# THINKBRANCH-OPD: LEARNING BRANCH CHOICES THROUGH COUNTERFACTUAL DISTILLATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A reasoning trajectory often takes shape at a few consequential choices. Once a model selects a substitution or a case split, later steps unfold along that direction. On-policy distillation lets a teacher supervise trajectories actually generated by the student, but feedback is usually delivered token by token along the path already taken. Even when the teacher favors another line of reasoning, that preference is difficult to express as a choice between paths. We introduce ThinkBranch-OPD, which compares the student's realized direction with a teacher-preferred alternative from their shared prefix at points of marked teacher–student disagreement. Supervision thus reaches the choice that determines how the reasoning path continues. Fixed-decision diagnostics show that, after training, the student becomes more inclined toward the teacher-preferred direction at key branches; overall performance on complete solutions also improves. Transferring reasoning ability therefore involves not only learning how to continue a path already begun, but also learning to choose a more promising direction when the path forks.

## 1 INTRODUCTION

Reasoning models such as OpenAI o1, DeepSeek-R1, and Qwen3 generate extended intermediate computations before returning an answer. Transferring this ability to smaller models is important for making deliberate reasoning practical under limited inference budgets. On-policy distillation (OPD) (He et al., 2026; Agarwal et al., 2024) offers a natural solution. The student samples its own reasoning trajectories, and a stronger teacher supervises the states that the student actually visits. This training distribution avoids the mismatch that arises when a student learns only from fixed, teacher-generated demonstrations.

Current OPD methods nevertheless treat a trajectory as a flat sequence. At each position $t$, the teacher supplies a score for the realized token $y_t$, and training aggregates these token-level signals along the sampled path. Reasoning, however, is often governed by a much smaller set of consequential choices. A model may choose a substitution, case split, or proof strategy that determines many later steps. If the student takes direction $A$ when the teacher favors direction $B$, supervision restricted to the continuation of $A$ cannot state the relevant comparison directly. It describes how the teacher would continue from the visited prefix, but not that another direction was preferable at the point of commitment.

This distinction connects several difficulties in reasoning-oriented OPD. Repeated exploration of unproductive directions lengthens trajectories (He et al., 2026), teacher feedback may weaken as an erroneous prefix grows (Liu et al., 2026), and uniform pressure on realized tokens may discourage alternatives needed for self-correction (Kaur et al., 2026). Each effect depends on how supervision represents a choice between paths, a structure obscured by treating positions independently.

Token weighting alone cannot identify the missing alternative, while a full-trajectory outcome merges the effects of many earlier choices. A branch point provides the useful intermediate unit: it retains the common prefix, isolates a local choice, and links that choice to its possible continuations.

Teacher–student disagreement exposes such choices even when both next steps remain locally plausible. We measure it with a truncated Jensen–Shannon divergence and select prominent local maxima as branch points. At each point, the teacher-score difference between its preferred candidate $b$ and the realized token $a$ defines $\hat{A}_k$, ranking two directions from the same prefix.



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

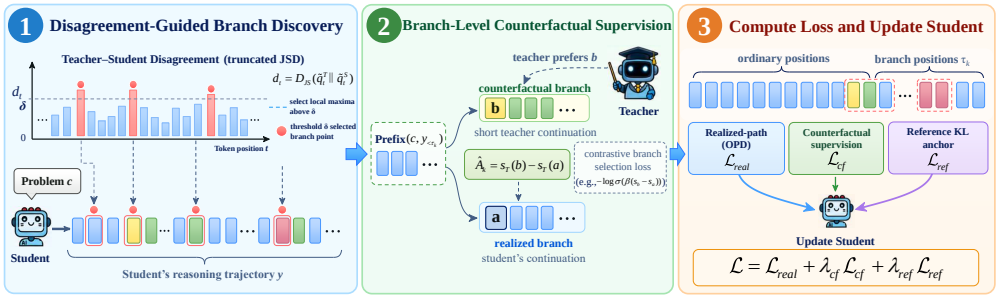



Figure 1: ThinkBranch-OPD detects predictive disagreements, constructs teacher-preferred counterfactual branches, and combines path, branch, and reference supervision.

Based on this observation, we propose ThinkBranch-OPD. It retains token-level OPD away from detected branches, distinguishes clear teacher preferences from ambiguous local alternatives, and applies a preference-weighted contrastive objective at the retained decisions. A short teacher continuation shows how the alternative direction begins to unfold. The resulting objective preserves ordinary on-policy supervision while making influential path choices explicit.

The analysis follows the same progression as the method. It first establishes that branch-free updates recover ordinary on-policy training, then shows how active comparisons move toward the teacher's preference and how avoiding dead ends reduces expected computation. The experiments examine both levels of this argument: fixed-decision diagnostics reveal whether local branch preferences change as intended, and broad end-task evaluation tests how those changes carry into complete solutions. Together, they connect the optimization of a local path choice to the behavior of the resulting reasoning model.

Our contributions are threefold:

1. We formulate a supervision-unit mismatch in reasoning-oriented OPD: training acts on individual tokens, whereas the course of a solution is often determined by choices between branches.
2. We introduce ThinkBranch-OPD, which combines distributional branch-point detection, reliable teacher-derived counterfactual advantages, and contrastive branch supervision within a single on-policy objective.
3. We establish teacher-aligned margin improvement and expected dead-end computation reduction, then connect these properties to local branch behavior and final-answer accuracy through controlled and end-task evaluation.

## 2 Related Work

### 2.1 On-Policy Distillation

Knowledge distillation traditionally transfers a teacher's predictive distribution to a smaller student (Hinton et al., 2015). For language generation, on-policy variants train on samples drawn from the student itself (Agarwal et al., 2024). Generalized Knowledge Distillation (Agarwal et al., 2024) combines teacher-forced and student-generated data with divergence-based objectives. SimpleOPD (He et al., 2026) adapts this approach to reasoning models through text-span alignment across tokenizers and a reference-model penalty that limits drift. ThinkBranch-OPD uses the same broad on-policy setting but represents selected decisions as comparisons between alternative continuations.

A related line of work studies the stability of OPD over long generations. Trust-region behavior blending (Plyusov et al., 2026) and computation-aware pruning (Yang et al., 2026b) address unstable or inefficient reasoning from complementary directions. Supervision fidelity decay (Liu et al., 2026) characterizes how teacher signals change as student-generated prefixes grow. Our approach

uses distributional disagreement at individual prefixes to locate positions where an explicit branch comparison may be useful.

### 2.2 Selective and Structured Reasoning Supervision

Several methods alter token-level supervision without representing explicit alternatives. Ex-OPD (Yang et al., 2026a) extrapolates the teacher–reference log-probability difference to define a stronger target. DEAR (Xiao et al., 2026) identifies decision and evidence tokens through representation-level signals, while other selective distillation methods reweight tokens by estimated importance (Kim & Baek, 2026; Huang et al., 2026). ThinkBranch-OPD instead constructs a pairwise target from two possible next steps at the same prefix. Our experiments additionally include an entropy-gated control that supplements the path objective with forward KL at high-teacher-entropy positions.

Trajectory-level methods provide a second point of contact. Path-Conditioned Self-Distillation (Li et al., 2026) attributes credit around divergence points on a realized path. Project Aletheia (Dixit et al., 2026) learns from verifier-supported backtracking traces, and reinforcement-learning distillation can capture multi-branch structure implicitly (Xu et al., 2026). Analyses of self-distillation further show that privileged supervision may suppress useful forks and self-correction (Kaur et al., 2026). Cognitive markers such as "wait" and "alternatively" have also been used to identify shifts in reasoning (Zhang et al., 2025). Our detector relies on predictive distributions rather than surface markers, and our target compares the realized step with a teacher-preferred alternative.

### 2.3 Preference Learning and Counterfactuals

Direct Preference Optimization (DPO) (Rafailov et al., 2023) expresses preferences through a logistic objective over response likelihood ratios. ThinkBranch-OPD adopts this contrastive form locally, comparing two candidate branches rather than two complete responses. Counterfactual evaluation has a longer history in sequential decision making, including structural causal formulations for alternative trajectories (Oberst & Sontag, 2019). In our setting, the teacher directly supplies a preference at a shared prefix and may generate a short continuation, so no separate causal estimator is required.

## 3 Method

### 3.1 From On-Policy Distillation to Branch Choices

On-policy distillation (OPD) trains on trajectories generated by the student: the teacher evaluates states the student actually reaches, rather than only complete teacher solutions (Agarwal et al., 2024; He et al., 2026). This is useful for reasoning, where an early choice of substitution or case split changes every subsequent prefix. The ordinary update can judge tokens on the chosen route. When another route is preferable, however, those tokens do not directly represent a comparison between the two choices at the point where the routes diverge.

To make the distinction precise, consider the Normal OPD baseline used in this paper. Let $c$ be a problem and $y_{1:n}$ a reasoning trajectory of $n$ tokens, with $t$ indexing a token position. The student $\pi_\theta$ has parameters $\theta$, the teacher $\pi_\phi$ has fixed parameters $\phi$, and $\pi_{\bar{\theta}}$ is a frozen copy of the student used to sample $y_{1:n} \sim \pi_{\bar{\theta}}(\cdot \mid c)$ for the current update. A frozen copy of the initial student, $\pi_{\text{ref}}$, provides the reference policy. At the visited prefix $h_t = (c, y_{<t})$, the teacher scores the realized token $y_t$.

Define $g_t = \log \pi_\phi(y_t \mid h_t) - \log \pi_{\bar{\theta}}(y_t \mid h_t)$, the teacher's relative log-score for that token, and $\rho_t(\theta) = \pi_\theta(y_t \mid h_t)/\pi_{\bar{\theta}}(y_t \mid h_t)$, the current student's probability relative to the rollout policy. A positive $g_t$ means the teacher assigns more probability to the realized token than the rollout policy does. The baseline minimizes

$$\mathcal{L}_0(\theta) = \frac{1}{n}\sum_{t=1}^{n}\left[\ell_t^{\text{path}}(\theta) + \lambda_{\text{ref}}\ell_t^{\text{ref}}(\theta)\right], \tag{1}$$

The two per-token terms are

$$\begin{aligned} \ell_t^{\text{path}}(\theta) &= \max[-g_t\rho_t, -g_t\,\text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)]\,, \\ \ell_t^{\text{ref}}(\theta) &= \exp(r_t - s_t) - (r_t - s_t) - 1. \end{aligned} \tag{2}$$

Here $\epsilon > 0$ is the clipping half-width, $r_t = \log \pi_{\text{ref}}(y_t \mid h_t)$ and $s_t = \log \pi_\theta(y_t \mid h_t)$ are reference and current-student log-probabilities, and $\lambda_{\text{ref}} \geq 0$ weights the reference penalty. The path term follows the teacher's relative score, while the reference term limits drift from the initial student. Every summand in Eq. 1 concerns the realized $y_t$. At a consequential choice, the teacher may prefer a different token $b$ to the sampled token $a$, yet $b$ has no explicit target in this objective. ThinkBranch-OPD retains the baseline update away from detected decisions. At detected decisions, it omits that update and applies a direct branch comparison only when the teacher preference is sufficiently strong.

Algorithm 1 gives the full procedure.

### 3.2 Detecting Decisions for Branch Supervision

The baseline provides a score for the sampled token at every position. To find positions that merit an explicit comparison, each policy retains its $K_{\text{cand}}$ highest-probability next tokens. Let $\widetilde{\pi}_T^t$ and $\widetilde{\pi}_S^t$ be the resulting distributions, renormalized on the union of those candidate sets. We measure their disagreement by the Jensen–Shannon divergence $d_t = D_{\text{JS}}(\widetilde{\pi}_T^t \| \widetilde{\pi}_S^t)$. With $\delta$ denoting the minimum disagreement, a position enters the detected set $\mathcal{D}$ when it is an interior local peak:

$$t \in \mathcal{D} \iff 1 < t < n, \quad d_t \geq \delta, \quad d_t \geq d_{t-1}, \quad d_t \geq d_{t+1}. \tag{3}$$

The distributional criterion can detect a meaningful difference even when the two policies share their most likely token. Peak selection concentrates the extra supervision at pronounced disagreements, leaving ordinary path supervision at other positions.

### 3.3 Teacher-Preferred Alternatives

Write $\tau_k$ for the $k$th detected position. At $\tau_k$, let $a = y_{\tau_k}$ be the realized token and $b = \arg\max_v \pi_\phi(v \mid c, y_{<\tau_k})$ the teacher-preferred token, with $v$ ranging over the vocabulary. Let $\widetilde{s}_T(v)$ equal the teacher log-probability on its retained $K_{\text{cand}}$ candidates and a finite lower score $-M$ outside that set, where $M > 0$ makes $-M$ no greater than any retained score. The teacher's preference strength and the reliable branch set are

$$\hat{A}_k = \widetilde{s}_T(b) - \widetilde{s}_T(a), \qquad \mathcal{D}_\gamma = \{\tau_k \in \mathcal{D} : \hat{A}_k \geq \gamma\}. \tag{4}$$

The advantage is nonnegative because $b$ maximizes the teacher probability. The threshold $\gamma \geq 0$ retains comparisons in which the teacher preference is sufficiently distinct; weaker detected positions receive no branch target.

### 3.4 Branch Supervision and Training Objective

For a retained branch, the student compares the two choices at their shared prefix. With $s_v = \log \pi_\theta(v \mid c, y_{<\tau_k})$, a positive preference scale $\beta$, and the logistic function $\sigma(x) = 1/(1+e^{-x})$, the immediate branch loss is

$$\ell_{\text{br}}(\theta; \tau_k) = -\log \sigma(\beta(s_b - s_a))\,. \tag{5}$$

This local pairwise objective increases the student's relative preference for $b$; it uses neither a complete teacher solution nor a reference-policy ratio inside the comparison (Rafailov et al., 2023).

The teacher may also greedily continue from the common prefix followed by $b$ for $m_k \geq 0$ tokens. This short counterfactual prefix supplies teacher-forced supervision on how the alternative begins to unfold. Denote by $\ell_{\text{cont}}(\theta; \tau_k)$ the sum of teacher-minus-student log-probabilities on those continuation tokens, setting it to zero if no continuation is generated. Its exact token-level form is in Appendix A. With $Z_\gamma = \sum_{\tau_k \in \mathcal{D}_\gamma}(1 + m_k)$ counting retained choice and continuation terms, the branch objective is

$$\mathcal{L}_{\text{cf}}(\theta) = \frac{1}{\max(1, Z_\gamma)} \sum_{\tau_k \in \mathcal{D}_\gamma} \hat{A}_k \big[\ell_{\text{br}}(\theta; \tau_k) + \ell_{\text{cont}}(\theta; \tau_k)\big]. \tag{6}$$

Let $\mathcal{L}_{\text{real}} = n^{-1} \sum_{t \notin \mathcal{D}} \ell_t^{\text{path}}$ and $\mathcal{L}_{\text{ref}} = n^{-1} \sum_{t \notin \mathcal{D}} \ell_t^{\text{ref}}$ be the baseline terms on positions outside the detected set. ThinkBranch-OPD combines them with branch supervision:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{real}}(\theta) + \lambda_{\text{cf}} \mathcal{L}_{\text{cf}}(\theta) + \lambda_{\text{ref}} \mathcal{L}_{\text{ref}}(\theta). \tag{7}$$

Here $\lambda_{\text{cf}} \geq 0$ weights the branch term. Detected positions leave the ordinary path and reference sums; those also meeting the preference threshold receive branch supervision. All other positions keep the Normal OPD update. If the detector selects no branch, Eq. 7 is exactly Eq. 1. In each update, the student samples trajectories, teacher and student predictions determine $\mathcal{D}_\gamma$, and the teacher supplies any short alternative continuations. The student then recomputes probabilities with gradients and optimizes Eq. 7. Appendix B specifies the update and Appendix A gives the detailed reductions.

## 4 Theoretical Analysis

We study the structure introduced by branch supervision. The analysis establishes exact agreement with the training baseline when no branch is selected, a teacher-aligned directional guarantee for every nonzero branch component, and the computation recovered when those local corrections prevent unproductive paths.

### 4.1 Branch-Free Equivalence

We begin with the limiting case in which the detector selects no positions.

**Theorem 1** (Exact Branch-Free Equivalence)**.** *Let $\mathcal{L}_0$ denote the Normal OPD objective in Eq. 1. For every context $c$ and student trajectory $y$, if $\mathcal{D}(\pi_\theta, \pi_\phi; c, y) = \emptyset$, then*

$$\mathcal{L}(\theta; c, y) = \mathcal{L}_0(\theta; c, y). \tag{8}$$

*Consequently, ThinkBranch-OPD and its baseline produce identical per-trajectory gradients whenever the detector returns no branch points.*

The direct substitution proof is given in Appendix A.2.

This result makes branch supervision a conservative extension of the training baseline: it introduces no hidden update on trajectories for which the detector finds no salient decision.

### 4.2 Teacher-Aligned Margin Improvement

At a detected branch, the relevant state variable is the student's relative log-probability

$$M_k(\theta) = s_b - s_a = \log \frac{\pi_\theta(b \mid c, y_{<\tau_k})}{\pi_\theta(a \mid c, y_{<\tau_k})}. \tag{9}$$

Unlike supervision on either token alone, $M_k$ represents the choice between the two directions directly.

**Theorem 2** (Strict Local Margin Improvement)**.** *Let $\hat{A}_k > 0$, $\beta > 0$, and $p_k = \sigma(\beta M_k(\theta))$. Consider a gradient step on the weighted branch term,*

$$\theta^+ = \theta - \alpha \nabla_\theta \Big[-\hat{A}_k \log \sigma(\beta M_k(\theta))\Big]. \tag{10}$$

*Then*

$$M_k(\theta^+) = M_k(\theta) + \alpha \hat{A}_k \beta (1 - p_k) \|\nabla_\theta M_k(\theta)\|^2 + \mathcal{O}(\alpha^2). \tag{11}$$

*For sufficiently small $\alpha$, the teacher-preferred margin increases strictly whenever $\nabla_\theta M_k \neq 0$.*

The differentiation and first-order expansion are given in Appendix A.2.

The teacher-aligned contribution grows with $\hat{A}_k$ and the remaining disagreement $1-p_k$. Section 5.5 evaluates the complete update on fixed decisions; further interpretation is in Appendix A.2.

### 4.3 Expected Reduction in Dead-End Computation

The final result connects a corrected branch probability to trajectory cost. Let $q_k$ be the baseline probability of selecting an unproductive option at branch opportunity $k$, and let $C_k \geq 0$ be its conditional expected token overhead, including exploration and backtracking. Denote the expected useful portion of the trajectory by $U$.

**Theorem 3** (Dead-End Computation Reduction)**.** *Suppose branch supervision changes the dead-end probability at opportunity $k$ from $q_k$ to $q'_k = (1-\eta_k)q_k$, where $\eta_k \in [0,1]$, while $U$ and the conditional costs $C_k$ remain fixed. If the baseline expected length is*

$$L_0 = U + \sum_{k=1}^{K} q_k C_k, \tag{12}$$

*then the expected length after branch correction is exactly*

$$L_{\mathrm{TB}} = L_0 - \sum_{k=1}^{K} \eta_k q_k C_k \leq L_0. \tag{13}$$

*The inequality is strict whenever at least one branch has $\eta_k q_k C_k > 0$.*

The expectation calculation is given in Appendix A.2.

The reduction scales with avoidable dead-end cost. Together, the results describe baseline-equivalent updates without detected branches, teacher-aligned active comparisons, and the computation recovered by successful branch corrections. The common-rate case is in Appendix A.2.

## 5 Experiments

Our experiments test how branch supervision affects end-task accuracy and whether it changes the local comparisons that the objective is designed to optimize.

### 5.1 Setup

**Models.** We evaluate Qwen3-4B with a Qwen3-8B teacher and Qwen3-0.6B with a Qwen3-1.7B teacher. For each pair, the reference policy is a frozen copy of the student before training. Each student and its teacher share a tokenizer, which permits direct token alignment along sampled trajectories.

**Training data.** We train on 4,237 olympiad mathematics problems from NuminaMath-CoT (Li et al., 2024).

**Comparisons.** Base student is the student before training, and the smaller-pair teacher is shown as a reference. Normal OPD uses the on-policy distillation surrogate in Eq. 18 with reference regularization. EOPD adds forward KL at high-entropy positions, and ExOPD constructs a reward-extrapolated target distribution. ThinkBranch-OPD replaces supervision at detected decisions with reliable counterfactual branch comparisons. Within each model pair, trained methods use the same student, teacher, data order, sampling schedule, and optimizer. For every trained method, we fix one checkpoint before evaluation and use it on every benchmark. Objective-specific settings and the preference-reliability schedule are listed in Appendix E.

**Evaluation.** The evaluation covers all 115 available AMC 2023–2025 problems; all 30 problems from each of AIME 2025 and 2026; all 400 problems from IMO-AnswerBench (Luong et al., 2025); the complete 100-problem English Easy and Hard splits of OlymMATH (Sun et al., 2026); all 33

HMMT February 2026 problems; and all 198 GPQA-Diamond questions. AnswerBench is scored with conservative symbolic verification. Every entry uses the full benchmark. Appendix H gives scoring details.

### 5.2 Full-Benchmark Results

Table 1 reports both model pairs on the same eight benchmarks. Overall is the unweighted mean of the eight benchmark accuracies, so each task contributes equally regardless of its number of questions.

Table 1: Eight-benchmark accuracy (%) and its unweighted mean. Bold/underlined values mark the best/second-best student results within each pair. Ans. denotes IMO-AnswerBench; Oly-E/H denote the English OlymMATH splits.

| Method | AMC | AIME25 | AIME26 | Ans. | Oly-E | Oly-H | HMMT | GPQA | Overall |
|---|---|---|---|---|---|---|---|---|---|
| *Qwen3-4B student / Qwen3-8B teacher* | | | | | | | | | |
| Base student | 57.4 | 30.0 | 43.3 | 41.5 | **76.0** | 14.0 | 36.4 | 45.5 | 43.0 |
| Normal OPD | 61.7 | 43.3 | 40.0 | 41.5 | 75.0 | 16.0 | 36.4 | 43.9 | 44.7 |
| EOPD | 59.1 | 40.0 | **50.0** | 39.8 | 74.0 | 16.0 | 36.4 | 42.9 | 44.8 |
| ExOPD | 60.0 | 46.7 | 36.7 | 40.8 | 74.0 | **19.0** | **39.4** | **46.0** | 45.3 |
| ThinkBranch-OPD | **62.6** | **53.3** | 43.3 | **41.8** | **76.0** | **19.0** | 36.4 | 45.5 | **47.2** |
| *Qwen3-0.6B student / Qwen3-1.7B teacher* | | | | | | | | | |
| Base student | 31.3 | 6.7 | 10.0 | 14.3 | 7.0 | **6.0** | 12.1 | 17.7 | 13.1 |
| Teacher | 51.3 | 26.7 | 26.7 | 26.0 | 42.0 | 13.0 | 36.4 | 26.3 | 31.0 |
| Normal OPD | 29.6 | **23.3** | 10.0 | 13.0 | **11.0** | 4.0 | 12.1 | **20.2** | 15.4 |
| EOPD | 31.3 | 13.3 | 6.7 | 12.8 | 10.0 | 4.0 | 18.2 | 19.2 | 14.4 |
| ExOPD | **32.2** | 20.0 | **13.3** | 13.5 | 10.0 | 5.0 | 9.1 | **20.2** | 15.4 |
| ThinkBranch-OPD | **32.2** | 20.0 | 10.0 | **15.3** | 10.0 | 5.0 | **33.3** | 19.2 | **18.1** |

**End-task accuracy.** With the 4B student, ThinkBranch-OPD reaches 47.2% Overall accuracy, 1.9 points above the strongest matched baseline, ExOPD. It leads on AMC, AIME 2025, and AnswerBench, and ties the best OlymMATH Easy and Hard results. With the 0.6B student, it reaches 18.1% Overall, 2.7 points above the strongest student-side baselines. The smaller pair shows especially large differences on HMMT and AnswerBench, while its AIME 2025 and GPQA scores remain close to the stronger baselines. Across both model sizes, explicit branch comparisons improve Overall accuracy.

**How the gains are distributed.** For the 4B student, branch supervision improves seven of the eight benchmark rows relative to Normal OPD and matches the remaining row. The largest difference is on AIME 2025, where accuracy rises from 43.3% to 53.3%. AIME 2026 and OlymMATH Hard gain 3.3 and 3.0 points, respectively, while AMC, AnswerBench, OlymMATH Easy, and GPQA also improve. Thus the macro-average gain does not come from replacing a broad loss of performance elsewhere with a single large win. Relative to ExOPD, the strongest Overall baseline, ThinkBranch-OPD gains 6.6 points on AIME 2025 and 6.6 on AIME 2026. The contest-math pattern is consistent with the method's focus on decisions that set up subsequent steps: a local choice of direction can affect the remainder of an otherwise plausible solution. Appendix D visualizes the benchmark-wise changes.

**Behavior across model sizes.** The smaller student shows a different concentration of gains. On HMMT, ThinkBranch-OPD reaches 33.3%, compared with 18.2% for the strongest other student-side method on that benchmark. It also attains the highest AnswerBench score, 15.3%, in this pair. These results suggest that the useful branch decisions depend on the student's capacity and the problem distribution: AIME 2025 contributes strongly for the 4B student, whereas HMMT is especially responsive for the 0.6B student. The common result is the highest Overall accuracy among trained student methods in both pairs, despite the different sources of improvement.

### 5.3 Ablation Study

Figure 7 compares the final method with variants that change one branch-supervision choice at a time. The full method reaches 47.2% Overall across all eight benchmarks, ahead of every tested variant. Using either the pairwise or continuation target alone lowers Overall accuracy to 44.8% or 45.3%. Shortening the continuation gives the strongest branch-selection variant at 45.1%, indicating that both the local comparison and the subsequent branch trajectory contribute to the aggregate result. Appendix F examines these differences in detail.

### 5.4 Case Study

Figure 2 traces two independently generated solutions to an OlymMATH Easy problem asking for the smallest team whose six-problem contest score is 161. Both reach the same counting state: the score represents 23 correct answers, and six problems contain 15 distinct pairs. Their paths separate when interpreting the condition that no three contestants solved the same pair. Normal OPD allows each pair to occur only once, imposing an unnecessarily tight budget of 15 pair occurrences and eventually answering 11. ThinkBranch-OPD allows two occurrences per pair. With the correct budget of 30, six contestants are impossible, while a valid seven-contestant construction attains all 23 correct answers. It answers 7, the reference solution. The example shows how a local interpretation of a constraint can redirect the subsequent combinatorial search and its final answer.

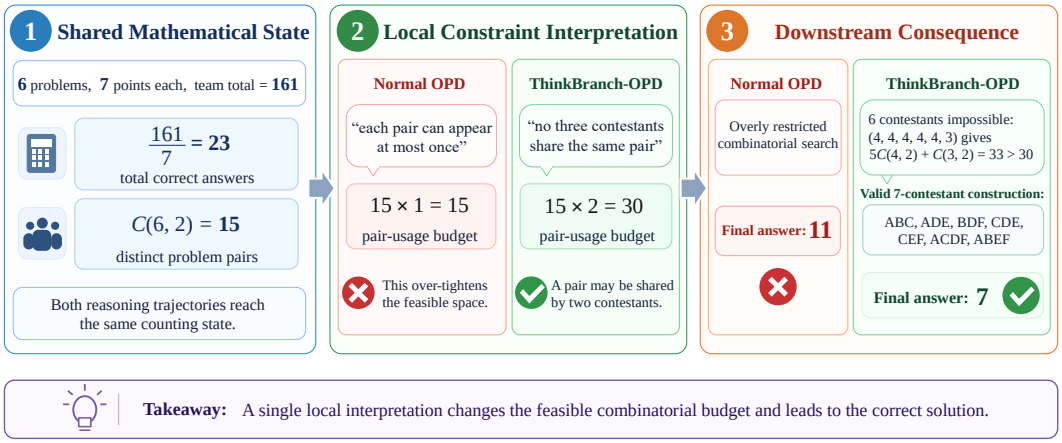



Figure 2: A qualitative OlymMATH Easy case. Two independently generated solutions reach the same mathematical state but follow different interpretations of the pair-sharing constraint. Normal OPD answers 11; ThinkBranch-OPD gives the correct answer 7.

### 5.5 Mechanism Analysis

We score all three models on 2,306 fixed decisions from 256 initial-student rollouts, isolating how branch supervision changes local preferences.

**Where the detector looks.** Figure 3 shows that the selected positions span the trajectory rather than clustering at the final answer; their median relative position is 0.30. The binned median teacher advantage rises with predictive disagreement, with a correlation of 0.50 across the detected branches. The right panel gives a fully evaluated OlymMATH Easy example: for a disc-coloring problem, Normal OPD answers 51, whereas ThinkBranch-OPD counts the non-identity rotational symmetries and obtains the correct answer, 52.

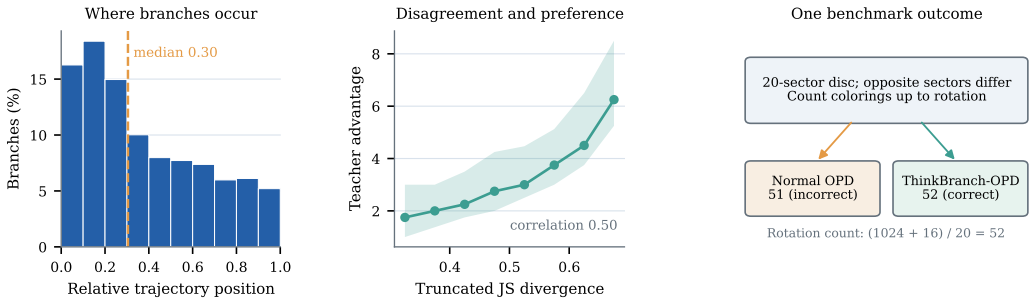



Figure 3: Detected branches and an evaluated example. The first panel shows the positions of 2,306 fixed decisions within their trajectories. The second relates truncated Jensen–Shannon divergence to teacher advantage. The right panel compares two matched methods on an OlymMATH Easy problem with reference answer 52.

**How training changes those decisions.** We next compare teacher-preferred margins and pair losses on the same fixed positions after training.

Figure 4 shows larger teacher-weighted margin and pair-loss gains for ThinkBranch-OPD than Normal OPD (0.114/0.107 versus 0.038/0.035). Its margin rises at 50.0% of fixed decisions versus 36.6% for Normal OPD, with a paired weighted gain of $+0.0759$.

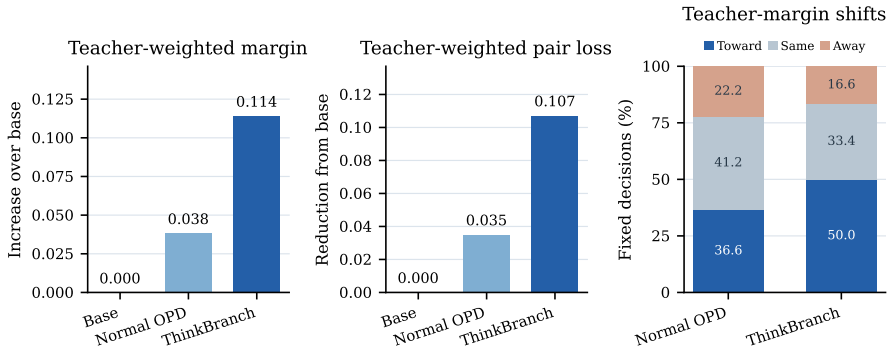



Figure 4: Fixed-decision diagnostics on 2,306 branches. The right panel shows teacher-preferred margin shifts relative to the base student.

The fixed-decision comparison links the end-task results to the quantity optimized by the branch objective. Both students are evaluated on the same original branch contexts, so the change in teacher-preferred margin reflects a change in the trained policy's local choice rather than a change in which branches were sampled. Normal OPD also moves some margins toward the teacher, but explicit branch comparison produces the larger shift. Together with the full-benchmark results, this shows that the proposed supervision affects both the local decision it targets and the accuracy of complete responses.

## 6 Conclusion

ThinkBranch-OPD distills trajectory-defining choices through teacher-preferred counterfactual branches, linking local margins and dead-end computation to stronger aggregate reasoning accuracy.

## AI USE STATEMENT

Generative AI tools assisted with language revision and Chinese–English translation, literature retrieval and discovery, formal-consistency checks, experiment scripting, result analysis, and figure preparation. All implementations, numerical results, citations, and AI-assisted content were independently verified by the authors, who take full responsibility for the paper.

## REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Aradhya Dixit, Tianxi Liang, and Jai Telang. Project aletheia: Verifier-guided distillation of backtracking for small language models. *arXiv preprint arXiv:2601.14290*, 2026.

Haonan He, Haodi Lei, Yun Luo, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Shengji Tang, Zhilin Wang, Runzhe Zhan, Lei Bai, et al. Simpleopd: Simple tokenizer-agnostic on-policy distillation for long-context reasoning. *arXiv preprint arXiv:2608.14277*, 2026.

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

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. Selectkd: Selective token-weighted knowledge distillation for llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19287–19297, 2026.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026.

Minsang Kim and Seung Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. In *International Conference on Learning Representations*, volume 2026, pp. 103174–103196, 2026.

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

Yu Li, Shu Hong, and Tian Lan. Localizing credit at the divergence: Path-conditioned self-distillation for llm reasoning. *arXiv preprint arXiv:2606.15576*, 2026.

Yanjiang Liu, Jie Lou, Xinyan Guan, Yuqiu Ji, Hongyu Lin, Ben He, Xianpei Han, Le Sun, Xing Yu, and Yaojie Lu. Your teacher can't help you here: Combating supervision fidelity decay in on-policy distillation. *arXiv preprint arXiv:2605.30833*, 2026.

Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Clara Huiyi Hu, Henryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu H. Trinh, Quoc V. Le, and Junehyuk Jung. Towards robust mathematical reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35418–35442, 2025.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pp. 4881–4890. PMLR, 2019.

Daniil Plyusov, Alexey Gorbatovski, Alexey Malakhov, Nikita Balagansky, Boris Shaposhnikov, Daria Korotyshova, and Daniil Gavrilov. Trust-region behavior blending for on-policy distillation. *arXiv preprint arXiv:2605.31159*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17438–17457, 2026.

Jinwei Xiao, Zhuowen Han, Yueqing Sun, Zhengxi Lu, Yuxin Liu, Zhiyuan Yao, Wentao Chen, Qi Gu, and Xunliang Cai. Finding the evidence: Discovering decision-supporting tokens for on-policy reasoning distillation. *arXiv preprint arXiv:2606.22830*, 2026.

Shicheng Xu, Liang Pang, Yunchang Zhu, Jia Gu, Zihao Wei, Jingcheng Deng, Feiyang Pan, Huawei Shen, and Xueqi Cheng. Rlkd: Distilling llms' reasoning via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34151–34159, 2026.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026a.

Zhicheng Yang, Zhijiang Guo, Yifan Song, Minrui Xu, Yongxin Wang, Yiwei Wang, Xiaodan Liang, and Jing Tang. Prune-opd: Efficient and reliable on-policy distillation for long-horizon reasoning. *arXiv preprint arXiv:2605.07804*, 2026b.

Zhenyu Zhang, Xiaoxia Wu, Zhongzhu Zhou, Qingyang Wu, Yineng Zhang, Pragaash Ponnusamy, Harikaran Subbaraj, Jue Wang, Shuaiwen Leon Song, and Ben Athiwaratkun. Understanding and steering the cognitive behaviors of reasoning models at test-time. *arXiv preprint arXiv:2512.24574*, 2025.

# A Method Details and Proofs

## A.1 Branch Objectives and Reductions

The branch objective uses the same student-generated prefixes as ordinary OPD. Its additional information is a comparison between two next-step choices at one of those prefixes. The following definitions specify how the comparison is constructed and how its loss is combined with supervision on the realized trajectory.

At position $t$, let $T_{T,t}^{(K_{\text{cand}})}$ and $T_{S,t}^{(K_{\text{cand}})}$ be the teacher and student sets of their $K_{\text{cand}}$ highest-probability tokens, and let $V_t^{(K_{\text{cand}})}$ be their union. Each truncated distribution retains its original probabilities on its own candidate set, assigns zero to the remaining entries of $V_t^{(K_{\text{cand}})}$, and renormalizes. The resulting disagreement score is

$$d_t = D_{\text{JS}}(\widetilde{\pi}_T^t \| \widetilde{\pi}_S^t) = \tfrac{1}{2}\text{KL}(\widetilde{\pi}_T^t \| m_t) + \tfrac{1}{2}\text{KL}(\widetilde{\pi}_S^t \| m_t), \quad m_t = \tfrac{1}{2}(\widetilde{\pi}_T^t + \widetilde{\pi}_S^t). \tag{14}$$

The union puts both distributions on the same finite support, including candidates that appear in only one policy's top set. Thus $d_t$ reflects disagreement over alternatives as well as disagreement over the most probable token. Comparing $d_t$ with its neighboring positions, as in Eq. 3, identifies a local concentration of disagreement rather than treating a run of adjacent high scores as separate decisions. The shared prefix at a selected position makes the ensuing teacher preference a direct comparison of two available choices.

For the branch advantage, $\widetilde{s}_T(v) = \log \pi_\phi(v \mid c, y_{<\tau_k})$ on the retained teacher candidates and $-M$ otherwise, with $-M$ no greater than their retained log-scores. This keeps $\hat{A}_k$ finite when the realized token falls outside that set. The finite score also preserves the ordering between the teacher's preferred candidate and an omitted realized token. Selection by $\mathcal{D}_\gamma$ then separates detection of a disagreement from the decision to use that disagreement as a preference target. A detected position with an insufficient teacher advantage contributes neither a branch comparison nor a continuation, while the path and reference terms continue to supervise positions outside $\mathcal{D}$.

If continuation is enabled, the teacher greedily generates $z_{1:m_k}^{(k)}$ after the alternative token $b$:

$$z_j^{(k)} = \arg\max_v \pi_\phi(v \mid c, y_{<\tau_k}, b, z_{<j}^{(k)}), \qquad j = 1, \ldots, m_k. \tag{15}$$

With $h_j^{(k)} = (c, y_{<\tau_k}, b, z_{<j}^{(k)})$, the continuation score used in Eq. 6 is

$$\ell_{\text{cont}}(\theta; \tau_k) = \sum_{j=1}^{m_k} \left[\log \pi_\phi(z_j^{(k)} \mid h_j^{(k)}) - \log \pi_\theta(z_j^{(k)} \mid h_j^{(k)})\right]. \tag{16}$$

Because the sequence is greedily decoded, this is a fixed-sequence log-ratio score, not a Monte Carlo estimate of a distributional KL. The teacher probabilities in Eq. 16 are fixed during a student update. Consequently, the student-dependent part rewards probability on the teacher's short continuation after $b$. The pairwise term acts at the choice itself, whereas the continuation term describes how that choice begins to develop. Both are weighted by the same teacher advantage in Eq. 6; the normalization by $Z_\gamma$ accounts for the number of retained choice and continuation terms.

For the realized trajectory, write $h_t = (c, y_{<t})$. The baseline teacher advantage and student-to-rollout ratio are

$$g_t = \log \pi_\phi(y_t \mid h_t) - \log \pi_{\bar{\theta}}(y_t \mid h_t), \qquad \rho_t(\theta) = \frac{\pi_\theta(y_t \mid h_t)}{\pi_{\bar{\theta}}(y_t \mid h_t)}. \tag{17}$$

The path term uses the Normal OPD surrogate at positions outside the detected set $\mathcal{D}$:

$$\mathcal{L}_{\text{real}}(\theta) = \frac{1}{n} \sum_{t \notin \mathcal{D}} \max[-g_t \rho_t(\theta), -g_t \,\text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)]\,. \tag{18}$$

The path term retains the full-trajectory denominator $n$ while excluding positions in $\mathcal{D}$; only positions in $\mathcal{D}_\gamma$ receive branch supervision. The reference term likewise acts outside $\mathcal{D}$:

$$\mathcal{L}_{\text{ref}}(\theta) = \frac{1}{n} \sum_{t \notin \mathcal{D}} \left[\exp(r_t - s_t) - (r_t - s_t) - 1\right], \tag{19}$$

where $r_t = \log \pi_{\text{ref}}(y_t \mid h_t)$ and $s_t = \log \pi_\theta(y_t \mid h_t)$. Each summand is nonnegative and vanishes when the log-probabilities agree. Under sampling from $\pi_\theta$ over the complete support, its expectation equals $\text{KL}(\pi_\theta \| \pi_{\text{ref}})$. Together, Eqs. 18 and 19 preserve the baseline's treatment of the realized path wherever no branch is detected. The counterfactual term changes the supervision at selected decisions, not the source of the sampled trajectory. This division is also why the objective recovers ordinary OPD exactly when $\mathcal{D}$ is empty.

### A.2 Proofs of Theoretical Results

The three results in Section 4 describe different parts of the same objective. The first concerns the complete loss on a trajectory without selected decisions. The second isolates the gradient of one active branch comparison. The third expresses the expected computation saved when a corrected choice avoids a dead end.

*Proof of Theorem 1.* Suppose $\mathcal{D} = \emptyset$. Since $\mathcal{D}_\gamma \subseteq \mathcal{D}$, the retained branch set is empty as well, $Z_\gamma = 0$, and the sum defining $\mathcal{L}_{\text{cf}}$ is zero. The index set $\{t : t \notin \mathcal{D}\}$ is now the full set $\{1, \ldots, n\}$. Equations 18 and 19 therefore reduce to the path and reference sums in Eq. 1, with the same denominator $n$ and the same coefficients. Substitution in Eq. 7 gives $\mathcal{L}(\theta; c, y) = \mathcal{L}_0(\theta; c, y)$ for every $\theta$. Because the two functions coincide, their gradients with respect to the student parameters also coincide for this trajectory. □

The equality is at the level of the objective and its update, not merely the value of a statistic measured after training. In particular, the branch mechanism adds no term when its detector does not select a position. For trajectories with detected positions, the path and reference sums continue to use the same per-token definitions on the remaining positions.

*Proof of Theorem 2.* At the shared prefix, let $M_k = s_b - s_a$ and $p_k = \sigma(\beta M_k)$. For the weighted pairwise term $\ell_k = -\hat{A}_k \log \sigma(\beta M_k)$, the derivative of $-\log \sigma(x)$ is $-(1 - \sigma(x))$. The chain rule yields

$$\nabla_\theta \ell_k = -\hat{A}_k \beta (1 - p_k) \nabla_\theta M_k, \qquad \nabla_\theta M_k = \nabla_\theta s_b - \nabla_\theta s_a. \tag{20}$$

Thus the gradient step changes the parameters by $\theta^+ - \theta = \alpha \hat{A}_k \beta (1 - p_k) \nabla_\theta M_k$. Expanding the margin around $\theta$ gives

$$\begin{aligned} M_k(\theta^+) - M_k(\theta) &= (\nabla_\theta M_k)^\top (\theta^+ - \theta) + \mathcal{O}(\alpha^2) \\ &= \alpha \hat{A}_k \beta (1 - p_k) \|\nabla_\theta M_k\|^2 + \mathcal{O}(\alpha^2), \end{aligned} \tag{21}$$

which is Eq. 11. The leading term is positive when $\hat{A}_k > 0$, $\beta > 0$, $p_k < 1$, and $\nabla_\theta M_k \neq 0$. It therefore dominates the second-order remainder for sufficiently small positive $\alpha$. □

The margin is a log-probability difference between two tokens under the same student distribution. Writing their pre-softmax logits as $u_b$ and $u_a$ gives $M_k = u_b - u_a$, because the shared log-normalizer cancels. The pairwise loss consequently acts on the relative choice rather than on an arbitrary shift of all logits. The squared-gradient term in the theorem is nonnegative regardless of the angle between $\nabla_\theta s_b$ and $\nabla_\theta s_a$. Its scale grows with the teacher advantage and the remaining disagreement $1 - p_k$, and decreases as the preferred choice becomes more likely. The complete update additionally includes the path, continuation, and reference terms; Section 5.5 measures their combined effect on fixed decisions.

*Proof of Theorem 3.* Let the expected useful computation be $U$. At opportunity $k$, an unproductive choice occurs with probability $q_k$ and adds conditional expected cost $C_k$. Its expected contribution is therefore $q_k C_k$, and linearity of expectation gives $L_0 = U + \sum_k q_k C_k$. After correction, that contribution becomes $q'_k C_k = (1 - \eta_k) q_k C_k$. Summing over opportunities yields

$$L_{\mathrm{TB}} = U + \sum_{k=1}^{K} (1 - \eta_k) q_k C_k = L_0 - \sum_{k=1}^{K} \eta_k q_k C_k. \tag{22}$$

Each subtracted term is nonnegative, so $L_{\mathrm{TB}} \leq L_0$. The inequality is strict if at least one term is positive. This use of linearity does not require the branch events to be independent. □

When all dead-end probabilities fall by the same fraction $\eta$, the expression simplifies to $L_{\mathrm{TB}} = L_0 - \eta W_0$, where $W_0 = \sum_k q_k C_k$ is the original dead-end overhead. The formula identifies which choices have the largest potential effect on trajectory cost: a correction matters most when the wrong direction is both likely and expensive to pursue. Together with the margin result, it explains how a local teacher-preferred choice can alter the amount of subsequent unproductive computation.

## B Training Algorithm

Algorithm 1 separates target construction from the differentiable student update. Freezing the rollout policy makes the sampled tokens and their old-policy probabilities well defined for the path term. Teacher scores, disagreement peaks, retained branch pairs, and optional short continuations are then computed for those sampled prefixes before optimizing the student. This ordering keeps each counterfactual comparison tied to a state the student actually visited.

An update begins with problems $c$ drawn for the current batch. The frozen rollout policy $\pi_{\bar{\theta}}$ generates a trajectory $y_{1:n}$ for each sampled response. Its token log-probabilities are stored alongside the trajectory because the path ratio in Eq. 17 compares the trainable student against precisely this rollout policy. The teacher and student are then evaluated on the visited prefixes $h_t = (c, y_{<t})$. Their candidate next-token distributions supply $d_t$ at every interior position, while the teacher also scores the realized $y_t$ for the ordinary path term.

For each local peak satisfying Eq. 3, the realized token $a$ and the teacher's preferred token $b$ share the same prefix. Their teacher-score difference gives $\hat{A}_k$. The preference threshold separates clear comparisons from weaker disagreements; only positions in $\mathcal{D}_\gamma$ enter the counterfactual loss. At a retained position, the teacher may continue after $b$ to provide the short sequence used by Eq. 16. The reference policy scores the realized path independently. All these quantities are targets for the current update, so their construction does not require gradients through the teacher, reference policy, or rollout generation.

The second phase replays the stored prefixes through the trainable student. It evaluates the probability ratio for each realized token, the relative log-probability $s_b - s_a$ for each retained pair, and the probabilities of any teacher continuation tokens. These quantities form $\mathcal{L}_{\text{real}}$, $\mathcal{L}_{\text{cf}}$, and $\mathcal{L}_{\text{ref}}$ with the reductions in Appendix A. Detected positions are removed from the ordinary path and reference sums; only the reliable subset receives explicit branch supervision. This distinction matters in implementation because $\mathcal{D}$ and $\mathcal{D}_\gamma$ need not be equal.

**Algorithm 1** One training update of ThinkBranch-OPD.

**Require:** Student $\pi_\theta$, teacher $\pi_\phi$, ref $\pi_{\text{ref}}$, problems $\mathcal{P}$, hyperparams $\lambda_{\text{cf}}, \lambda_{\text{ref}}, \beta, \delta, \gamma, K_{\text{cand}}$

Construct supervision targets without gradients:
Freeze the rollout policy $\pi_{\bar{\theta}} \leftarrow \pi_\theta$ and sample $\{(c, y_{1:n})\}$ from it
**for** each rollout $(c, y_{1:n})$ **do**
  Evaluate teacher log-probabilities and top-$K_{\text{cand}}$ candidates at every position
  Evaluate student top-$K_{\text{cand}}$ candidates at every position
  Compute truncated $d_t = D_{\text{JS}}(\widetilde{\pi}_T^t \| \widetilde{\pi}_S^t)$; detect $\mathcal{D}$ via Eq. 3
  Compute $\hat{A}_k$ for each $\tau_k \in \mathcal{D}$ via Eq. 4
  Retain $\mathcal{D}_\gamma = \{\tau_k \in \mathcal{D} : \hat{A}_k \geq \gamma\}$
  If continuation is enabled: teacher generates $z_{1:m_k}^{(k)}$ from each retained $\tau_k$
  Evaluate reference log-probabilities $\log \pi_{\text{ref}}(y_t \mid \cdot)$
**end for**
Optimize the student:
Set $\pi_\theta$ to training mode and clear gradients
**for** each record **do**
  Recompute student log-probabilities $s_t = \log \pi_\theta(y_t \mid \cdot)$ with gradients and ratios to $\pi_{\bar{\theta}}$
  Compute $\mathcal{L}_{\text{real}}$ (Eq. 18), $\mathcal{L}_{\text{cf}}$ (Eq. 6), $\mathcal{L}_{\text{ref}}$ (Eq. 19)
  $\mathcal{L} = \mathcal{L}_{\text{real}} + \lambda_{\text{cf}}\mathcal{L}_{\text{cf}} + \lambda_{\text{ref}}\mathcal{L}_{\text{ref}}$
  Backpropagate $\mathcal{L}/N$ and release the computation graph
**end for**
Clip gradients and update student parameters

During optimization, the student recomputes log-probabilities for both the realized trajectory and the teacher-preferred branch. The path objective covers ordinary positions; the branch objective assigns relative probability to the two choices and, when present, to the preferred continuation. The reference term is evaluated on the realized trajectory outside detected positions. After the gradient update, the next rollout uses the updated student, so later branch comparisons follow the student's changing distribution while each individual update remains anchored to its frozen rollout.

The two normalizers play different roles. The path and reference losses divide by the full realized length $n$, preserving the baseline reduction even when several positions receive branch supervision. The counterfactual loss divides by $\max(1, Z_\gamma)$, where $Z_\gamma$ counts one choice term and $m_k$ continuation tokens for each retained branch. If no branch is detected, the counterfactual sum vanishes and the update is exactly the ordinary OPD update, as proved in Appendix A. When branches are present, the loss acts both on what the student actually generated and on teacher-preferred alternatives from those same prefixes.

## C Matched Diagnostic Protocol

The mechanism diagnostic compares Normal OPD with a matched ThinkBranch-OPD checkpoint. We select branches once from initial-student rollouts over 256 problems and reuse exactly those decisions for every model; 254 problems contain at least one selected branch. Table 2 reports absolute scores on all 2,306 fixed decisions. This design connects the training objective to a measurable change in local choice: each policy faces the same prefix and the same realized and teacher-preferred alternatives, so differences in the diagnostic reflect how the policy ranks those alternatives.

Relative to Normal OPD, ThinkBranch-OPD changes mean margin, teacher-weighted margin, and pair loss by $+0.0670$, $+0.0759$, and $-0.0721$, respectively.

Table 2: Absolute fixed-decision diagnostic scores. A larger teacher-weighted margin and a smaller teacher-weighted pair loss are better.

| Model | Teacher-weighted margin | Teacher-weighted pair loss |
|---|---|---|
| Base Qwen3-4B | $-3.648$ | 3.780 |
| Normal OPD | $-3.610$ | 3.745 |
| ThinkBranch-OPD | $-3.534$ | 3.673 |

Holding the decision contexts fixed matters because the detector depends on the student and teacher distributions. If each trained model selected its own branches, an average-margin change could also reflect a different set of positions. Here every model scores the same prefix, realized token, and teacher-preferred alternative. The direction counts complement the mean: ThinkBranch-OPD moves 50.0% of fixed decisions toward the teacher preference, compared with 36.6% for Normal OPD, while the fraction moving away falls from 22.2% to 16.6%. The change therefore extends across more decisions, in addition to the larger teacher-weighted mean shift. The pair-loss reduction also agrees with the relative-probability target in Eq. 5.

The two scores in Table 2 describe complementary aspects of this change. The margin tracks the student's relative log-probability of the teacher-preferred next token against its original choice. The pair loss evaluates the same comparison through the objective's logistic form. Their joint movement indicates that the update raises the teacher-preferred alternative relative to the realized choice at the shared prefix. That distinction is central to branch supervision: the useful signal is which direction to take before a continuation commits to it.

The direction counts show that this effect is distributed over the fixed decision set. More positions move toward the teacher preference, and fewer move away, so the mean gain is accompanied by a broader change in the student's local choices. This provides a behavioral counterpart to the full-benchmark improvement: the end-task gains occur alongside the particular shift in branch preference that the method is designed to produce. The diagnostic does not depend on selecting a different set of positions after training; it measures what the trained student does when revisiting the same decisions.

## D Complete-Benchmark Visualization

The two visualizations present complementary views of the complete-benchmark results. Figure 5 shows where the 4B student's gains over Normal OPD arise and compares the Overall results for both student sizes with their strongest matched baselines. The unchanged HMMT result remains visible in Table 1. Figure 6 then places every 4B student-side method on the same eight benchmark axes, making the aggregate comparison readable alongside the individual task outcomes.

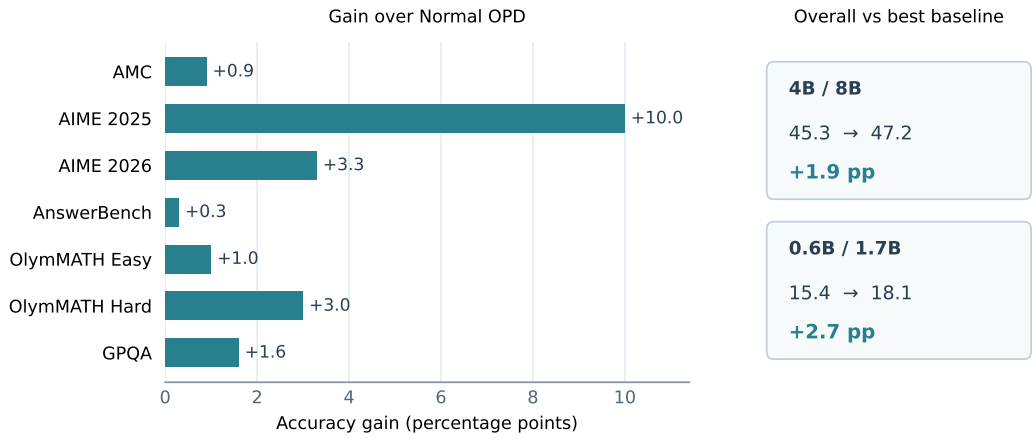



Figure 5: Nonzero accuracy changes from Normal OPD to ThinkBranch-OPD with the Qwen3-4B student (left), computed from the main-table results; the unchanged HMMT result remains in Table 1. The right panel reports Overall accuracy, the eight-benchmark unweighted mean, against the strongest matched student-side baseline for each model pair.

The matrix uses within-column color normalization only to make methods comparable on benchmarks with different accuracy ranges; its annotations retain the original percentages. Read together with the main table, the two figures distinguish a broad shift across tasks from a gain concentrated in a single benchmark.

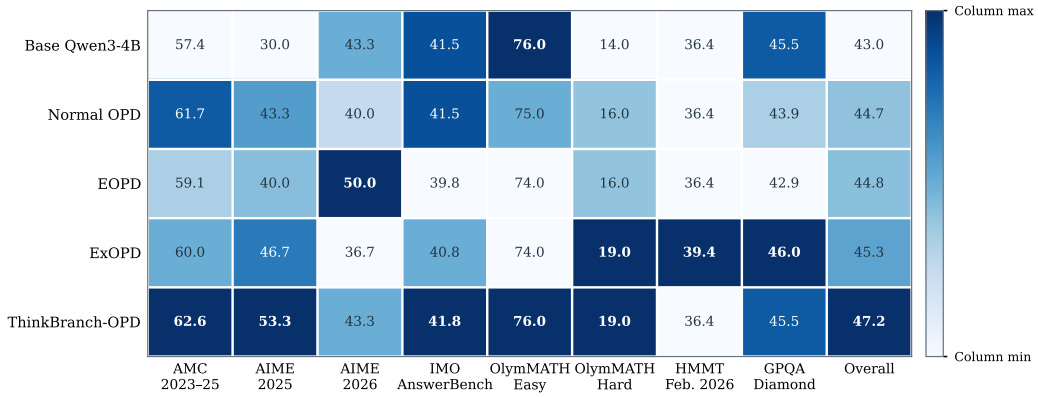



Figure 6: Accuracy of five matched methods on eight complete benchmarks and their unweighted macro average. Darker cells indicate higher accuracy relative to other methods in the same column; cell labels give absolute percentages.

**Retention relative to the initial student.** ThinkBranch-OPD improves the 4B student's AMC, AIME 2025, AnswerBench, and OlymMATH Hard results while matching its initial accuracy on AIME 2026, OlymMATH Easy, HMMT, and GPQA. The 47.2% Overall score therefore combines gains with retention of the base model's stronger rows. The smaller pair improves Overall from 13.1% before training to 18.1%, with especially clear progress on AIME 2025 and HMMT. These different task profiles are consistent with a detector that responds to the current model predictions rather than a benchmark-specific intervention location.

The 4B gain is especially informative because the student already starts with substantial accuracy on several benchmarks. The method raises performance where additional branch guidance is most

visible while preserving the initial result on the remaining rows. Thus the higher Overall score reflects stronger performance from an already capable student, with improvements added to retained capabilities. For the smaller student, the larger relative movement from its initial Overall score shows that the same supervision can reshape choices even when the starting policy is much weaker. In both cases, the intervention is attached to disagreements encountered during training, whereas the reported gains are measured on complete answers across different test collections.

**Comparison with other distillation objectives.** EOPD adds forward-KL targets at selected high-entropy positions of the sampled path, while ExOPD extrapolates a reward-weighted target distribution. ThinkBranch-OPD instead compares two directions from the same prefix. Its 47.2% Overall accuracy with the 4B student exceeds ExOPD's 45.3%; with the 0.6B student, it reaches 18.1% versus 15.4% for the strongest alternatives. The benchmarks contributing most to these gains differ across scales, yet the aggregate result leads the matched student-side methods in both pairs.

These comparisons highlight the distinction between representing a choice and adding another token-level target to the sampled path. Relative to Normal OPD, the 4B student gains on AIME 2025 and several other benchmarks; the 0.6B result draws a particularly large contribution from HMMT. The different profiles are compatible with the method's model-dependent branch detection: what matters is a teacher–student disagreement at a consequential prefix, not a fixed list of task-specific positions. The common outcome is that explicitly supervising the alternative direction improves the eight-benchmark balance at both capacities. Together with the fixed-decision results in Appendix C, the benchmark view shows both the intended local preference shift and its association with better complete solutions.

## E Matched Distillation Baselines

The matched comparison holds the student, teacher, data, optimizer, and sampling schedule fixed. Training draws from all 4,237 problems and uses 16 problems per update, two student rollouts per problem, a 768-token rollout budget, and AdamW with a learning rate of $5 \times 10^{-7}$. Evaluation uses one checkpoint fixed before evaluation. The full result matrix appears in Table 1. Each method therefore receives teacher information at states reached by the same kind of student rollout; the main distinction is how that information becomes an update target. Normal OPD scores the realized path, EOPD adds distributional supervision at high-entropy positions, and ExOPD changes the target distribution through reward extrapolation. ThinkBranch-OPD instead identifies a disagreement at a shared prefix and directly contrasts the realized next token with the teacher-preferred alternative. Table 3 records the objective-specific choices used to implement those distinctions.

Table 3: Objective-specific settings in the matched comparison. Settings not listed in the table are shared across methods.

| Method | Distillation objective | Objective-specific setting |
|---|---|---|
| Normal OPD | Clipped on-policy path surrogate | Detector disabled; reference weight 0.5 |
| EOPD | Clipped path surrogate with entropy-selected forward KL | Entropy threshold 0.8; forward-KL top-16 |
| ExOPD | Reward-extrapolated target distribution | Extrapolation scale 1.25 |
| ThinkBranch-OPD | Path loss with counterfactual branch comparisons | Branch weight 0.4; advantage floor 2.5 after warm start; reference weight 0.5 |

The reliable-preference configuration begins with the ordinary branch objective and activates the minimum advantage of 2.5 after a warm start. It preserves the shared data order while introducing preference filtering later in training. The same checkpoint is used across all benchmarks in the main table. The full-benchmark comparison shows that this change in supervision is useful at both student sizes. The 4B student's 47.2% Overall exceeds the strongest alternative student-side result of 45.3%; the 0.6B student reaches 18.1% versus 15.4% for the strongest alternatives. The largest contributing benchmarks are not identical across the two pairs: the 4B result has a marked AIME 2025 gain, while the smaller student's HMMT gain is especially pronounced. This pattern connects the shared branch-choice objective to improvements across distinct model capacities without making the intervention specific to one benchmark.

## F Ablation Analysis

The ablations in Figure 7 separate decisions about what a branch should learn from decisions about where branch supervision should be applied. All scores use the same eight-benchmark Overall definition as the main table. This makes the effects of the branch targets readable alongside their contribution to complete answers, rather than through a metric defined only on detected positions.

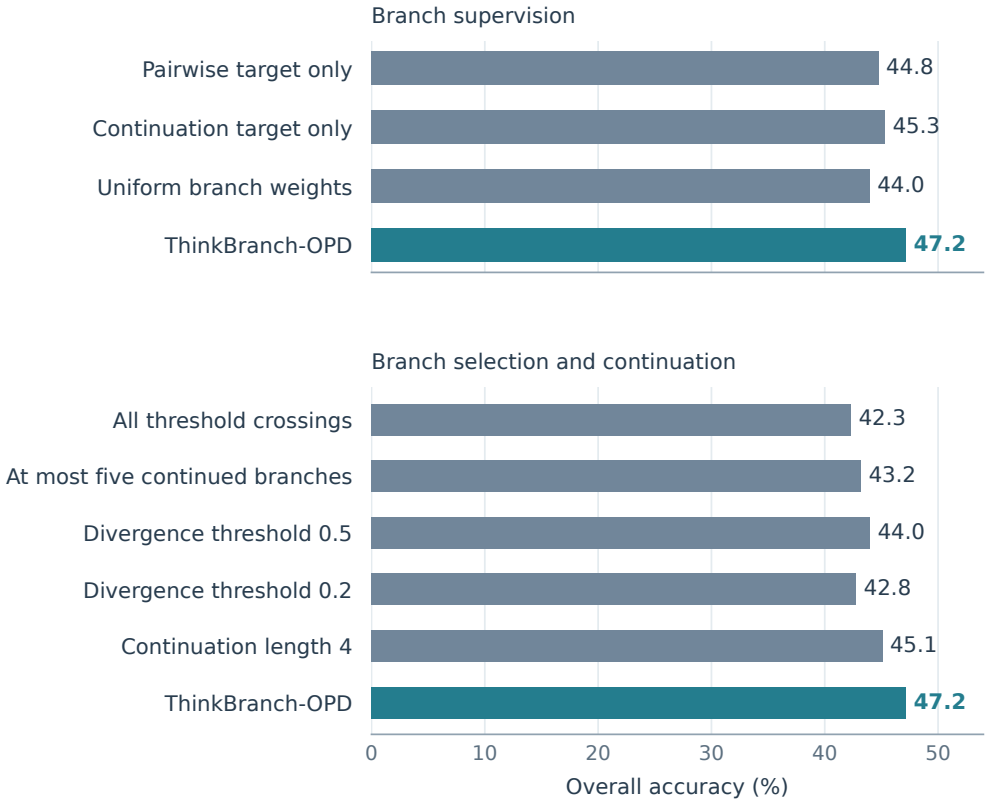



Figure 7: Eight-benchmark Overall accuracy of branch-supervision variants. The upper panel compares the branch targets and weighting; the lower panel compares branch selection and continuation. Both include AnswerBench and use the same unweighted mean as Table 1.

The upper panel first separates the immediate branch comparison from supervision of what follows the alternative choice. With only the pairwise target, the student receives a direct preference between the realized next token and the teacher-preferred one at their shared prefix. With only the continuation target, it instead learns from how the candidate directions unfold after that choice. These variants reach 44.8% and 45.3% Overall, respectively, while combining the targets reaches 47.2%. The result is consistent with the two signals addressing different parts of the same decision: the pairwise term identifies the preferred direction at the fork, and continuation feedback connects that direction to the reasoning it enables.

The weighting ablation probes which detected comparisons should carry more influence. Replacing reliability-sensitive weights with uniform branch weights yields 44.0% Overall. A detected disagreement does not by itself imply that its alternative is equally informative as every other one. Weighting the comparisons lets clearer teacher preferences contribute more strongly while retaining supervision across selected branches. The aggregate difference supports treating branch reliability as part of the learning signal, rather than merely as a condition for finding candidate positions.

The lower panel turns to branch selection. Retaining every position that crosses the disagreement threshold yields 42.3% Overall, compared with 47.2% for the full method. Neighboring positions on a trajectory may reflect the same developing choice, so selecting local peaks concentrates compar-

isons at distinct decision points. Changing the divergence threshold also changes which disagreements enter training: the two alternatives shown in the figure reach 44.0% and 42.8%. Together, these results connect the benefit of branch supervision to the positions it selects, not just to the presence of an additional loss term.

Continuation controls provide a complementary view. Limiting the number of continued branches gives 43.2% Overall, and shortening each continuation gives 45.1%. Both variants retain branch comparison, yet the full configuration gives the strongest aggregate result. The continuation supplies evidence about the consequences of a local choice; a preference at one token becomes more useful when it is informed by the reasoning paths that follow it. Across the component and selection ablations, the same design emerges: identify consequential forks, compare the available directions there, and use their subsequent trajectories to shape the student's preference.

## G Implementation Details

Table 4: Training configuration.

| Parameter | Value |
|---|---|
| Student model | Qwen3-4B |
| Teacher model | Qwen3-8B |
| Training data | NuminaMath-CoT olympiad subset |
| Problems per batch | 16 |
| Rollouts per problem | 2 |
| Learning rate | $5 \times 10^{-7}$ |
| Optimizer | AdamW |
| Gradient clipping | 2.0 |
| $\lambda_{\mathrm{cf}}$ (ThinkBranch) | 0.4 |
| $\lambda_{\mathrm{ref}}$ | 0.5 |
| $\beta$ (pairwise temperature) | 1.0 |
| $\epsilon$ (path-ratio clipping) | 0.2 |
| $\delta$ ($D_{\mathrm{JS}}$ threshold) | 0.3 |

Table 4 summarizes the Qwen3-4B/Qwen3-8B training configuration. Two student trajectories are sampled for each problem in a batch, yielding 32 realized trajectories per update. The student and teacher use the same tokenizer, allowing teacher scores to be evaluated at the student's sampled token positions. The reference policy is a frozen copy of the student before training; it supplies the reference term while the current student changes across updates.

The listed coefficients distinguish the parts of the objective. The reference weight regulates movement from the initial student, and the counterfactual weight scales supervision at selected branches. The disagreement threshold governs which positions are considered for a branch comparison. Table 3 separately records how the matched baseline objectives differ; Table 4 describes the proposed method's training configuration.

## H Evaluation Benchmark Details

The main evaluation covers complete problem sets rather than sampled subsets. The benchmarks differ in answer format as well as subject matter. Competition-mathematics collections require a short answer extracted from a longer generated solution; depending on the reference format, the extracted value is checked by exact matching, normalization, or symbolic equivalence. GPQA-Diamond instead requires a choice among supplied options. Keeping these scoring rules tied to each benchmark lets the comparison measure completed solutions while respecting the answer form each task expects. The constituent collections and their scoring rules are described below.

**AIME 2025.** This row contains all 30 problems from the American Invitational Mathematics Examination 2025. The required answers are integers from 0 to 999. We extract the final answer from the `\boxed{}` expression and compare the resulting integer exactly with the reference. The score

therefore evaluates whether the completed reasoning reaches the required answer, irrespective of the wording of intermediate steps.

**AIME 2026.** This row contains all 30 problems from AIME 2026. It uses the same integer answer format and final-answer verification as AIME 2025. The two years remain separate rows in the main table, so their accuracies contribute independently to Overall.

**AMC 2023–2025.** The AMC row combines all 115 available problems in the evaluation files: 40 from 2023, 45 from 2024, and 30 from 2025. Their references may be integers or fractions. We extract the final boxed expression and normalize its numerical form before comparison, allowing equivalent representations of the same value to receive the same decision. The main table reports accuracy over the full combined AMC collection as one benchmark row.

**IMO-AnswerBench.** We use all 400 short-answer olympiad problems in AnswerBench-v2 (Luong et al., 2025). The benchmark metadata divides them equally among algebra, combinatorics, geometry, and number theory, with 100 problems per category. The category balance makes this a broad short-answer collection while its 400 questions remain a single row in the Overall average.

**OlymMATH.** We evaluate all 100 problems in each of the English Easy and English Hard splits (Sun et al., 2026). They appear as separate rows in Table 1, preserving the distinction between the two provided difficulty splits in the aggregate. For both, the final answer is extracted from the completed response and checked through normalization or symbolic equivalence. The same evaluation form thus applies to both splits while each retains its own accuracy.

**HMMT February 2026.** We use all 33 short-answer problems in the released MathArena collection. The final answer is extracted and checked by symbolic equivalence in the same manner as the other competition-mathematics sets. HMMT has fewer questions than the other mathematics collections, but its reported accuracy remains a separate benchmark row and contributes equally to Overall.

**GPQA-Diamond.** We evaluate all 198 questions in the Diamond split. Each problem supplies answer options in the order stored in the evaluation file. The model's final selected option is compared exactly with the gold option, rather than being passed through the mathematical-expression verifier. This provides a distinct answer format within the same eight-benchmark summary.

Within each benchmark, every method is evaluated on the complete question set under its benchmark-specific scoring rule. The resulting accuracies measure final answers from full generated responses. The fixed-prefix branch diagnostics in Section 5.5 complement this view: they measure how model preferences move at selected decisions, while the benchmark scores show how the trained models perform after those decisions unfold into complete solutions.