCTT	Gain
AMC	52.5	**57.0**	+4.5
MATH-500	71.6	**75.2**	+3.6
AIME-24	20.0	**23.7**	+3.7
GPQA-Diamond	34.2	**38.4**	+4.2
Hotpot	71.3	**72.9**	+1.6
MMLU	79.8	**81.3**	+1.5
Average	54.9	**58.1**	**+3.2**

LF-SRFT reaches an average accuracy of 54.9%, compared with 58.1% for DiSCTT, a 3.2-point difference under the matched training setting. This comparison examines joint SFT/RL integration and consensus-conditioned exclusive objective allocation under the reported configurations. These controls do not establish global optimality of hard routing; learned soft or three-way routing remains open.

I.7 Consensus versus External-Verifier Routing and Pseudo-Labeling

Consensus provides both the routing signal and the source of SFT pseudo-labels in DiSCTT without requiring a separately trained verifier. To compare this self-generated supervision with an external signal, we replace both consensus-based routing and consensus-based SFT pseudo-label selection with `Qwen2.5-Math-PRM-7B` (Zhang et al., 2025), while holding the Qwen3-1.7B backbone, M rollouts, optimization objectives, and numerical threshold ρ fixed. Each rollout is scored by the verifier, and an input is routed to SFT when the fraction of verifier-passing rollouts reaches ρ. For

inputs routed to SFT, the verifier scores are also used to select the supervised pseudo-target. Thus this comparison evaluates external-verifier-based routing and pseudo-labeling jointly rather than isolating the routing decision alone. Because consensus and verifier scores have different distributions, the comparison is threshold-matched rather than coverage-matched.

Table 15: Consensus-based versus external-verifier-based routing and pseudo-labeling on Qwen3-1.7B. Coverage, SFT pseudo-label precision, conditional error, and JointWrong are measured at the initial routing checkpoint. Conditional error is $1 - \mathrm{Precision}$, and JointWrong is the fraction of all inputs that are both routed to SFT and assigned an incorrect pseudo-label. Final accuracy is measured after adaptation. The comparison uses the same numerical routing threshold ρ but does not imply matched coverage. Extra FLOPs are relative to consensus routing.

Dataset	Router	Coverage	SFT Precision	Cond. Error	Joint Wrong	Final Acc.	Extra FLOPs
AMC	Consensus	37.5	66.7	33.3	12.5	48.8	0%
	Verifier	40.0	87.5	12.5	5.0	50.0	+6.1%
AIME-2024	Consensus	13.3	50.0	50.0	6.7	26.7	0%
	Verifier	16.7	60.0	40.0	6.7	27.3	+4.7%
GPQA-D	Consensus	41.4	68.3	31.7	13.1	29.3	0%
	Verifier	39.9	62.0	38.0	15.2	27.8	+6.4%

The math-specialized verifier yields more reliable SFT selections on the two mathematical benchmarks at the initial checkpoint: SFT pseudo-label precision increases from 66.7% to 87.5% on AMC and from 50.0% to 60.0% on AIME-2024. On GPQA-Diamond, however, consensus achieves higher SFT precision (68.3% versus 62.0%), lower JointWrong (13.1% versus 15.2%), and higher final accuracy (29.3% versus 27.8%). Across the three tasks, final accuracy is similar overall, while the verifier-based variant incurs an additional 4.7–6.4% FLOPs. Because the verifier changes both routing and SFT pseudo-label selection, this experiment should be interpreted as a comparison between two sources of test-time self-supervision rather than as a controlled isolation of routing quality alone. The results show that consensus remains competitive without requiring a separately trained verifier, while verifier effectiveness depends on domain alignment.

J Extended Evaluations

J.1 DiSCTT-OpenEnded: Method Modifications

Table 16: DiSCTT-OE evaluation on open-ended tasks. Values are means and sample standard deviations across 10 independent adaptation/evaluation seeds. The reference-guided best-of-32 comparator is diagnostic. For WMT24++, candidate selection uses sentence-level BLEU and reporting uses corpus BLEU; it is not a corpus-BLEU upper bound or a bound on adapted-policy performance.

Model	Method	CNN/DailyMail		WMT24++ EN→DE
		ROUGE-Lsum	BLEURT	BLEU
Qwen3-1.7B-Base	Base	$0.2335_{\pm 0.0009}$	$0.3881_{\pm 0.0013}$	$16.33_{\pm 0.845}$
	Cluster voting	$0.2469_{\pm 0.0008}$	$0.4047_{\pm 0.0016}$	$20.38_{\pm 0.992}$
	Uniform RL	$0.2632_{\pm 0.0011}$	$0.4098_{\pm 0.0022}$	$23.11_{\pm 0.902}$
	DiSCTT-OE	$0.2997_{\pm 0.0005}$	$0.4283_{\pm 0.0010}$	$25.40_{\pm 0.342}$
	Reference-guided best-of-32	$0.3381_{\pm 0.0014}$	$0.4421_{\pm 0.0031}$	$33.82_{\pm 1.425}$
Llama-3.2-3B-Instruct	Base	$0.2315_{\pm 0.0011}$	$0.2954_{\pm 0.0017}$	$13.09_{\pm 1.005}$
	Cluster voting	$0.2343_{\pm 0.0010}$	$0.3013_{\pm 0.0014}$	$16.15_{\pm 1.124}$
	Uniform RL	$0.2412_{\pm 0.0015}$	$0.3102_{\pm 0.0025}$	$18.69_{\pm 1.117}$
	DiSCTT-OE	$0.2619_{\pm 0.0007}$	$0.3215_{\pm 0.0012}$	$23.91_{\pm 0.542}$
	Reference-guided best-of-32	$0.3054_{\pm 0.0019}$	$0.3310_{\pm 0.0031}$	$31.35_{\pm 1.649}$

Open-ended evaluation protocol. For CNN/DailyMail (Hermann et al., 2015), we use the Hugging Face `abisee/cnn_dailymail` dataset with configuration `3.0.0` and the test split. To obtain a deterministic 1,000-example evaluation subset, we sort the test examples lexicographically by their

id field and select the first 1,000 examples. The id field is the SHA-1 identifier associated with each source article. Source articles are truncated to 2,048 tokens using the tokenizer of the corresponding backbone model.

For WMT24++ (Deutsch et al., 2025), we use the google/wmt24pp dataset and the en-de_DE language-pair subset. We discard examples for which is_bad_source=True and use the post-edited target field as the reference translation. This filtering produces 960 English–German source–reference pairs.

The results in Table 16 are computed across 10 independent adaptation/evaluation seeds. Each adapted checkpoint is evaluated with one fresh generation per input using temperature 0.9, top-$p = 0.9$, and a maximum of 512 generated tokens.

For WMT24++, BLEU is computed with SacreBLEU (Post, 2018) version 2.6.0 using the signature nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.6.0. BLEURT is computed using the BLEURT-20 checkpoint (Sellam et al., 2020). For CNN/DailyMail, we report rougeLsum (Lin, 2004) using rouge-score==0.1.2 with Porter stemming enabled. Following the rougeLsum convention, generated summaries and references are sentence-tokenized with NLTK and sentences are joined by newline characters before scoring.

For the diagnostic best-of-32 comparator, we draw $N_{\text{oracle}} = 32$ independent base-model samples per input. Candidate selection is performed separately for each reported reference-based metric: the highest-ROUGE-Lsum candidate defines the ROUGE-Lsum comparator, the highest-BLEURT candidate defines the BLEURT comparator, and WMT24++ selects candidates using sentence-level BLEU before reporting corpus BLEU. Consequently, different columns need not correspond to the same generated candidate. This procedure is diagnostic, does not participate in adaptation, and does not define a corpus-BLEU upper bound or a bound on adapted-policy performance.

The DiSCTT-OpenEnded (DiSCTT-OE) variant evaluated in Section 4.4 replaces exact-match operations with semantic analogues and uses a larger routing sample count, $M_{\text{OE}} = 32$, to support density-based semantic clustering. The routing threshold ρ, five-round curriculum, SFT/RL alternation, and GRPO group size $N = 32$ are otherwise retained unless explicitly stated below. We describe the semantic-consensus rule, SFT pseudo-target selection, RL semantic-consistency gate, novelty definition, output boundaries, and failure cases explicitly.

Semantic consensus and clustering configuration. The open-ended variant uses a larger routing sample count than the exact-match setting. Specifically, standard DiSCTT uses $M = 8$ routing samples per input, whereas DiSCTT-OE uses $M_{\text{OE}} = 32$ routing samples per input. We use the larger candidate population only for the open-ended extension to support semantic density-based clustering and permit multiple candidate clusters. The cluster-voting baseline is evaluated with the same $M_{\text{OE}} = 32$ candidate-generation budget.

The uniform-RL baseline uses the same $N = 32$ rollout-group size, five-round adaptation schedule, and GRPO optimization settings as the RL branch of DiSCTT-OE, but applies RL to every input and therefore does not perform the additional routing stage. We accordingly treat cluster voting as the sampling-budget-matched inference baseline and uniform RL as the matched optimization baseline rather than claiming identical total generation cost across all methods.

For each input x_j, the M_{OE} generated outputs are embedded and L2-normalized (normalize_embeddings=True). We use HDBSCAN (Campello et al., 2013; Malzer & Baum, 2020) (version 0.8.44) with min_cluster_size=5, min_samples=2, metric="euclidean", cluster_selection_method="eom", and allow_single_cluster=True. Thus, Euclidean distance is computed between L2-normalized embedding vectors. Noise points (label -1) are excluded from cluster membership counts but remain part of the fixed M_{OE} denominator.

Let $\mathcal{C}_j^{\text{dom}}$ denote the largest non-noise cluster. When at least one non-noise cluster exists, semantic consensus is defined as

$$c_j = \frac{|\mathcal{C}_j^{\text{dom}}|}{M_{\text{OE}}}. \tag{52}$$

If HDBSCAN returns no non-noise cluster, we define $\mathcal{C}_j^{\text{dom}} = \varnothing$ and $c_j = 0$; the input is therefore routed to the RL branch and no SFT pseudo-target is constructed.

With $M_{\mathrm{OE}} = 32$ and $\rho = 0.45$, an input requires at least 15 of the 32 routing samples to belong to the dominant semantic cluster to enter the SFT branch. Therefore, `min_cluster_size=5` does not by itself imply high-consensus routing.

SFT pseudo-label selection. On confident inputs ($c_j \geq \rho$), the SFT target is the trajectory in $\mathcal{C}_j^{\mathrm{dom}}$ with the highest average cosine similarity to other cluster members — the most semantically central generation. This replaces uniform selection among modal-answer-consistent completions used for verifiable tasks.

Semantic-consistency gate. For each input routed to RL, we generate a fresh group of $N = 32$ rollouts. These outputs are embedded, L2-normalized, and clustered independently using the same HDBSCAN configuration as the routing stage. Let $\mathcal{C}_{\mathrm{dom}}^{\mathrm{RL}}(x)$ denote the largest non-noise cluster in this fresh rollout group. A rollout y_i passes the semantic-consistency gate if and only if it belongs to $\mathcal{C}_{\mathrm{dom}}^{\mathrm{RL}}(x)$.

Thus, the routing-time cluster is used only to determine the SFT/RL assignment, whereas every fresh RL rollout group constructs its own semantic-consistency target. This mirrors the exact-match formulation, where $a_{\mathrm{route}}(x)$ is used for routing while $a_{\mathrm{RL}}(x)$ is recomputed independently for each fresh GRPO group.

If HDBSCAN returns no non-noise cluster for a fresh RL group, all task-dependent rewards in that group are set to zero. The centered group-relative advantages are consequently all zero, so the policy-gradient component vanishes for that group; the KL regularization term may still contribute.

Semantic novelty. For a rollout y_i that passes the semantic-consistency gate, let z_i denote its L2-normalized generated-output embedding and let $\bar{z}_{\mathrm{dom}}$ denote the centroid of the corresponding dominant semantic cluster. We define semantic novelty as

$$\mathrm{Nov}_{\mathrm{OE}}(y_i) = \frac{\log 2}{2} \left[1 - \cos(z_i, \bar{z}_{\mathrm{dom}})\right]. \tag{53}$$

Because cosine distance lies in $[0, 2]$, this normalization gives $\mathrm{Nov}_{\mathrm{OE}} \in [0, \log 2]$, matching the range of the natural-log Jensen–Shannon novelty term used in the exact-match formulation. We therefore retain $\beta = 1$ without changing the maximum novelty-shaping magnitude.

Output boundaries. For the open-ended experiments, semantic embeddings are computed over the generated task output rather than an explicit chain-of-thought trace. For CNN/DailyMail this corresponds to the generated summary, and for WMT24++ EN→DE it corresponds to the generated German translation. Prompt/source tokens are not included in the candidate-output embedding. Semantic consensus, SFT pseudo-target selection, and semantic novelty therefore operate on the final generated task output. This differs from the exact-match reasoning setting, where the token-distribution novelty term is computed over reasoning-token positions while final-answer tokens are excluded.

For relevance shaping, the generated-output embedding is compared with the corresponding source-input embedding.

Multilingual semantic encoder. All semantic embeddings in the reported DiSCTT-OE experiments are computed using `BAAI/bge-m3`. BGE-M3 is a multilingual sentence encoder, so both German generated translations and English source inputs in WMT24++ are represented within the same multilingual embedding space. The same encoder is used for semantic consensus, cluster-based pseudo-target selection, semantic novelty, and relevance shaping. Embeddings are L2-normalized before the HDBSCAN and cosine similarity computations described above.

J.2 Out-of-Distribution Generalization

A key risk in test-time adaptation is over-specialization. Figure 4 reports held-out transfer after source-task adaptation. Under the evaluated configurations, adaptation frequently improves performance on disjoint reasoning, science, and coding tasks.

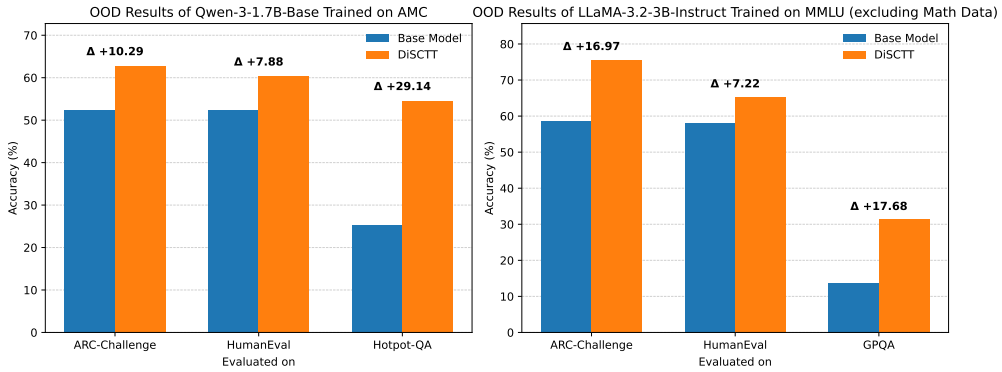


Figure 4: Out-of-distribution transfer performance. Models are adapted on a source task and evaluated on disjoint target tasks. Under the reported configurations, DiSCTT frequently improves performance on held-out reasoning, science, and coding tasks.

For example, adapting Qwen3-1.7B-Base on AMC improves ARC-Challenge, HumanEval, and HotpotQA, while adapting LLaMA-3.2-3B-Instruct on MMLU improves ARC-Challenge and GPQA without regressions.

Interpretation. The observed transfer is consistent with adaptation that is not confined to the source benchmark, although these experiments do not directly identify representation-level changes.

Sequential distribution shift. The transfer results above are relevant to, but not equivalent to, an online easy-to-hard shift. For example, adapting Llama-3.2-3B-Instruct on MMLU while excluding math data improves held-out GPQA, and AMC adaptation transfers to disjoint reasoning and coding tasks. However, we have not run a protocol in which the adaptation stream itself switches from a stabilized easy source (e.g., MATH-500) to a harder target such as MathOlympiad. Such non-stationary adaptation remains a separate evaluation target.

J.3 Training Dynamics Across Difficulty Levels

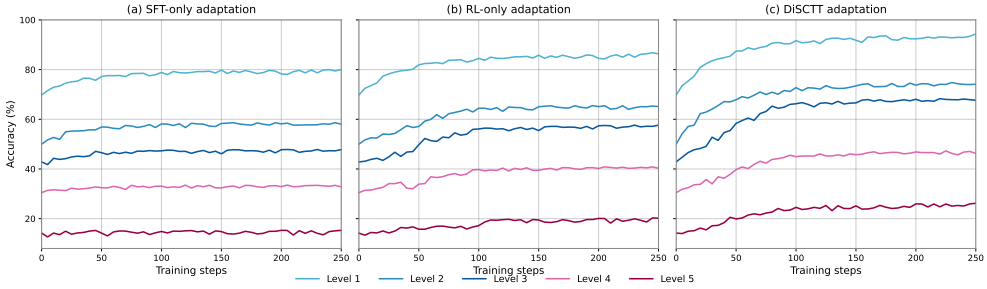


Figure 5: Accuracy across difficulty levels on MATH-500 under different training paradigms. SFT improves easier instances but saturates, RL exhibits slower and more oscillatory training dynamics, and DiSCTT combines rapid early gains with sustained improvements on harder instances.

Figure 5 shows the effect of objective allocation across difficulty levels. DiSCTT achieves faster early improvements while retaining sustained gains on harder instances. The training curves show that supervised adaptation improves rapidly on easier instances but saturates, whereas RL continues to improve harder instances with more oscillatory dynamics. DiSCTT combines these complementary empirical behaviors through consensus-conditioned objective allocation. These curves describe optimization dynamics and should not be interpreted as direct measurements of SFT or RL gradient variance.

Table 17: Ablation of reward components on AMC using Qwen3-4B-Base. Values are mean final-answer accuracy and sample standard deviation across 10 independent adaptation/evaluation runs, with one fresh final-answer generation per question in each run. The Base and complete-reward rows correspond to the canonical Base and DiSCTT results reported in Table 1; intermediate rows compare the incremental reward components.

Variant	Accuracy (%)
Base model	$38.5_{\pm 1.29}$
+ modal-consistency gate	$44.8_{\pm 1.42}$
+ Population-relative novelty	$54.0_{\pm 1.29}$
+ Relevance shaping	$\mathbf{57.0}_{\pm \mathbf{1.05}}$

K Ablations and Sensitivity

K.1 Reward Component Ablation

Table 17 compares the incremental reward components in terms of mean accuracy and run-to-run variability.

Mean accuracy. The modal-consistency gate alone yields a $+6.3$ point improvement under this configuration. Adding population-relative novelty increases accuracy by $+9.2$ points, and including relevance shaping provides a further $+3.0$-point gain.

Run-to-run variability. The reported standard deviations measure variability of final accuracy across independent runs; they are not direct estimates of gradient-estimator variance. The modal-consistency-only variant has the largest observed standard deviation (± 1.42). Adding population-relative novelty reduces it to ± 1.29, and adding the relevance term further reduces it to ± 1.05. This monotonic decrease is consistent with more stable optimization under the shaped reward, but we do not infer quantitative gradient-variance ratios from these accuracy statistics.

K.2 Sensitivity Analyses

A common concern with adaptive frameworks is that strong empirical results may depend on careful hyperparameter tuning. We address this by varying key DiSCTT hyperparameters while holding the remaining training protocol fixed. All experiments in this section use Qwen3-1.7B-Base. The default five-round schedule described in Appendix A is used unless explicitly stated otherwise. We separately study sensitivity to the routing threshold ρ, consensus sample size M, and number of training rounds R.

K.3 Routing Threshold ρ

The threshold ρ determines the consensus level above which an input is routed to supervised consolidation rather than reinforcement learning. Lower ρ shifts more inputs to SFT; higher ρ shifts more to RL.

Table 18: Sensitivity of DiSCTT to the routing threshold ρ. Performance across the tested settings is highest at the default $\rho = 45\%$ and remains relatively stable over the tested interior cutoffs. Because $M = 8$ yields a discrete consensus grid, these results should be interpreted over the corresponding effective routing cutoffs rather than as evidence for a continuous theoretical optimum.

ρ	MATH-500	Avg. across six benchmarks
30%	63.5	48.7
45% (default)	**67.0**	**51.2**
60%	64.8	49.6
75%	61.2	47.3

With $M = 8$, these nominal thresholds correspond to effective grid cutoffs $c \geq 3/8, 4/8, 5/8$, and $6/8$, respectively; values such as 45% and 50% are therefore routing-equivalent. Performance is reasonably stable over the interior cutoffs and degrades at the most RL-heavy setting. This supports robustness of the fixed cutoff on the tested grid, but does not evaluate a time-varying adaptive threshold ρ_t.

K.4 Consensus Sample Size M

Estimating consensus requires sampling M trajectories per input during each recomputation. Smaller M produces noisier consensus estimates at lower cost; larger M tightens the estimate but multiplies sampling overhead linearly.

Table 19: Sensitivity of DiSCTT to the consensus sample size M. Performance is stable across $M \in \{4, 8, 16\}$. We adopt $M = 8$ as the default because it captures essentially all of the available gain over $M = 4$ at half the sampling cost of $M = 16$.

M	MATH-500	AIME-24	AMC
4	65.8	23.3	46.7
8 (default)	67.0	**26.7**	48.8
16	**67.3**	26.7	**49.2**

Increasing M from 4 to 8 improves accuracy on all three benchmarks. Increasing it to 16 produces only marginal further changes while doubling the routing-sample count. We therefore use $M = 8$; these comparisons do not establish statistical significance of the small differences.

K.5 Number of Training Rounds R

The number of training rounds R controls the total duration of adaptation. Each round begins with a fresh routing refresh using $M = 8$ samples per input, followed by 2 SFT epochs on $\mathbb{D}_{\text{conf}}$ and 8 RL epochs on $\mathbb{D}_{\text{unc}}$. The default method uses $R = 5$ rounds. Increasing R therefore provides additional opportunities for both parameter adaptation and dynamic re-routing, at the cost of additional routing, SFT, and RL computation.

Table 20: Sensitivity of DiSCTT to the number of training rounds R. Performance improves substantially from 3 to 5 rounds and remains stable when adaptation is extended to 8 or 10 rounds. We use $R = 5$ as the default to balance adaptation performance and computational cost.

R	MATH-500	AIME-24	AMC
3	61.5	21.3	44.0
5 (default)	67.0	26.7	**48.8**
8	**67.6**	**27.7**	48.5
10	67.3	27.3	48.7

Performance improves from three to five rounds and largely plateaus thereafter. We use five rounds because additional rounds incur further routing, SFT, and RL computation with limited observed benefit.

K.6 HDBSCAN `min_cluster_size` for DiSCTT-OpenEnded

For the open-ended variant (Section 4.4, Appendix J.1), semantic consensus is computed by HDBSCAN clustering of L2-normalized generated-output embeddings. The `min_cluster_size` hyperparameter controls clustering granularity: smaller values permit smaller dominant clusters; larger values require denser agreement.

Across the tested configurations, WMT24++ BLEU ranges from 25.04 to 25.40 and CNN/DailyMail ROUGE-Lsum ranges from 0.2984 to 0.2997. Because these sensitivity runs do not provide paired

Table 21: Sensitivity of DiSCTT-OE to HDBSCAN `min_cluster_size` on WMT24++ EN→DE and CNN/DailyMail. Across the three tested settings, the observed ranges are 0.36 BLEU and 1.3×10^{-3} ROUGE-Lsum. These sensitivity experiments are reported as point estimates and are not used to infer statistical equivalence between clustering configurations.

`min_cluster_size`	WMT24++ BLEU	CNN/DailyMail ROUGE-Lsum
3	25.18	0.2991
5 (default)	**25.40**	**0.2997**
7	25.04	0.2984

repeated-run uncertainty estimates, we report these differences descriptively and do not claim that they fall within stochastic noise. We use `min_cluster_size=5` as the reported configuration.