OPD-COOKBOOK: PRACTICAL RECIPES FOR STABLE AND EFFECTIVE ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-Policy Distillation (OPD) combines on-policy training with dense token-level supervision from a stronger teacher, making it an increasingly important component of modern model post-training. Despite its simple formulation, making OPD effective in practice is not always straightforward. To address this challenge, we propose an OPD-Cookbook consisting of three complementary algorithmic recipes for reliable and effective OPD. The Cookbook starts from the mechanism of OPD and organizes its analysis around two fundamental aspects of the distillation process: the supervision signal provided by the teacher and the objective used to align the teacher and student. From the perspective of teacher supervision, we first introduce Teacher Adaptation on OPD Data. When the teacher retains improvement headroom on the data used for OPD, further adapting it with reinforcement learning can strengthen the supervision subsequently provided to the student. We further introduce Teacher Recovery Training on Student States, motivated by the observation that a teacher capable of solving a problem on its own trajectory may still provide unreliable guidance when conditioned on imperfect student prefixes. We therefore explicitly train the teacher to recover from states visited by the student. From the perspective of the alignment objective, we propose a Decoding-aware KL Objective that aligns the token distributions actually used during decoding rather than only the raw model distributions, better matching the training objective with generation behavior. Across our experiments, these three recipes provide complementary improvements over standard OPD and together offer a practical set of principles for building more effective OPD pipelines.

1 INTRODUCTION

On-Policy Distillation (OPD) has emerged as a critical paradigm in the post-training pipeline of modern large foundation models. Mechanistically, this approach allows the student model to generate its own trajectories and queries a stronger teacher to evaluate the target next-token distribution at each token position. Consequently, OPD functions as a highly effective bridge between supervised fine-tuning and reinforcement learning, empowering foundation models to steadily improve their complex generative capabilities via both on-policy learning and dense token-level supervision.

Despite its theoretical elegance and growing popularity, making OPD consistently effective in practice is non-trivial. Recent empirical studies have established that standard implementations often suffer from fragility and instability, occasionally even leading to policy degradation (Li et al., 2026; Fu et al., 2026; Luo et al., 2026; Zhu et al., 2026). These findings underscore a fundamental realization: the success of OPD demands far more than passively applying the standard objective to an off-the-shelf teacher and student.

To understand what governs OPD, we revisit the underlying mechanism of OPD, from which its practical effectiveness is inherently governed by two fundamental dimensions: (1) the reliability of teacher supervision, and (2) the formulation of the alignment objective. The former dictates the quality of guidance the student receives along its generated trajectories, while the latter determines how effectively this guidance transfers into the student policy. Guided by this principled decomposition, we develop an **OPD Cookbook**, systematically deriving three complementary recipes that address the hidden bottlenecks within each dimension.

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.

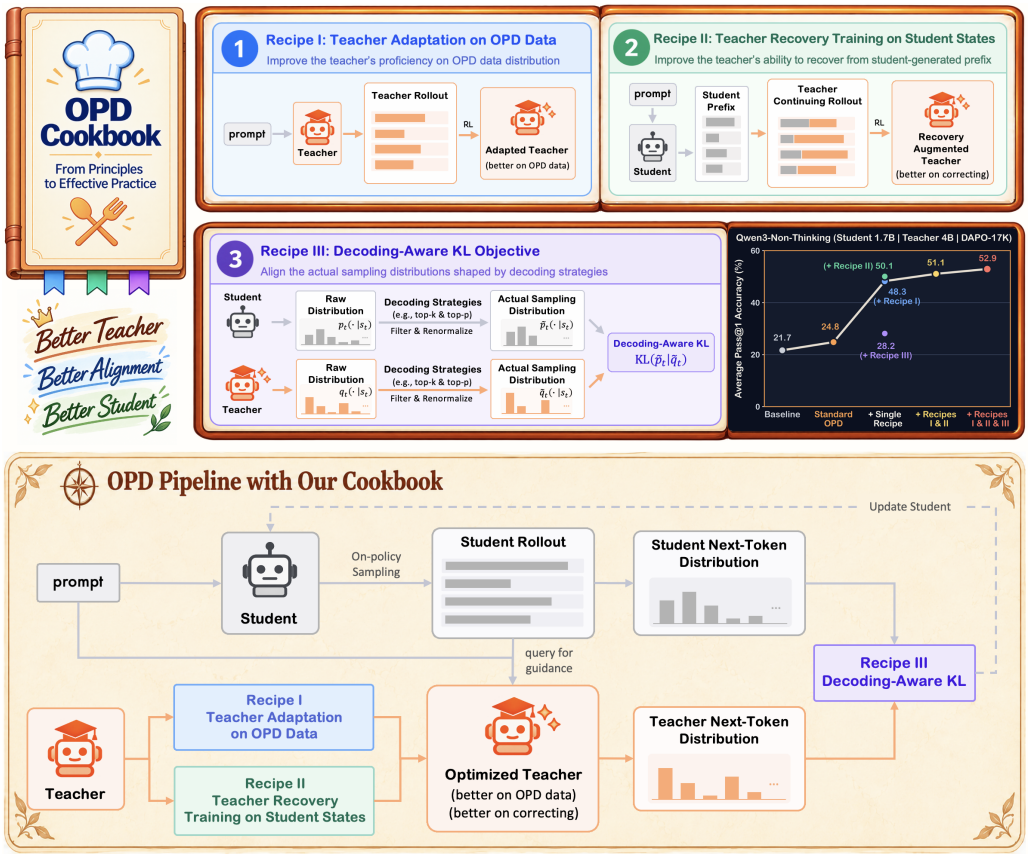

Figure 1: Overview of the OPD Cookbook. **(Top)** Illustrations of three proposed recipes: Recipes I and II enhance the teacher's intrinsic proficiency and error-recovery capabilities. Recipe III aligns the actual sampling distributions shaped by decoding strategies to resolve train-inference mismatches. The inset curve demonstrates the cumulative performance gains when stacking these recipes. The performance is averaged on four challenging benchmarks (AIME24, AIME25, AIME26, AMC23). **(Bottom)** The end-to-end workflow illustrates how the proposed recipes are seamlessly incorporated into the standard OPD training pipeline.

Teacher Supervision Reliability. The first dimension concerns whether the teacher can provide reliable supervision for the student. We propose two complementary recipes for improving this supervision by treating the teacher as an actively optimizable component rather than a static oracle.

- First, even a capable teacher can retain measurable adaptation headroom on the specific OPD training data. We find that exploiting this headroom can further improve teacher's intrinsic proficiency on the specific tasks it teaches, which subsequently transfers into stronger downstream student performance. This motivates our ***Recipe I: Teacher Adaptation on OPD Data***, which directly adapts the teacher on the OPD data distribution via reinforcement learning to extract its untapped headroom and sharpen its in-distribution proficiency before the actual distillation begins. This principle is analogous to an educator refining their expertise for a specific curriculum to ensure effective instruction, despite already having strong foundational knowledge.
- Second, a reliable teacher must not only excel as an independent problem-solver but also as an astute critic capable of identifying and correcting errors in student trajectories. A counterintuitive finding from our recoverability tests reveals that a teacher capable of solving a problem on its own can still struggle to recover when forced to continue from student prefix. Notably, this recoverability deteriorates as the prefix lengthens, causing the teacher to become increasingly derailed by the student's mistakes and thereby triggering cascading error propagation. This motivates our ***Recipe II: Teacher Recovery Training on Student States***, which explicitly trains the teacher to recover from student-generated prefixes, equipping the teacher to provide constructive corrections when encountering suboptimal student states.

Teacher-Student Alignment Objective. The second dimension concerns whether the alignment objective faithfully reflects the distributions that govern the student's actual generation behavior. Standard OPD typically aligns the raw output distributions of the teacher and the student. However, during actual generation, models rarely sample from these raw distributions; instead, they rely on the sampling distributions shaped by decoding strategies (e.g., top-k or top-p filtering). Consequently, this creates a fundamental mismatch between the distribution being optimized during training and the distribution governing actual generation. This motivates our ***Recipe III: Decoding-Aware KL Objective***, which aligns the actual sampling distributions shaped by decoding strategies, rather than the nominal raw distributions the model outputs. Since the decoding strategies typically restricts probability mass to specific candidate subsets, the teacher and student naturally yield different support sets. Directly aligning them can produce infinite KL penalties if the student samples a token outside the teacher's support set, triggering a zero probability assignment. To resolve this numerical instability, we introduce a dynamic expansion mechanism that augments the teacher's support set with the student's support set, followed by re-normalization over this unified set. This yields a finite and well-defined divergence while strictly preserving the decoding-aware nature of the objective.

OPD Cookbook at a Glance. The resulting OPD Cookbook is summarized below. Together, these three recipes form a systematic framework that addresses the core bottlenecks of OPD from both the supervision source and the alignment objective. Crucially, these recipes are mechanistically orthogonal. They can be applied independently to mitigate specific training instabilities or combined synergistically to unlock the full potential of the distillation process.

OPD Cookbook

- **Recipe I: Teacher Adaptation on OPD Data (Sec. 3).** When the teacher retains adaptation headroom on the OPD training data, further adapting it with reinforcement learning can strengthen its supervision and improve downstream student performance.
- **Recipe II: Teacher Recovery Training on Student States (Sec. 4).** A capable teacher may still be misled by imperfect student prefixes. Training the teacher to recover from such states improves its ability to provide reliable corrective guidance during OPD.
- **Recipe III: Decoding-Aware KL Objective (Sec. 5).** Align the actual sampling distributions shaped by decoding strategies (such as top-k & top-p filtering), rather than the raw distributions directly output by the models.

Empirically, the proposed recipes yield substantial and complementary improvements over standard OPD. As shown in Figure 1, under a Qwen3-Non-Thinking setup (1.7B student, 4B teacher; DAPO-17K dataset), standard OPD yields only marginal gains, raising the average accuracy from 21.7% to 24.8%. In contrast, individually applying Recipe I, Recipe II, and Recipe III provides significant boosts, delivering additional gains of +23.5, +25.3, and +3.4 points over standard OPD, and achieving Pass@1 accuracies of 48.3%, 50.1%, and 28.2%, respectively. Combining the two teacher-side recipes (Recipe I & II) further reaches 51.1%, and incorporating all three recipes achieves 52.9%, representing a **+28.1** point improvement over standard OPD. These results highlight that optimizing teacher supervision and refining the alignment objective are not merely complementary, but each plays a critical role in overcoming the inherent bottlenecks of standard OPD. Jointly addressing both dimensions is therefore essential to unlocking the full potential of the distillation process.

2 PRELIMINARIES

On-Policy Distillation (OPD) trains a student model using token-level supervision from a teacher model on trajectories generated by the student's current policy. Benefiting from both on-policy training and dense token-level supervision, OPD has emerged as an increasingly important component of modern model post-training.

Problem Formulation. Let $\mathcal{D}$ denote a training dataset of prompts. Given a prompt $x \sim \mathcal{D}$, the student model π_θ generates a response sequence $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_L)$, where $L = |\hat{y}|$ denotes the response length. At decoding step t, conditioned on the prompt x and the student-generated prefix

$\hat{y}_{<t} = (\hat{y}_1, \ldots, \hat{y}_{t-1})$, we denote the raw next-token distributions of the student and teacher as

$$p_t(\cdot) = \pi_\theta(\cdot \mid x, \hat{y}_{<t}), \qquad q_t(\cdot) = \pi_T(\cdot \mid x, \hat{y}_{<t}), \tag{1}$$

respectively. Importantly, both models are evaluated on the same prefix generated by the student. OPD trains the student by minimizing the discrepancy between the student and teacher distributions along student-generated trajectories. A common formulation minimizes the expected cumulative token-level KL divergence:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \hat{y}\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{L} \mathrm{KL}\big(p_t \,\|\, q_t\big) \right], \tag{2}$$

where $\mathrm{KL}(p_t \| q_t)$ denotes the reverse KL following the convention.

Practical Forms of the KL Objective. The KL term in Eq. (2) can be instantiated at different levels of vocabulary granularity. These choices determine how much information from the teacher distribution is retained at each token position, as well as the associated computational and memory costs. We consider three representative formulations that will be referenced throughout the paper.

(1) Full Vocabulary KL. The most direct formulation computes the reverse KL over the entire vocabulary $\mathcal{V}$:

$$\mathrm{KL}^{\text{OPD}}_{\text{Full}}\big(p_t \,\|\, q_t\big) = \sum_{v\in\mathcal{V}} p_t(v) \log \frac{p_t(v)}{q_t(v)}. \tag{3}$$

This formulation aligns the complete next-token distributions of the student and teacher and therefore provides dense supervision over all vocabulary entries. However, a straightforward implementation that materializes the full distributions at every token position requires $\mathcal{O}(L|\mathcal{V}|)$ storage for a response of length L, which introduces substantial memory and communication overhead when the vocabulary is large.

(2) Top-K Vocabulary KL. A more compact formulation restricts the KL computation to a high-probability subset of the vocabulary. Let

$$\mathcal{P}_t^K = \mathrm{TopK}\big(p_t(\cdot), K\big) \subseteq \mathcal{V} \tag{4}$$

denote the set of the K tokens assigned the highest probability by the student at step t. The student and teacher distributions are then re-normalized over this shared support:

$$\widetilde{p}_t^{(\mathcal{P}_t^K)}(v) = \frac{p_t(v)\mathbb{I}[v \in \mathcal{P}_t^K]}{\sum_{u\in\mathcal{P}_t^K} p_t(u)}, \qquad \widetilde{q}_t^{(\mathcal{P}_t^K)}(v) = \frac{q_t(v)\mathbb{I}[v \in \mathcal{P}_t^K]}{\sum_{u\in\mathcal{P}_t^K} q_t(u)}. \tag{5}$$

The resulting KL objective is

$$\mathrm{KL}^{\text{OPD}}_{\text{TopK}}\big(p_t \,\|\, q_t\big) = \sum_{v\in\mathcal{P}_t^K} \widetilde{p}_t^{(\mathcal{P}_t^K)}(v) \log \frac{\widetilde{p}_t^{(\mathcal{P}_t^K)}(v)}{\widetilde{q}_t^{(\mathcal{P}_t^K)}(v)}. \tag{6}$$

Compared with full vocabulary KL, this formulation retains supervision over multiple candidate tokens while avoiding the need to preserve the complete teacher distribution at each position, providing a practical balance between supervision richness and computational viability.

(3) Sampled-Token KL. A more lightweight yet effective formulation is Sampled-Token KL, which considers only the token $\hat{y}_t$ sampled along the student trajectory. This is the formulation adopted in the OPD implementation presented by Thinking Machines Lab (Thinking Machines Lab & Lu, 2025) and has since been widely used in practice. Specifically,

$$d_t = \log p_t(\hat{y}_t) - \log q_t(\hat{y}_t), \qquad \mathrm{KL}^{\text{OPD}}_{\text{Sampled}}\big(p_t \,\|\, q_t\big) = \mathbb{E}_{\hat{y}_t\sim p_t}\left[d_t\right]. \tag{7}$$

Strictly speaking, d_t itself is a pointwise log-probability ratio rather than a KL divergence, while its expectation under the student distribution recovers the reverse KL. Following the official implementation in (Thinking Machines Lab & Lu, 2025), this formulation can be readily incorporated into standard policy optimization frameworks such as PPO-like importance-sampling loss (Schulman et al., 2017; Shao et al., 2024), where the negative sampled log-ratio directly serves as the token-level advantage in the policy loss, *i.e.*, $\hat{A}_t = -d_t$. We provide the corresponding PPO formulation in Appendix D.

Overall, these formulations span different levels of supervision granularity, ranging from alignment over the full vocabulary distribution to supervision on only the sampled token. In practice, Top-K Vocabulary KL and Sampled-Token KL are more commonly adopted in scalable OPD settings.

3 RECIPE I: TEACHER ADAPTATION ON OPD DATA

Although a teacher selected for OPD is typically stronger than the student, it may still retain room to improve on the data used for OPD. We find that when such adaptation headroom exists, further optimizing the teacher on the OPD training data can strengthen the supervision subsequently provided to the student. Accordingly, our first recipe is to treat the teacher itself as an optimizable component of the OPD pipeline rather than a fixed input to it. Intuitively, this mirrors how a human teacher prepares for a lesson: even with sufficient knowledge of the subject, reviewing and refining their understanding of the material beforehand can lead to more effective teaching.

Motivation. Our motivation stems from an empirical observation: *a stronger teacher can still retain measurable adaptation headroom on the OPD data.* As shown in Figure 2(a), further RL training of Qwen3-4B model on the widely-used DAPO-17K dataset improves its accuracy from 50.5% to 76.0%. This observation motivates a natural question: *can exploiting this adaptation headroom translate into stronger downstream OPD performance?* In other words, can a teacher further adapted on the OPD data produce a better student after distillation? The rationale is straightforward, since the teacher is queried to provide dense token-level supervision on student generated trajectories, any improvement in the teacher may therefore translate into more informative guidance for the student.

Recipe. Our recipe introduces ***teacher adaptation training***, which optimizes the teacher on the OPD training data before using it to supervise the student. Specifically, for each prompt $x \sim \mathcal{D}$ in the OPD training data, the teacher π_T samples a group of G_{I} candidate trajectories, denoted as $\mathcal{Y}_T(x) = \{y^{(i)}\}_{i=1}^{G_{\mathrm{I}}}$, and is optimized with reinforcement learning (RL). This adaptation training can be implemented with any standard policy optimization method, and our experiments instantiate it with GRPO and outcome rewards for simplicity. Concretely, each sampled trajectory $y^{(i)}$ receives a verifiable outcome reward $R(x, y^{(i)}) \in [0, 1]$, and its credit is assigned through the group-normalized advantage

$$\hat{A}_{\mathrm{ada}}\big(x, y^{(i)}\big) = \frac{R\big(x, y^{(i)}\big) - \mathrm{mean}\big(\{R\big(x, y^{(j)}\big)\}_{j=1}^{G_{\mathrm{I}}}\big)}{\mathrm{std}\big(\{R\big(x, y^{(j)}\big)\}_{j=1}^{G_{\mathrm{I}}}\big)}. \tag{8}$$

The teacher is then updated by maximizing the clipped surrogate objective $\mathcal{J}$ in Eq. (15) (Appendix D), instantiated with the advantage given by Eq. (8).

A crucial practical question is how to determine when this adaptation training should stop. We find that excessive adaptation can lead to overfitting, which subsequently reduces the effectiveness of downstream OPD. To prevent this, we maintain a held-out validation set to monitor teacher performance throughout adaptation and indicate the stopping point. In our experiments, the validation set is constructed as a compact coreset selected from the OPD data and excluded from teacher adaptation training. We periodically evaluate the teacher on this set and select the checkpoint with the strongest validation performance. Other appropriate validation sets can also be used in practice. The selected checkpoint then serves as the teacher for OPD, with the student training procedure otherwise unchanged.

Setup. We pair a Qwen3-4B teacher with a Qwen3-1.7B student, both in Non-Thinking mode, and adopt the widely used DAPO-17K as the OPD training data. Performance is assessed across seven popular complex reasoning benchmarks. To isolate the effect of teacher quality, we strictly fix the student initialization, training data, and optimization hyperparameters across all runs, varying only the teacher checkpoint used to provide supervision during OPD.

Results. Our experiments indicate that exploiting teacher adaptation headroom before distillation can substantially strengthen the supervision available to OPD, while validation-guided checkpoint selection is essential for fully realizing this benefit. ***First,*** the teacher's in-distribution proficiency directly governs the student's downstream generalization. As shown in Figure 2(a), as the teacher adapts, its validation accuracy and the student's averaged test accuracy rise in tandem, both peaking around step 140 before declining due to teacher overfitting. This empirical alignment confirms that monitoring the teacher's held-out performance effectively prevents the student from inheriting overfitting artifacts, making validation-based early stopping a necessary safeguard. ***Second,*** leveraging the adapted teacher yields consistent and substantial performance gains across diverse complex reasoning tasks. As illustrated in Figure 2(b), replacing the off-the-shelf teacher with the adapted checkpoint elevates the student's average Pass@1 accuracy from 27.5% to 45.4% (+17.9%).

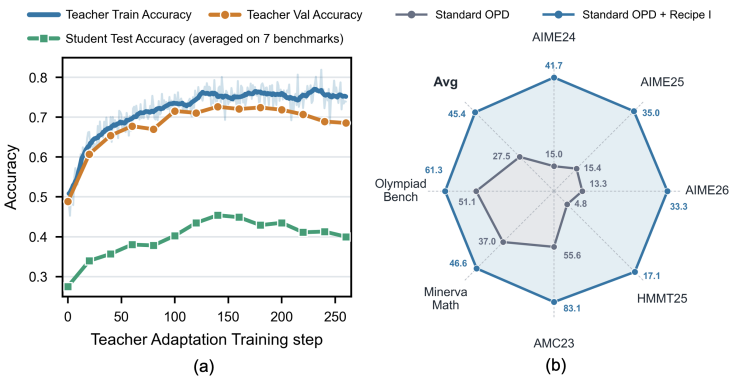

Figure 2: **Teacher adaptation dynamics and downstream effect of Recipe I. (a)** Teacher's train and validation accuracy together with downstream student accuracy. Validation accuracy closely tracks downstream student performance, while excessive adaptation degrades both validation and student accuracy. **(b)** Downstream impact of Recipe I on OPD, evaluated by per-benchmark student Pass@1.

Notably, these improvements are universal, with the adapted teacher outperforming the standard baseline on all seven benchmarks, demonstrating that exploiting the teacher's adaptation headroom directly translates into a significantly stronger student policy.

4 Recipe II: Teacher Recovery Training on Student States

We find that a teacher that excels at independent problem-solving may still struggle to steer the remaining trajectory back to a correct solution when forced to continue from an imperfect student-generated prefix, rendering its token-level supervision unreliable. To address this vulnerability, our second recipe explicitly trains the teacher to recover from student-generated states. Intuitively, this mirrors the role of a human educator, who must not only possess deep subject knowledge but also the ability to identify and correct a student's misconceptions to guide them back to the right path.

Motivation. Our motivation stems from a stark empirical observation: *a teacher's reliability degrades sharply when forced to continue from student-generated prefixes.* Figure 3(a) reveals a pronounced recovery bottleneck as the student prefix becomes longer. The teacher's Pass@1 accuracy decreases from 48.3% when generating the solution independently to 26.3% when continuing from an 8K-token student prefix. At the same time, the total number of generated tokens increases from 9.8K to 18.2K, consistent with the teacher expending substantially more computation to repair or work around the preceding student reasoning, yet often remaining unable to recover a correct solution. This highlights teacher recoverability as a critical, yet often overlooked, bottleneck in OPD: beyond solving problems well on its own, the teacher must remain reliable when providing supervision on student-induced states.

Recipe. Our second recipe introduces ***teacher recovery training***, which optimizes the teacher to improve its behavior after entering a student-generated state. Specifically, for each prompt $x \sim \mathcal{D}$, we first sample a student trajectory $\hat{y} \sim \pi_\theta(\cdot \mid x)$. We then sample a prefix length $\ell \in \{0, \ldots, L_{\max}\}$, where $L_{\max}$ denotes the maximum prefix length considered during recovery training, and construct the corresponding student state $s_{\ell+1} = (x, \hat{y}_{\le \ell})$. Conditioned on this state, the teacher samples a group of G_{II} continuations, denoted as $\mathcal{Y}_T(x, \hat{y}_{\le \ell}) = \{y^{(i)}_{\ell+1:}\}_{i=1}^{G_{\text{II}}}$. Each continuation is combined with the student prefix to form a complete trajectory $y^{(i)} = [\hat{y}_{\le \ell}; y^{(i)}_{\ell+1:}]$. Each completed trajectory receives the outcome reward $R(x, y^{(i)}) \in [0, 1]$. Analogous to Recipe I, we optimize the teacher with reinforcement learning using the group-normalized advantage

$$\hat{A}_{\text{rec}}\big(x, \hat{y}_{\le \ell}, y^{(i)}_{\ell+1:}\big) = \frac{R\big(x, y^{(i)}\big) - \text{mean}\left(\left\{R\big(x, y^{(j)}\big)\right\}_{j=1}^{G_{\text{II}}}\right)}{\text{std}\left(\left\{R\big(x, y^{(j)}\big)\right\}_{j=1}^{G_{\text{II}}}\right)}. \tag{9}$$

The teacher is then updated with the same policy optimization objective described in Appendix D, while the optimization signal is applied to the teacher continuation rather than the student prefix.

Training across different prefix lengths exposes the teacher to a range of student states, from relatively early states that remain close to the original prompt to later states that contain more accumulated student decisions. This directly targets the failure mode observed in Figure 3(a): rather than

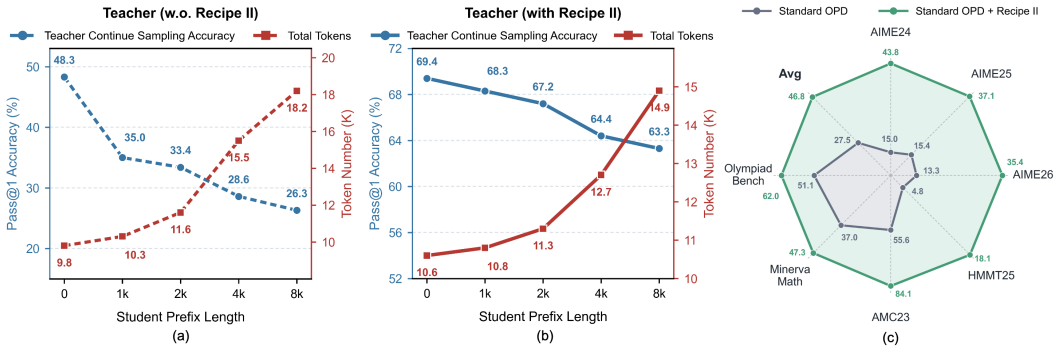

Figure 3: **Teacher recoverability and downstream effect of Recipe II. (a&b)** Teacher recoverability before and after recovery training, measured by Pass@1 accuracy and total generated tokens when continuing from student generated prefixes of increasing length. Recipe II substantially improves continuation accuracy and mitigates the degradation under longer prefixes, while reducing token consumption. **(c)** Downstream impact of Recipe II on OPD, evaluated by per-benchmark student Pass@1 accuracy.

only solving each problem independently, the teacher is trained to remain effective after inheriting the student's partial trajectory.

Setup. We adopt the same configuration as Recipe I: Qwen3-4B-Non-Thinking as the teacher, Qwen3-1.7B-Non-Thinking as the student, and DAPO-17K as the OPD training data.

Results. Our experiments indicate that teacher recovery training substantially strengthens the teacher's reliability on student-induced states, incidentally reinforces its independent problem-solving, and that these gains transfer directly into downstream student performance. ***First,*** recovery training flattens the degradation of the teacher's continue-sampling accuracy as the student prefix lengthens. Comparing Figure 3(a) and (b), without recovery training the accuracy degrades by 22.0 points as the prefix grows to 8K tokens (from 48.3% to 26.3%); with recovery training, the degradation shrinks to only 6.1 points (from 69.4% to 63.3%). ***Second,*** recovery training makes the recovery process itself cheaper. Without recovery training, the completed trajectory length grows from 9.8K to 18.2K tokens as the prefix lengthens, reflecting lengthy yet ineffective self-correction attempts; with recovery training, this growth is contained from 10.6K to 14.9K tokens, indicating that the recovery-trained teacher can restore correctness with fewer tokens. ***Third,*** recovery training also yields a side benefit on the teacher's independent problem-solving. At the empty prefix, which corresponds to solving the problem from scratch, accuracy rises from 48.3% to 69.4% (+21.1 points). ***Fourth,*** these reliability gains transfer to the student. As shown in Figure 3(c), the recovery-trained teacher improves the student on all seven benchmarks, raising the average Pass@1 accuracy from 27.5% to 46.8% (+19.3 points).

5 Recipe III: Decoding-Aware KL Objective

We find a fundamental train-inference mismatch in the alignment objective of standard OPD: it aligns raw next-token distributions, whereas actual generation is governed by the sampling distributions shaped by decoding strategies. Our third recipe closes this gap with a decoding-aware KL objective that incorporates the decoding strategy into teacher student alignment. Intuitively, OPD should align what the model actually samples, rather than only what it nominally outputs.

Motivation. Our motivation stems from how this mismatch arises: standard OPD constructs its token-level supervision from the raw next-token distributions of the two models, whereas generation samples from distributions reshaped by decoding strategies such as temperature scaling, top-k filtering, or top-p filtering. These transformations can substantially alter both the probability mass and the retained token set, so alignment established before decoding does not necessarily imply alignment during actual generation. Since OPD learns on trajectories sampled from these post-decoding distributions, the alignment objective should account for the decoding strategy.

Recipe. Our third recipe performs teacher student alignment on the distributions that actually govern token sampling. At a student generated state $s_t = (x, \hat{y}_{<t})$, let p_t and q_t denote the raw student and teacher next token distributions. Given decoding transformations $\mathcal{G}_S$ and $\mathcal{G}_T$, we define the corresponding sampling distributions as $p_t^{\text{dec}} = \mathcal{G}_S(p_t)$ and $q_t^{\text{dec}} = \mathcal{G}_T(q_t)$. Recipe III replaces the pre decoding probabilities in standard OPD with these post decoding probabilities. When hard truncation produces different supports, we additionally expand the teacher support with the student tokens involved in the corresponding OPD formulation.

- ***TopK-Vocabulary Decoding-Aware KL.*** Let $\mathcal{P}_t^K$ denote the top-K student candidate set selected from the decoded sampling distribution p_t^{dec}, and let $\mathcal{S}_t^T = \text{supp}(q_t^{\text{dec}})$ denote the decoded teacher support. We expand the teacher support as $\overline{\mathcal{S}}_t^K = \mathcal{S}_t^T \cup \mathcal{P}_t^K$. Let $\widetilde{p}_t^{(\mathcal{P}_t^K)}$ denote the student sampling distribution renormalized on $\mathcal{P}_t^K$, and let $\overline{q}_t^{(\overline{\mathcal{S}}_t^K)}$ denote the teacher distribution renormalized on the expanded support using its probabilities before hard truncation. Treating the student distribution as zero outside $\mathcal{P}_t^K$, the decoding-aware TopK-Vocabulary KL is

$$\text{KL}_{\text{TopK}}^{\text{Ours}}\big(p_t \,\|\, q_t\big) = \sum_{v \in \mathcal{P}_t^K} \widetilde{p}_t^{(\mathcal{P}_t^K)}(v) \log \frac{\widetilde{p}_t^{(\mathcal{P}_t^K)}(v)}{\overline{q}_t^{(\overline{\mathcal{S}}_t^K)}(v)}. \tag{10}$$

- ***Sampled-Token Decoding-Aware KL.*** For Sampled-Token OPD, let $\mathcal{S}_t^S = \text{supp}(p_t^{\text{dec}})$ and $\mathcal{S}_t^T = \text{supp}(q_t^{\text{dec}})$ denote the decoded student and teacher supports. We expand the teacher support with the entire student support, $\overline{\mathcal{S}}_t = \mathcal{S}_t^T \cup \mathcal{S}_t^S$, and renormalize the teacher probabilities over $\overline{\mathcal{S}}_t$ to obtain $\overline{q}_t$. For a student token $\hat{y}_t \sim p_t^{\text{dec}}$, we define

$$d_t^{\text{Ours}} = \log p_t^{\text{dec}}(\hat{y}_t) - \log \overline{q}_t(\hat{y}_t), \qquad \text{KL}_{\text{Sampled}}^{\text{Ours}}\big(p_t \,\|\, q_t\big) = \mathbb{E}_{\hat{y}_t \sim p_t^{\text{dec}}}\left[d_t^{\text{Ours}}\right]. \tag{11}$$

As in the standard Sampled-Token formulation, d_t^{Ours} is a pointwise log ratio rather than a KL divergence by itself. Since the expanded teacher distribution depends on $\hat{y}_t$, the expectation in Equation Eq. (11) is more precisely a decoding-aware sampled-token objective rather than a KL divergence between two fixed distributions.

Setup. We follow the same experimental configuration as Recipes I and II.

Results. Our results show that accounting for the decoding strategy improves both teacher student alignment during OPD and downstream student performance. ***First,*** Recipe III produces consistently stronger alignment between the support sets that actually govern generation. As shown in Figure 4(a), both methods begin from a similar support overlap of approximately 0.892, but the decoding-aware objective quickly establishes a persistent advantage throughout training, reaching roughly 0.897–0.898 compared with about 0.895 for standard OPD. This indicates that explicitly optimizing the post-decoding distributions reduces the train-inference mismatch targeted by Recipe III. ***Second,*** the improved alignment translates into broad downstream gains. As shown in Figure 4(b), Recipe III increases the average Pass@1 accuracy from 27.5% to 31.4% (+3.9 points) and improves performance on all seven benchmarks. Together, these results support the central principle of Recipe III: aligning the distributions that govern actual sampling provides a more effective supervision signal than aligning only their pre-decoding counterparts.

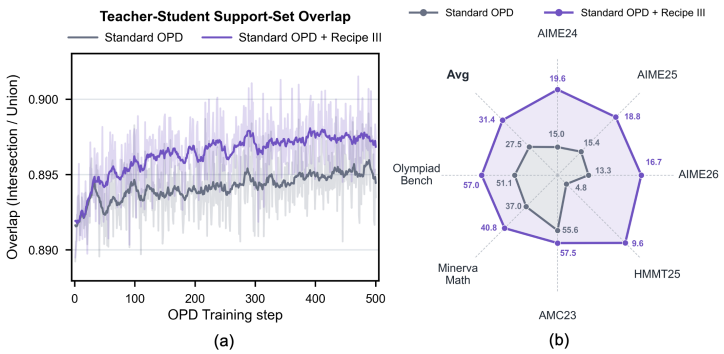

Figure 4: **Effect of Recipe III. (a)** Teacher-student support-set overlap during OPD training, measured by $|\mathcal{S}_S \cap \mathcal{S}_T|/|\mathcal{S}_S \cup \mathcal{S}_T|$. Recipe III consistently improves the alignment between the post-decoding supports that govern token sampling. **(b)** Downstream impact of Recipe III on OPD, evaluated by per-benchmark student Pass@1 accuracy.

Table 1: Comparison with different training algorithms.

Method	AIME24	AIME25	AIME26	HMMT25	AMC23	MineraMath	OlympiadBench	Avg
Student: Qwen3-1.7B-Non-Thinking *Teacher: Qwen3-4B-Non-Thinking*								
Baseline	12.9	12.5	13.8	4.8	47.5	33.7	41.1	23.8
+ GRPO	32.1	29.6	27.5	11.5	73.4	41.2	60.0	39.3
+ OPD	15.0	15.4	13.3	4.8	55.6	37.0	51.1	27.5
+ OPD + Recipes I & II & III	**48.3**	**39.6**	**37.9**	**21.7**	**85.9**	**49.1**	**64.0**	**49.5**
Δ (over OPD)	**(+33.3)**	**(+24.2)**	**(+24.6)**	**(+16.9)**	**(+30.3)**	**(+12.1)**	**(+12.9)**	**(+22.0)**
Student: Qwen3-1.7B-Thinking *Teacher: Qwen3-4B-Thinking*								
Baseline	46.7	35.4	36.7	26.0	85.6	49.5	57.8	48.2
+ GRPO	48.8	37.1	37.5	25.8	85.6	48.9	59.8	49.1
+ OPD	50.0	37.5	38.8	26.1	85.9	49.2	60.5	49.7
+ OPD + Recipes I & II & III	**55.0**	**42.9**	**44.2**	**29.3**	**87.2**	**51.0**	**66.3**	**53.7**
Δ (over OPD)	**(+5.0)**	**(+5.4)**	**(+5.4)**	**(+3.2)**	**(+1.3)**	**(+1.8)**	**(+5.8)**	**(+4.0)**

6 Experiments

6.1 Implementation Details

Joint Training of Recipes I and II. Recipes I and II both optimize the teacher with outcome based reinforcement learning and can therefore be combined into a single training stage. Specifically, each teacher training batch contains an equal mixture of trajectories generated independently by the teacher for Recipe I and teacher continuations from student generated prefixes for Recipe II. This joint training simultaneously improves the teacher's task proficiency and its recoverability on student induced states without requiring two separate teacher training procedures. Further implementation details are provided in Appendix B.2.

Setup. We defer the full experimental configuration, including model settings, training hyperparameters, decoding settings, and evaluation details, to Appendix B.1.

6.2 Main Results

Table 1 compares the complete OPD Cookbook against standard OPD and outcome-based RL (GRPO) under two student-teacher configurations.

Substantial gains over standard OPD. Incorporating all three recipes raises the average Pass@1 accuracy from 27.5 to 49.5 under the Non-Thinking setup, a +22.0 point gain over OPD with every benchmark improving by +12.1 to +33.3 points. The Thinking setup shows the same trend, rising from 49.7 to 53.7.

From fragile OPD to surpassing RL. In the Non-Thinking setup, standard OPD trails GRPO by a wide margin (27.5 versus 39.3) and even regresses on AIME26 (13.3 versus a baseline of 13.8). The Cookbook reverses this ordering, reaching 49.5 and exceeding GRPO by 10.2 points.

Robustness across reasoning paradigms. Under the Thinking setup, the strong baseline (48.2) leaves GRPO and OPD nearly saturated (+0.9 and +1.5), yet the Cookbook still adds +4.0 over OPD with gains on all seven benchmarks. The benefits therefore depend neither on the reasoning paradigm nor on a weak starting point.

6.3 Ablation Study

Appendix C provides a detailed ablation of recipe combinations, confirming that the three recipes are strictly complementary: their gains stack monotonically, and the full Cookbook strictly outperforms every individual or pairwise combination.

7 Conclusion

We revisit On-Policy Distillation from its underlying mechanism and identify two fundamental factors that govern its effectiveness: the reliability of teacher supervision and the formulation of the teacher student alignment objective. Guided by this decomposition, we develop an OPD Cookbook with three complementary recipes that improve teacher proficiency on the OPD data, strengthen teacher recoverability on student induced states, and align the sampling distributions that actually govern generation. Across both Non-Thinking and Thinking settings, the proposed recipes consistently improve standard OPD, with the strongest performance obtained when they are combined.

REFERENCES

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

AIME. American invitational mathematics examination, 2024. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

AIME. American invitational mathematics examination, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

AIME. American invitational mathematics examination, 2026. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

AMC. American mathematics competitions, 2023. URL https://artofproblemsolving.com/wiki/index.php/2023_AMC_12A.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *arXiv preprint arXiv:2605.00674*, 2026.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes, 2026. URL https://arxiv.org/abs/2603.25562.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

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

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026. URL https://arxiv.org/abs/2604.13016.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. Demystifying OPD: Length inflation and stabilization strategies for large language models, 2026. URL https://arxiv.org/abs/2604.08527.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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

Thinking Machines Lab and Kevin Lu. On-policy distillation, October 2025. URL https://thinkingmachines.ai/blog/on-policy-distillation/. Published October 27, 2025.

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

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64784.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes, 2026. URL https://arxiv.org/abs/2605.11182.

Broader Impact

This paper presents a practical framework for improving the stability and effectiveness of on-policy distillation. By enabling smaller language models to learn more reliably from stronger teachers on student-generated trajectories, the proposed recipes may facilitate the development of more capable and computationally accessible reasoning systems. This may reduce the cost of deploying advanced language models and broaden access to their benefits. At the same time, more effective distillation can transfer not only useful capabilities but also biases, factual errors, unsafe behaviors, and potentially harmful capabilities inherited from the teacher. The additional teacher adaptation and recovery training may also increase computational and environmental costs. These risks call for careful teacher selection, safety evaluation, transparent reporting, and appropriate governance before distilled models are deployed in high-stakes or socially sensitive settings.

Limitations

Although the proposed OPD Cookbook is theoretically motivated and empirically effective, the present evaluation remains limited in scope. Our experiments focus on Qwen3 students and teachers, the DAPO-17K training set, and seven mathematical reasoning benchmarks; therefore, the conclusions may not directly generalize to other model families, modalities, domains, or substantially larger scales. Recipes I and II require additional teacher optimization, access to suitable student-generated states, and reliable outcome signals, which introduce extra computational cost and may be difficult to obtain for tasks without verifiable rewards. Recipe III assumes access to teacher and student token probabilities and must be configured for the decoding rule used at deployment; its behavior under different tokenizers or rapidly changing decoding policies remains underexplored. Furthermore, adapting a teacher on the target distribution may amplify dataset-specific biases or overfitting despite validation-based checkpoint selection. Future work should evaluate these recipes across broader model families and tasks, quantify their compute–performance trade-offs, and study robustness under imperfect teachers, cross-tokenizer settings, and non-verifiable objectives.

AI-Usage Statement

In the preparation of this manuscript, artificial intelligence (AI) tools were employed exclusively for language polishing, grammatical correction, and debugging of specific code segments. These tools served solely as editorial and technical assistants to enhance clarity and correctness; they were not involved in any aspect of conceptualization, hypothesis generation, experimental design, literature synthesis, or autonomous research. All core intellectual contributions, including the formulation of research questions, methodological development, data interpretation, and scholarly argumentation, are entirely the product of human authorship. The authors affirm full responsibility for the originality, accuracy, and integrity of all content presented herein.

A Related Work

From offline sequence distillation to learner-state supervision. Conventional sequence-level knowledge distillation trains a student on complete sequences generated by a fixed teacher, so the supervision distribution is determined before student optimization rather than by the student's current behavior (Kim & Rush, 2016). This can simplify a multimodal sequence-prediction problem, but it does not directly supervise the states that the evolving student will visit at inference time. The underlying distribution-shift issue is closely related to imitation learning. In particular, DAgger collects states visited by a learner-dependent mixture policy and queries an expert on those states, providing an early algorithmic precedent for learner-state supervision (Ross et al., 2011). In autoregressive language modeling, strict on-policy distillation instantiates this principle by sampling trajectories from the current student and evaluating the teacher on the resulting student-generated prefixes. This state-distribution distinction, rather than the particular divergence or optimizer, separates OPD from offline response distillation.

On-policy distillation for language models. Two ICLR 2024 methods established complementary formulations of this idea. Generalized Knowledge Distillation (GKD) trains on self-generated

student sequences, queries the teacher distribution on those prefixes, and allows different divergence choices within a common framework (Agarwal et al., 2024). MiniLLM instead formulates large-language-model distillation through sequence-level reverse KL and develops a policy-gradient-style estimator using student samples (Gu et al., 2024). DistiLLM further combines student-generated outputs with a skew KL objective and an adaptive off-policy mechanism that reuses those outputs, illustrating a hybrid trade-off between fresh on-policy data and sample efficiency (Ko et al., 2024). These methods share the goal of reducing the mismatch between the training distribution and the student's own generation distribution, but they differ in whether teacher information is consumed as full token distributions, sequence-level likelihood ratios, or mixtures of fresh and reused samples.

A recent practical formulation computes the reverse-KL log-ratio only at the token sampled by the student and uses its negative as a token-level advantage in a PPO-like update (Thinking Machines Lab & Lu, 2025). This sampled-token estimator substantially reduces the cost of transferring a full teacher distribution, but it also exposes optimization to the particular tokens and prefixes visited by the student. On-policy self-distillation has additionally shown that the supervising policy need not be a permanently separate model: a shared model can act as a teacher under privileged reasoning context and as a student under the original problem context, while still learning on student-generated rollouts (Zhao et al., 2026). Thus, "on-policy" specifies where supervision is evaluated, while teacher construction and divergence estimation remain independent design choices.

Relation to reinforcement learning. OPD can share optimization machinery with reinforcement learning without sharing its learning signal. PPO and related group-based policy optimization methods update a policy from sampled trajectories using scalar reward-derived advantages (Schulman et al., 2017; Shao et al., 2024); OPD instead obtains dense token-level supervision by comparing the student and teacher on those trajectories. Full-distribution OPD can minimize a divergence directly, whereas sampled estimators can express a teacher-student log-ratio as an advantage and reuse a policy-gradient implementation. The two paradigms are therefore composable but not interchangeable: reward optimization determines which outcomes are preferred, while distillation specifies how a teacher's conditional behavior should be transferred on the states visited by the student.

Failure modes and stabilization. Recent concurrent studies have challenged the assumption that placing a standard KL objective on student rollouts is sufficient for reliable improvement. They report sensitivity to teacher-student compatibility, unreliable or imbalanced supervision, response-length inflation, and the exact form of the distillation objective, and propose corresponding warm-up, filtering, or stabilization strategies (Li et al., 2026; Fu et al., 2026; Luo et al., 2026; Zhu et al., 2026). Collectively, these results motivate treating OPD as a coupled system whose behavior depends on the teacher, the student-induced state distribution, and the estimator used to transfer supervision, rather than as a drop-in replacement for supervised fine-tuning.

Positioning of this work. The representative methods above predominantly vary the student-side divergence, gradient estimator, or balance between on-policy sampling and data reuse. Our study instead organizes practical OPD around two complementary dimensions: the reliability of the teacher supervision available on student-induced states, and the faithfulness of the alignment objective to the distribution that actually governs generation. Recipes I and II adapt the teacher on the target OPD distribution and explicitly train it to recover from imperfect student prefixes. Recipe III transforms both policies through the deployed decoding rule before alignment and resolves support mismatches introduced by truncation. Among the representative methods reviewed here, we did not identify one jointly centered on teacher adaptation, teacher recovery, and decoding-aware KL; our contribution is therefore a systematic empirical cookbook for these three complementary interventions rather than a new definition of OPD.

B Implementation Details

B.1 Experimental Setup

Models. We evaluate the OPD Cookbook under two reasoning settings based on the Qwen3 model family (Yang et al., 2025). In the *Non-Thinking* setting, we use Qwen3-1.7B-Non-Thinking as the student and Qwen3-4B-Non-Thinking as the teacher. In the *Thinking* setting, we use Qwen3-1.7B-

Thinking as the student and Qwen3-4B-Thinking as the teacher. This allows us to examine whether the proposed recipes remain effective across different levels of reasoning depth while keeping the student and teacher model scales fixed.

Training Dataset. All experiments are conducted on DAPO-17K (**?**). The same training data are used across different methods unless otherwise specified, enabling controlled comparisons among GRPO, standard OPD, and the proposed OPD Cookbook.

Evaluation. We evaluate model performance on seven challenging mathematical reasoning benchmarks: AIME24 (AIME, 2024), AIME25 (AIME, 2025), AIME26 (AIME, 2026), HMMT25 (Dekoninck et al., 2026), AMC23 (AMC, 2023), MinervaMath (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). For each problem, we independently sample 16 responses using a sampling temperature of 0.6, top-$p = 0.95$, top-$k = 20$, and a maximum generation length of 32K tokens. We compute the average correctness over these samples and report Pass@1 accuracy by averaging this quantity across all problems in each benchmark. The seven benchmarks cover a range of competition-level and general mathematical reasoning problems, providing a broad evaluation of downstream reasoning performance.

Training Settings. Unless otherwise specified, all experiments use a batch size of 256, with one trajectory sampled for each training problem. We optimize the models with AdamW using a constant learning rate of 1×10^{-6} and a KL regularization coefficient of 1×10^{-3}. During training, trajectories are sampled with temperature 0.6, top-p 0.95, and top-k 20, with the maximum generation length set to 16K tokens. Each experiment is trained for 500 optimization steps on eight NVIDIA H200 GPUs.

B.2 Implementation Details for OPD Cookbook

Recipes I and II both optimize the teacher through outcome-based reinforcement learning, but they differ in the states from which the teacher generates its trajectories. Recipe I trains the teacher to solve the OPD prompts independently, while Recipe II trains the teacher to continue from states induced by the student. When applying the two recipes together, we therefore combine them into a single teacher RL stage rather than performing two sequential training procedures.

Concretely, each teacher training batch contains an equal mixture of two trajectory types. For half of the samples, the teacher generates a complete trajectory directly from the prompt,

$$y \sim \pi_T(\cdot \mid x), \tag{12}$$

which corresponds to the teacher adaptation training in Recipe I. For the other half, we first sample a trajectory $\hat{y} \sim \pi_\theta(\cdot \mid x)$ from the student and select a prefix length ℓ. The teacher then continues generation from the resulting student state $s_{\ell+1} = (x, \hat{y}_{\le \ell})$, which corresponds to the recovery training in Recipe II. The completed trajectory is formed by concatenating the student prefix and teacher continuation.

Both trajectory types are optimized with the same verifiable outcome reward and policy optimization framework. Equivalently, the joint teacher objective can be written as

$$\mathcal{J}_{\text{Teacher}} = \lambda \mathcal{J}_{\text{Adapt}} + (1-\lambda)\mathcal{J}_{\text{Recovery}}, \qquad \lambda = 0.5, \tag{13}$$

where $\mathcal{J}_{\text{Adapt}}$ is computed from trajectories independently generated by the teacher, and $\mathcal{J}_{\text{Recovery}}$ is computed from teacher continuations conditioned on student-generated prefixes. For the recovery samples, only tokens generated by the teacher contribute to the policy optimization objective, while the student prefix is treated as fixed context.

This mixed training strategy allows the two teacher side recipes to complement each other within the same optimization process. The independently generated trajectories improve the teacher's task proficiency on the OPD data, whereas the student-conditioned trajectories expose the teacher to states that it will encounter when supervising the student during OPD. In our implementation, the two trajectory types are sampled with equal probability, although their mixing ratio can be adjusted when different emphasis on adaptation and recovery is desired.

After joint teacher training, we select the teacher checkpoint using the validation-based procedure described in Section 3. The selected checkpoint is then used for downstream OPD. Recipe III is applied at this stage by replacing the corresponding standard TopK Vocabulary or Sampled Token

Table 2: Ablation study on the combinations of the three recipes across complex reasoning benchmarks. All models are trained on the DAPO-17K dataset.

Method	AIME24	AIME25	AIME26	HMMT25	AMC23	MineraMath	OlympiadBench	Avg
Student: Qwen3-1.7B-Non-Thinking *Teacher: Qwen3-4B-Non-Thinking*								
OPD	15.0	15.4	13.3	4.8	55.6	37.0	51.1	27.5
+Recipe I	41.7	35.0	33.3	17.1	83.1	46.6	61.3	45.4
+Recipe II	43.8	37.1	35.4	18.1	84.1	47.3	62.0	46.8
+Recipe III	19.6	18.8	16.7	9.6	57.5	40.8	57.0	31.4
+Recipe I & II	45.4	37.9	36.3	20.0	84.7	48.5	62.7	47.9
+Recipe I & III	43.3	36.3	34.6	19.2	83.8	47.8	62.3	46.8
+Recipe II & III	46.3	38.3	36.3	20.8	85.0	48.4	63.0	48.3
+Recipe I & II & III	**48.3**	**39.6**	**37.9**	**21.7**	**85.9**	**49.1**	**64.0**	**49.5**
Δ (over OPD)	**(+33.3)**	**(+24.2)**	**(+24.6)**	**(+16.9)**	**(+30.3)**	**(+12.1)**	**(+12.9)**	**(+22.0)**
Student: Qwen3-1.7B-Thinking *Teacher: Qwen3-4B-Thinking*								
OPD	50.0	37.5	38.8	26.1	85.9	49.2	60.5	49.7
+Recipe I	52.0	39.2	40.0	27.1	86.1	49.8	61.9	50.9
+Recipe II	52.9	40.4	41.3	27.6	86.4	50.2	62.6	51.6
+Recipe III	53.3	41.3	42.9	27.6	86.4	50.2	64.2	52.3
+Recipe I & II	53.8	40.8	42.0	28.0	86.5	50.4	63.9	52.2
+Recipe I & III	54.2	42.5	42.5	28.4	86.7	50.7	65.2	52.9
+Recipe II & III	54.6	42.5	43.3	28.7	87.0	50.7	65.9	53.2
+Recipe I & II & III	**55.0**	**42.9**	**44.2**	**29.3**	**87.2**	**51.0**	**66.3**	**53.7**
Δ (over OPD)	**(+5.0)**	**(+5.4)**	**(+5.4)**	**(+3.2)**	**(+1.3)**	**(+1.8)**	**(+5.8)**	**(+4.0)**

alignment with the decoding-aware formulation in Section 5. Therefore, the complete OPD Cookbook requires one joint teacher RL stage followed by one OPD stage:

$$\boxed{\text{Joint Teacher RL with Recipes I and II} \longrightarrow \text{OPD with Recipe III.}} \tag{14}$$

C Ablation Study

Combination effects of the recipes. Table 2 reports the performance of every subset of the three recipes under both student-teacher configurations. We summarize the findings as follows.

- **Each recipe is individually beneficial.** Every single recipe improves over standard OPD in both configurations: under the Non-Thinking setup, Recipes I, II, and III raise the average accuracy by +17.9, +19.3, and +3.9 points, respectively, and under the Thinking setup by +1.2, +1.9, and +2.6 points, confirming that each recipe addresses a genuine bottleneck of standard OPD.
- **Combinations improve monotonically without interference.** Adding any recipe to an existing subset never degrades performance: in the Non-Thinking setup, each two-recipe combination exceeds both of its single-recipe constituents (e.g., Recipe I & II reaches 47.9 versus 45.4 and 46.8), and the full combination further reaches 49.5, exceeding the best two-recipe subset (48.3); the Thinking setup exhibits the same monotone ordering, culminating at 53.7.
- **The two dimensions are orthogonal and complementary across regimes.** The marginal contribution of Recipe III, which targets the alignment objective, remains nearly constant when added on top of the teacher-side recipes (+1.6 points over Recipe I & II in the Non-Thinking setup and +1.5 points in the Thinking setup), even though the individual gains of the teacher-side recipes shrink by an order of magnitude across the two regimes (from +17.9 and +19.3 to +1.2 and +1.9). Conversely, in the Thinking setup where teacher-side headroom is limited, Recipe III alone (+2.6) surpasses either teacher-side recipe alone (+1.2 and +1.9). The relative importance of the two dimensions thus shifts with the regime while their complementarity persists, making the full combination necessary in both.

D PPO Objective in Sampled-Token OPD

When deploying Sampled-Token KL within a Reinforcement Learning paradigm, the point-wise divergence acts as the guiding advantage signal. Formally, at the k-th training iteration, given a prompt $x \sim \mathcal{D}$, the behavior policy (i.e., the old student model) $\pi_{\theta_{\text{old}}}$ rollouts a group of G responses,

denoted as $\mathcal{Y}^k(x) = \{\hat{y}^{(i)}\}_{i=1}^{G}$. The surrogate optimization objective for PPO is formulated as:

$$\mathcal{J}(\theta) = \mathbb{E}_{\hat{y}\in\mathcal{Y}^k(x)}\left[\hat{A}(x,\hat{y}) \cdot \min\Big(\rho(x,\hat{y}),\ \mathrm{clip}\big(\rho(x,\hat{y}),\ 1-\epsilon,\ 1+\epsilon\big)\Big)\right] - \beta\, \mathbb{D}_{\mathrm{KL}}\big(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big), \tag{15}$$

where $\rho(x,\hat{y}) = \frac{\pi_\theta(\hat{y}|x)}{\pi_{\theta_{\mathrm{old}}}(\hat{y}|x)}$ calculates the importance sampling ratio, which corrects the distributional shift between the actively updating policy π_θ and the sampling policy $\pi_{\theta_{\mathrm{old}}}$.

To guarantee optimization stability and prevent catastrophic policy degradation, the objective enforces two critical regularization mechanisms:

- **Pessimistic Clipping:** The operator $\mathrm{clip}(\cdot, 1-\epsilon, 1+\epsilon)$ bounds the per-step policy update magnitude, preventing excessively large gradient steps when the importance ratio diverges significantly from 1.
- **Reference Regularization:** A sequence-level KL penalty, scaled by a strictness coefficient β, anchors the learning policy toward a frozen reference model π_{ref} (typically the SFT checkpoint initialized before RL), thereby mitigating reward hacking.

Crucially, in the context of On-Policy Distillation, the standard reward-derived advantage $\hat{A}(x,\hat{y})$ is substituted or directly augmented by the teacher-guided signal $\mathrm{KL}^{\mathrm{OPD}}_{\mathrm{Sampled}}$ defined in Equation 7. This mechanism equips the PPO algorithm with rigorous, token-by-token expert supervision without necessitating a separately trained reward model.

E A Unified Theoretical Analysis of the OPD Cookbook

This section provides a unified theoretical framework for the three recipes comprising the OPD Cookbook. Our central result decomposes the final student error into three distinct factors, corresponding to intrinsic teacher quality, teacher reliability on student-generated states, and decoding-aware teacher-student alignment.

Organization. The analysis proceeds from the main theoretical conclusions to their formal derivations:

- Section E.1 introduces the unified performance decomposition, elucidates its three components, and establishes their direct correspondence with the proposed recipes.
- Sections E.2 through E.4 provide focused analyses demonstrating how Teacher Adaptation, Teacher Recovery Training, and the Decoding-Aware KL objective specifically mitigate their corresponding bottlenecks in the decomposition.
- Section E.5 formally defines the continuation values and recovery quantities required for the complete proof, derives the decomposition step-by-step, and discusses its theoretical scope.

E.1 Unified OPD Performance Decomposition

We begin by characterizing the sources of error in the final student policy. Let π_S and π_T denote the sequential student and teacher policies under the decoding rules used for actual generation, and let H denote the maximum response length. For a bounded sequence-level reward $R(x,y) \in [0,1]$, define the expected performance of a policy π as

$$J(\pi) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi(\cdot|x)}\left[R(x,y)\right]. \tag{16}$$

Let $J^\star$ denote the optimal performance within a reference policy class. We define the student and teacher suboptimalities as

$$\Delta_S = J^\star - J(\pi_S), \qquad \Delta_T = J^\star - J(\pi_T). \tag{17}$$

Thus, Δ_S measures how far the final student remains from the reference optimum, while Δ_T measures the intrinsic performance gap of the teacher before accounting for its subsequent transfer to the student. To expose how teacher quality and policy transfer jointly constrain the final student, we decompose Δ_S as follows.

Proposition E.1 (Unified OPD performance decomposition)**.** *Consider a sequential generation process with maximum response length H. Let $\epsilon^{\mathrm{dec}}_{S,T}$ denote the average discrepancy between the student and teacher distributions that actually govern token sampling, evaluated over states visited by the student. Let $\overline{\kappa}_{T|S} \in [0,1]$ denote the mismatch-weighted average sequence-level performance loss caused by one unit of local disagreement between the student and teacher. Then the student suboptimality satisfies*

$$\boxed{\Delta_S \leq \Delta_T + H \cdot \overline{\kappa}_{T|S} \cdot \epsilon^{\mathrm{dec}}_{S,T}.} \tag{18}$$

The formal definitions of $\epsilon^{\mathrm{dec}}_{S,T}$ and $\overline{\kappa}_{T|S}$, together with the complete proof of Proposition E.1, are deferred to Section E.5. The mathematical structure of this decomposition builds upon classical on-policy imitation learning and policy performance difference bounds (Ross et al., 2011; Schulman et al., 2015). By extending these frameworks, our bound structurally isolates the final student error into two fundamental parts: the intrinsic error inherited from the teacher (Δ_T), and the additional divergence incurred during the policy transfer process ($H \cdot \overline{\kappa}_{T|S} \cdot \epsilon^{\mathrm{dec}}_{S,T}$).

Crucially, this transfer term reveals the compounding nature of sequential alignment. It explicitly models how local behavioral disagreements ($\epsilon^{\mathrm{dec}}_{S,T}$) accumulate over the generation horizon (H) and interact with the teacher's sensitivity to deviations ($\overline{\kappa}_{T|S}$). This decomposition directly exposes the three distinct bottlenecks in standard OPD, providing a unified theoretical foundation for our proposed recipes:

$$\boxed{\begin{array}{ll} \text{Recipe I} : \Delta_T \downarrow, & \\ \text{Recipe II} : \overline{\kappa}_{T|S} \downarrow & \text{when decoded alignment is controlled,} \\ \text{Recipe III} : \epsilon^{\mathrm{dec}}_{S,T} \downarrow . & \end{array}} \tag{19}$$

Recipe I improves the supervision source ($\Delta_T \downarrow$). Teacher Adaptation on OPD Data optimizes the teacher against the sequence-level task reward. By extracting the teacher's untapped performance headroom, this adaptation intrinsically lowers Δ_T. Consequently, it tightens the irreducible floor of the student's suboptimality bound, ensuring the student targets a superior optimal reference.

Recipe II improves teacher reliability on student states ($\overline{\kappa}_{T|S} \downarrow$). Teacher Recovery Training forces the teacher to navigate reasoning prefixes generated by the student. By learning to construct viable continuation trajectories from suboptimal states, the teacher mitigates the sequence-level penalty of localized student errors. Mathematically, this suppresses the sensitivity factor $\overline{\kappa}_{T|S}$, acting as a robust dampener on the cascading error propagation within the transfer term.

Recipe III improves teacher-student alignment ($\epsilon^{\mathrm{dec}}_{S,T} \downarrow$). The Decoding-Aware KL objective eliminates the mismatch between the training target and the inference behavior. By aligning the effective sampling distributions rather than nominal logits, this recipe ensures that the optimization directly minimizes $\epsilon^{\mathrm{dec}}_{S,T}$, thereby strictly compressing the base multiplier of the policy transfer error.

Equation Eq. (19) highlights the primary analytical quantity targeted by each recipe, though their effects in practice are deeply synergistic. For instance, teacher adaptation (Recipe I) may concurrently alter the teacher's recovery landscape ($\overline{\kappa}_{T|S}$). Sections E.2 through E.4 detail the precise mechanisms driving each of these reductions, while Section E.5 supplies the comprehensive derivation of the bound itself.

E.2 Theoretical Analysis of Recipe I: Bounding Teacher Suboptimality

This subsection formalizes how Recipe I targets the teacher suboptimality term Δ_T in Proposition E.1. Our analysis proceeds in three steps. First, we show that a population performance gain in the teacher produces an equal reduction in Δ_T, which motivates outcome based reinforcement learning as the adaptation mechanism. Second, we provide a finite sample guarantee for selecting among a fixed set of teacher checkpoints using an independent validation set. Finally, we derive a sufficient condition under which the reduction in teacher suboptimality outweighs any accompanying change in transfer difficulty and thereby tightens the final student performance bound.

E.2.1 A POPULATION TEACHER GAIN DIRECTLY REDUCES Δ_T

The key theoretical role of Recipe I is to improve the supervision source itself, thereby reducing the teacher suboptimality term in Proposition E.1. Let $\pi_T^{(0)}$ denote the original teacher and let $\pi_T^{(A)}$ denote an adapted teacher. Define the population performance gain of teacher adaptation as

$$\gamma_T = J\big(\pi_T^{(A)}\big) - J\big(\pi_T^{(0)}\big). \tag{20}$$

The following proposition establishes the direct connection between this adaptation gain and the teacher suboptimality.

Proposition E.2 (Reduction in teacher suboptimality)**.** *If teacher adaptation produces a population performance gain $\gamma_T > 0$, then*

$$\boxed{\Delta_T^{(A)} = \Delta_T^{(0)} - \gamma_T.} \tag{21}$$

Proof. By definition,

$$\Delta_T^{(0)} = J^\star - J\big(\pi_T^{(0)}\big), \tag{22}$$

and

$$\Delta_T^{(A)} = J^\star - J\big(\pi_T^{(A)}\big). \tag{23}$$

Using the definition of γ_T in Equation Eq. (20),

$$\begin{aligned}\Delta_T^{(A)} &= J^\star - J\big(\pi_T^{(A)}\big) \\ &= J^\star - J\big(\pi_T^{(0)}\big) - \Big[J\big(\pi_T^{(A)}\big) - J\big(\pi_T^{(0)}\big)\Big] \\ &= \Delta_T^{(0)} - \gamma_T. \end{aligned} \tag{24}$$

□

Proposition E.2 shows that the connection between Recipe I and Δ_T is algebraically exact. Teacher performance does not merely correlate with the first term in the unified decomposition; an improvement of γ_T in population task performance reduces that term by exactly γ_T.

This exact relationship motivates the use of outcome based reinforcement learning for teacher adaptation. At the population level, the idealized objective underlying Teacher Adaptation on OPD Data is

$$\max_{\pi_T} J(\pi_T) = \max_{\pi_T} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_T(\cdot|x)}\left[R(x,y)\right]. \tag{25}$$

Since $J^\star$ is fixed, maximizing $J(\pi_T)$ is equivalent to minimizing

$$\Delta_T = J^\star - J(\pi_T). \tag{26}$$

When the adaptation prompts are sampled from $\mathcal{D}$ and the training reward corresponds to $R(x, y)$, teacher rollouts provide samples for improving the same expected task performance that defines Δ_T. Algorithms such as GRPO and PPO provide practical policy optimization procedures for this objective. Their surrogate losses, however, do not by themselves guarantee monotonic improvement in the population performance $J(\pi_T)$.

By contrast, supervised fine tuning on fixed target responses optimizes a token level likelihood objective, which is not generally equivalent to maximizing the expected task reward in Equation Eq. (25). This distinction does not imply that reinforcement learning must outperform supervised fine tuning in every setting. Rather, it shows that the outcome based reinforcement learning objective is directly aligned with the teacher performance term in Proposition E.1.

E.2.2 A Finite Sample Guarantee for Validation Based Checkpoint Selection

The previous analysis concerns population performance, whereas practical teacher adaptation operates on a finite training set and may overfit. An empirical training gain therefore need not imply a reduction in the population quantity Δ_T. Recipe I uses an independent held out set to select among a fixed collection of teacher checkpoints. We next formalize the statistical role of this procedure under an idealized setting in which the validation prompts are independently sampled from $\mathcal{D}$ and the candidate checkpoint set is constructed without using those validation samples.

Suppose teacher adaptation produces M candidate checkpoints

$$\left\{\pi_T^{(1)}, \ldots, \pi_T^{(M)}\right\}, \tag{27}$$

where the original teacher is included among the candidates. We assume that this finite candidate set is fixed independently of the validation set. Let the validation set contain n independent prompts sampled from $\mathcal{D}$.

For a clean finite-sample guarantee, suppose the validation set contains n independent prompts sampled from $\mathcal{D}$. Let

$$Z_i^{(k)} \in [0,1] \tag{28}$$

denote the evaluation reward of checkpoint k on validation prompt i. Its empirical validation performance is

$$\widehat{J}_{\mathrm{val}}\left(\pi_T^{(k)}\right) = \frac{1}{n}\sum_{i=1}^{n} Z_i^{(k)}. \tag{29}$$

The selected checkpoint is

$$\widehat{k} = \arg\max_{1\le k\le M} \widehat{J}_{\mathrm{val}}\left(\pi_T^{(k)}\right). \tag{30}$$

Under these assumptions, the following result shows that validation based selection returns, with high probability, a checkpoint whose population performance lies within a finite sample error of the best candidate.

Proposition E.3 (Validation selection guarantee). *For any $\delta \in (0,1)$, define*

$$c_{n,M,\delta} = \sqrt{\frac{\log(2M/\delta)}{2n}}. \tag{31}$$

Then, with probability at least $1-\delta$,

$$\boxed{J\left(\pi_T^{(\widehat{k})}\right) \ge \max_{1\le k\le M} J\left(\pi_T^{(k)}\right) - 2c_{n,M,\delta}.} \tag{32}$$

Proof. For any fixed checkpoint k, Hoeffding's inequality gives

$$\Pr\left(\left|\widehat{J}_{\mathrm{val}}\left(\pi_T^{(k)}\right) - J\left(\pi_T^{(k)}\right)\right| > c_{n,M,\delta}\right) \le \frac{\delta}{M}. \tag{33}$$

Applying a union bound over all M candidate checkpoints gives, with probability at least $1-\delta$,

$$\left|\widehat{J}_{\mathrm{val}}\left(\pi_T^{(k)}\right) - J\left(\pi_T^{(k)}\right)\right| \le c_{n,M,\delta} \tag{34}$$

simultaneously for every k.

Let

$$k^\star = \arg\max_{1\le k\le M} J\left(\pi_T^{(k)}\right) \tag{35}$$

denote the best candidate according to population performance. On the event in Equation Eq. (34),

$$\begin{aligned} J\left(\pi_T^{(\widehat{k})}\right) &\ge \widehat{J}_{\mathrm{val}}\left(\pi_T^{(\widehat{k})}\right) - c_{n,M,\delta} \\ &\ge \widehat{J}_{\mathrm{val}}\left(\pi_T^{(k^\star)}\right) - c_{n,M,\delta} \\ &\ge J\left(\pi_T^{(k^\star)}\right) - 2c_{n,M,\delta}. \end{aligned} \tag{36}$$

The second inequality follows from the definition of $\widehat{k}$, while the first and third inequalities follow from uniform concentration. This proves Equation Eq. (32). □

Under the stated assumptions, Proposition E.3 shows that validation based selection returns a checkpoint whose population performance is close to that of the best candidate. The finite sample error decreases with the validation size n and depends only logarithmically on the number of candidate checkpoints M.

We next use this result to interpret the observed validation improvement over the original teacher. Let $\pi_T^{(0)}$ denote the original teacher and define the empirical validation gain as

$$\widehat{\gamma}_T = \widehat{J}_{\mathrm{val}}\big(\pi_T^{(\widehat{k})}\big) - \widehat{J}_{\mathrm{val}}\big(\pi_T^{(0)}\big). \tag{37}$$

On the same uniform concentration event,

$$J\big(\pi_T^{(\widehat{k})}\big) - J\big(\pi_T^{(0)}\big) \geq \widehat{\gamma}_T - 2c_{n,M,\delta}. \tag{38}$$

Therefore, if $\widehat{\gamma}_T > 2c_{n,M,\delta}$, the selected checkpoint has a positive population performance gain over the original teacher with probability at least $1-\delta$. Proposition E.2 then implies a corresponding reduction in Δ_T.

This analysis explains why adaptation training performance alone may be insufficient for checkpoint selection. Training performance can continue to increase after population performance has begun to deteriorate, whereas held out performance provides a more relevant estimate of the population quantity appearing in the unified bound.

The finite sample statement above applies specifically to a fixed candidate set constructed independently of iid validation prompts drawn from $\mathcal{D}$. If the validation results are repeatedly used to alter the training procedure or generate new candidate checkpoints, the independence assumption no longer holds directly. Likewise, when the practical validation set is a deliberately selected coreset rather than an iid sample, the result should be interpreted as an idealized justification for validation based selection rather than a literal finite sample guarantee for that coreset.

E.2.3 Sufficient conditions for tightening the student guarantee

Recipe I reduces Δ_T whenever adaptation produces a positive population gain. At the same time, adaptation may alter the other factors in Proposition E.1. In particular, the adapted teacher may move farther from the student policy, increasing the decoded policy mismatch, or its behavior on student generated states may change. We therefore derive the condition under which the teacher quality gain is sufficient to tighten the complete student upper bound.

Let $\pi_S^{(0)}$ denote the student trained with the original teacher, and let $\pi_S^{(A)}$ denote the student trained with the adapted teacher. The corresponding theoretical upper bounds are

$$\mathcal{B}_0 = \Delta_T^{(0)} + H \cdot \overline{\kappa}_{T^{(0)}|S^{(0)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(0)},T^{(0)}}, \tag{39}$$

and

$$\mathcal{B}_A = \Delta_T^{(A)} + H \cdot \overline{\kappa}_{T^{(A)}|S^{(A)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(A)},T^{(A)}}. \tag{40}$$

Using $\Delta_T^{(A)} = \Delta_T^{(0)} - \gamma_T$, their difference is given by

$$\begin{aligned}\mathcal{B}_A - \mathcal{B}_0 = &-\gamma_T \\ &+ H \cdot \left[\overline{\kappa}_{T^{(A)}|S^{(A)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(A)},T^{(A)}} - \overline{\kappa}_{T^{(0)}|S^{(0)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(0)},T^{(0)}}\right].\end{aligned} \tag{41}$$

It follows that Recipe I produces a tighter theoretical guarantee whenever

$$\boxed{\gamma_T > H \cdot \left[\overline{\kappa}_{T^{(A)}|S^{(A)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(A)},T^{(A)}} - \overline{\kappa}_{T^{(0)}|S^{(0)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(0)},T^{(0)}}\right].} \tag{42}$$

The condition has a direct interpretation: the improvement in teacher task quality must exceed any increase in the transfer term. This gives three cases.

Stable Transfer Difficulty. If

$$\overline{\kappa}_{T^{(A)}|S^{(A)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(A)},T^{(A)}} = \overline{\kappa}_{T^{(0)}|S^{(0)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(0)},T^{(0)}},$$

then any positive population teacher gain $\gamma_T > 0$ tightens the bound. If the transfer term changes slightly rather than remaining exactly fixed, the teacher gain must exceed the corresponding residual increase.

Improved Transfer Difficulty. If

$$\overline{\kappa}_{T^{(A)}|S^{(A)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(A)},T^{(A)}} \leq \overline{\kappa}_{T^{(0)}|S^{(0)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(0)},T^{(0)}},$$

then any positive population teacher gain tightens the bound. A decrease in either recovery sensitivity or decoded policy mismatch contributes to this case whenever an increase in the other factor does not offset that decrease.

Increased Transfer Difficulty. If

$$\overline{\kappa}_{T^{(A)}|S^{(A)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(A)},T^{(A)}} > \overline{\kappa}_{T^{(0)}|S^{(0)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(0)},T^{(0)}},$$

then higher teacher performance alone is not sufficient. The population teacher gain must exceed the increase in the transfer term, exactly as specified by Equation Eq. (42).

Equation Eq. (42) therefore avoids the stronger claim that a more accurate teacher must always yield a better student. Instead, it identifies when the teacher quality gain targeted by Recipe I dominates any accompanying change in transfer difficulty.

Overall, the theoretical role of Recipe I can be summarized as follows:

$$\boxed{\text{Teacher Adaptation} \quad \Longrightarrow \quad J(\pi_T) \uparrow \quad \Longrightarrow \quad \Delta_T \downarrow,} \tag{43}$$

provided that the adaptation gain generalizes beyond the training samples.

Validation based checkpoint selection estimates this generalizable improvement under the assumptions stated above, while Equation Eq. (42) characterizes when the teacher quality gain is sufficient to tighten the complete upper bound on student suboptimality.

E.3 Theoretical Analysis of Recipe II: Mitigating Recovery Sensitivity

This subsection formalizes how Recipe II targets the recovery related transfer term in Proposition E.1. Under controlled decoded alignment, this improvement can be further interpreted as a reduction in the recovery sensitivity $\overline{\kappa}_{T|S}$. Our analysis proceeds in four steps. First, we show that outcome based training from student generated prefixes directly targets the continuation values of student selected branches. Second, we derive a sufficient condition under which this intervention does not increase the transfer term in the unified bound. Third, we connect these theoretical quantities to the empirical recoverability profiles measured in our experiments. Finally, we provide a sufficient state coverage condition that clarifies when improvements learned from collected prefixes transfer to the states encountered during subsequent OPD.

E.3.1 Recovery Training Targets Student Induced Continuation Values

A teacher may perform well on trajectories generated from the original prompt while remaining unreliable when generation begins from an imperfect student prefix. We formalize this distinction using teacher continuation values.

Fix a student policy π_S. At step t, let $s_t = (x, y_{<t})$ be a state visited by the student. For a candidate token a, define the teacher action value as

$$Q^T_t(s_t, a) = \mathbb{E}\left[R(x, y) \,|\, s_t,\, a_t = a,\, a_{t+1:H} \sim \pi_T\right]. \tag{44}$$

This quantity is the expected final reward when token a is selected at state s_t and the teacher generates the remainder of the trajectory.

Let $q_t^{\mathrm{dec}}(\cdot \mid s_t)$ and $p_t^{\mathrm{dec}}(\cdot \mid s_t)$ denote the teacher and student distributions that actually govern token sampling. The teacher value at state s_t is

$$V_t^T(s_t) = \mathbb{E}_{a \sim q_t^{\mathrm{dec}}(\cdot|s_t)} \left[Q_t^T(s_t, a) \right]. \tag{45}$$

It measures the expected final reward when the teacher controls the current token and all remaining tokens.

In contrast, define the teacher continuation value after a student action as

$$C_t^{T|S}(s_t) = \mathbb{E}_{a \sim p_t^{\mathrm{dec}}(\cdot|s_t)} \left[Q_t^T(s_t, a) \right]. \tag{46}$$

This quantity measures the expected final reward when the student selects the current token but the teacher generates the remainder of the trajectory.

The distinction between $V_t^T(s_t)$ and $C_t^{T|S}(s_t)$ captures the recovery problem. If the teacher can successfully continue after student selected tokens, then $C_t^{T|S}(s_t)$ remains close to $V_t^T(s_t)$. If student actions lead to states from which the teacher cannot recover, then $C_t^{T|S}(s_t)$ can be substantially smaller.

The connection between Recipe II and $C_t^{T|S}$ can now be made explicit. Once the student selects a_t, it produces the next state

$$s_{t+1} = \big(x, y_{\le t}\big). \tag{47}$$

By the definition of the teacher action value,

$$Q_t^T(s_t, a_t) = \mathbb{E}\left[V_{t+1}^T(s_{t+1}) \mid s_t, a_t \right]. \tag{48}$$

Taking expectations over the student action gives

$$C_t^{T|S}(s_t) = \mathbb{E}_{a_t \sim p_t^{\mathrm{dec}}(\cdot|s_t)} \left[V_{t+1}^T(s_{t+1}) \right]. \tag{49}$$

Averaging over states visited by the student therefore yields

$$\mathbb{E}_{s_t \sim d_S^t} \left[C_t^{T|S}(s_t) \right] = \mathbb{E}_{s_{t+1} \sim d_S^{t+1}} \left[V_{t+1}^T(s_{t+1}) \right]. \tag{50}$$

Equation Eq. (50) establishes the direct connection between Recipe II and the student branch continuation values. Sampling a student prefix produces a state from d_S^{t+1}. Training the teacher on continuations from such prefixes using sequence level task reward directly targets the right hand side of Equation Eq. (50), which is equal to the expected continuation value of student selected branches on the left hand side.

For a uniform distribution over nonempty student prefix lengths, the population recovery objective can be written as

$$\begin{aligned} J_{\mathrm{rec}}(\pi_T; \pi_S) &= \frac{1}{H-1} \sum_{t=2}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[V_t^T(s_t) \right] \\ &= \frac{1}{H-1} \sum_{t=1}^{H-1} \mathbb{E}_{s_t \sim d_S^t} \left[C_t^{T|S}(s_t) \right]. \end{aligned} \tag{51}$$

If different prefix lengths are sampled with different probabilities, the same identity holds with the corresponding weights.

Recipe II is therefore not merely ordinary teacher training with different inputs. Its population objective is aligned with increasing the teacher values associated with student induced branches, which determine whether local student deviations propagate into sequence level failure. This alignment of objectives does not imply that every practical optimization run must successfully increase those values.

E.3.2 A Sufficient Condition for a Nonincreasing Transfer Term

Recipe II directly targets the continuation value $C_t^{T|S}$. Increasing this value alone, however, does not guarantee that the transfer term becomes smaller, because recovery training may also change the teacher value V_t^T. We therefore derive a sufficient condition under which the local transfer gap does not increase.

Let $\pi_T^{(0)}$ denote the original teacher and let $\pi_T^{(R)}$ denote the recovery trained teacher. To isolate the effect of Recipe II, consider a fixed student policy and its induced state distributions. Define

$$\Delta V_t(s) = V_t^{T^{(R)}}(s) - V_t^{T^{(0)}}(s), \tag{52}$$

and

$$\Delta C_t(s) = C_t^{T^{(R)}|S}(s) - C_t^{T^{(0)}|S}(s). \tag{53}$$

Recall that the local transfer gap is

$$g_t^{T|S}(s) = \left[V_t^T(s) - C_t^{T|S}(s)\right]_+. \tag{54}$$

The following proposition states when recovery training prevents this gap from increasing.

Proposition E.4 (Sufficient Condition for Recovery Improvement)**.** *Suppose that, for every state visited by the student,*

$$\Delta C_t(s) \geq \Delta V_t(s). \tag{55}$$

Then the local transfer gap does not increase:

$$g_t^{T^{(R)}|S}(s) \leq g_t^{T^{(0)}|S}(s). \tag{56}$$

Consequently,

$$H \cdot \overline{\kappa}_{T^{(R)}|S} \cdot \epsilon^{\mathrm{dec}}_{S,T^{(R)}} \leq H \cdot \overline{\kappa}_{T^{(0)}|S} \cdot \epsilon^{\mathrm{dec}}_{S,T^{(0)}}. \tag{57}$$

Proof. Using Equations Eq. (52) and Eq. (53),

$$\begin{aligned} V_t^{T^{(R)}}(s) - C_t^{T^{(R)}|S}(s) = V_t^{T^{(0)}}(s) - C_t^{T^{(0)}|S}(s) \\ + \Delta V_t(s) - \Delta C_t(s). \end{aligned} \tag{58}$$

Condition Eq. (55) implies

$$\Delta V_t(s) - \Delta C_t(s) \leq 0. \tag{59}$$

Therefore,

$$V_t^{T^{(R)}}(s) - C_t^{T^{(R)}|S}(s) \leq V_t^{T^{(0)}}(s) - C_t^{T^{(0)}|S}(s). \tag{60}$$

Since the positive part function is monotone,

$$\left[V_t^{T^{(R)}}(s) - C_t^{T^{(R)}|S}(s)\right]_+ \leq \left[V_t^{T^{(0)}}(s) - C_t^{T^{(0)}|S}(s)\right]_+. \tag{61}$$

This proves Equation Eq. (56).

Taking expectations over student states and summing over token positions gives

$$\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[g_t^{T^{(R)}|S}(s_t)\right] \leq \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[g_t^{T^{(0)}|S}(s_t)\right]. \tag{62}$$

Section E.5 establishes that

$$\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[g_t^{T|S}(s_t)\right] = H \cdot \overline{\kappa}_{T|S} \cdot \epsilon^{\mathrm{dec}}_{S,T}. \tag{63}$$

Substituting this identity into Equation Eq. (62) proves Equation Eq. (57). □

Condition Eq. (55) has a direct operational interpretation. Recovery training should improve the value obtained after a student selected action at least as much as it improves the value obtained when the teacher controls the current action. Recipe II is designed to encourage this preferential improvement because its training states are prefixes generated by the student rather than only trajectories generated by the teacher.

A transparent special case occurs when recovery training approximately preserves the teacher controlled value, so that $\Delta V_t(s) \approx 0$. In this case, any nonnegative improvement $\Delta C_t(s) \geq 0$ prevents the local transfer gap from increasing, up to the approximation error. The gap decreases strictly at states where $\Delta C_t(s) > \Delta V_t(s)$ and the original local transfer gap is positive.

The proposition does not claim that every recovery training run must satisfy Equation Eq. (55). Nonconvex policy optimization may fail, and recovery training may change the teacher policy in unintended ways. Instead, the proposition identifies a sufficient behavioral condition under which Recipe II produces a no larger transfer component.

The result in Equation Eq. (57) concerns the transfer term in Proposition E.1. If the decoded policy mismatch remains unchanged,

$$\epsilon^{\mathrm{dec}}_{S,T^{(R)}} = \epsilon^{\mathrm{dec}}_{S,T^{(0)}} > 0, \tag{64}$$

then Equation Eq. (57) directly implies

$$\overline{\kappa}_{T^{(R)}|S} \leq \overline{\kappa}_{T^{(0)}|S}. \tag{65}$$

If recovery training also changes the decoded policy mismatch, the more precise conclusion is that the product

$$\overline{\kappa}_{T|S} \cdot \epsilon^{\mathrm{dec}}_{S,T} \tag{66}$$

does not increase. This product is the quantity that enters the unified student bound, even when the two factors cannot be isolated individually.

Proposition E.4 controls the transfer component rather than the complete student bound. For the fixed student policy considered above, the complete upper bound is no larger whenever

$$\Delta^{(R)}_T - \Delta^{(0)}_T \leq H \cdot \left[\overline{\kappa}_{T^{(0)}|S} \cdot \epsilon^{\mathrm{dec}}_{S,T^{(0)}} - \overline{\kappa}_{T^{(R)}|S} \cdot \epsilon^{\mathrm{dec}}_{S,T^{(R)}} \right]. \tag{67}$$

A simple sufficient case is that recovery training does not increase the teacher suboptimality, $\Delta^{(R)}_T \leq \Delta^{(0)}_T$, while satisfying Proposition E.4. If the student is retrained and its induced state distribution changes, the quantities in the bound must be evaluated under the corresponding new student policy. The fixed student analysis above isolates the teacher side mechanism targeted by Recipe II.

E.3.3 Recovery Profiles as Empirical Diagnostics

The continuation value analysis yields a direct empirical diagnostic. A teacher with stronger recovery ability should retain higher expected task performance when generation begins from student generated prefixes.

For a student prefix of length ℓ, define the population recovery profile as

$$\mathcal{R}_T(\ell) = \mathbb{E}_{s_{\ell+1} \sim d_S^{\ell+1}} \left[V^T_{\ell+1}(s_{\ell+1}) \right], \qquad 1 \leq \ell \leq H-1. \tag{68}$$

Using Equation Eq. (50),

$$\mathcal{R}_T(\ell) = \mathbb{E}_{s_\ell \sim d_S^\ell} \left[C^{T|S}_\ell(s_\ell) \right]. \tag{69}$$

At the population level, the recovery profile equals the expected continuation value of student selected branches. The finite sample recoverability curve measured in experiments therefore provides a direct estimator of this population quantity rather than an exact measurement of it.

A rapidly decreasing $\mathcal{R}_T(\ell)$ indicates that the teacher becomes increasingly unreliable as the student prefix grows. This pattern is consistent with error accumulation: longer prefixes expose the teacher to more student generated decisions and potentially more deviations from trajectories the teacher would generate independently.

When recovery optimization successfully improves the targeted continuation values, it should raise $\mathcal{R}_T(\ell)$. Improvements at larger ℓ are particularly informative because these states contain longer student generated contexts and empirically tend to present greater recovery difficulty. Such improvements may appear as a flatter recovery profile, although this shape is an empirical signature rather than a universal theoretical guarantee.

An improvement in the recovery profile alone does not unconditionally prove that $\overline{\kappa}_{T|S}$ decreases, because the teacher controlled value V_t^T may change at the same time. However, when the continuation values of student induced branches improve according to Equation Eq. (55), the transfer term does not increase.

E.3.4 A Sufficient Student State Coverage Condition

The preceding analysis is expressed under the student state distribution appearing in Proposition E.1. In practice, Recipe II trains on prefixes collected from one or more student policies. Its benefit transfers most directly when the recovery training distribution covers the states encountered during subsequent OPD.

Let μ_t denote the state distribution used for recovery training at step t, and let d_S^t denote the student state distribution relevant to the later OPD process. Suppose there exists a finite coverage coefficient C_{cov} such that

$$d_S^t(s) \le C_{\text{cov}} \mu_t(s) \tag{70}$$

for every relevant state s.

For any nonnegative state dependent quantity $f_t(s)$,

$$\begin{aligned} \mathbb{E}_{s\sim d_S^t}\left[f_t(s)\right] &= \sum_s d_S^t(s) f_t(s) \\ &\le C_{\text{cov}} \sum_s \mu_t(s) f_t(s) \\ &= C_{\text{cov}} \mathbb{E}_{s\sim\mu_t}\left[f_t(s)\right]. \end{aligned} \tag{71}$$

For example, let

$$f_t(s) = 1 - V_t^T(s) \tag{72}$$

denote the teacher recovery failure at state s. Then Equation Eq. (71) shows that low expected recovery failure under the training state distribution controls the corresponding failure under the later student distribution up to the coverage coefficient.

This result clarifies why the source of student prefixes matters. Recovery training on states that differ substantially from those encountered during OPD may provide limited benefit, even if the teacher performs well on the collected recovery samples. Collecting prefixes from representative student policies, or refreshing the prefix distribution as the student evolves, can improve this coverage.

The coverage condition is sufficient rather than necessary. It provides a transparent way to state a distributional condition under which recovery improvements transfer, but Recipe II may still be beneficial when no uniform finite C_{cov} exists.

E.3.5 Summary

The theoretical role of Recipe II can be summarized as follows. Outcome based training on student generated prefixes directly targets an increase in the expected continuation values of student selected branches:

$$\boxed{\text{Recovery Training on Student Prefixes} \quad \text{targets} \quad C_t^{T|S}(s) \uparrow .} \tag{73}$$

When these continuation values improve at least as much as the values obtained when the teacher controls the current action,

$$\Delta C_t(s) \ge \Delta V_t(s), \tag{74}$$

the local transfer gaps do not increase, yielding

$$\boxed{H \cdot \overline{\kappa}_{T^{(R)}|S} \cdot \epsilon^{\mathrm{dec}}_{S,T^{(R)}} \le H \cdot \overline{\kappa}_{T^{(0)}|S} \cdot \epsilon^{\mathrm{dec}}_{S,T^{(0)}}.} \tag{75}$$

Under unchanged decoded alignment, this inequality corresponds directly to a nonincrease in $\overline{\kappa}_{T|S}$. More generally, it controls the complete transfer component in Proposition E.1. It yields a no larger complete student bound only when the accompanying change in teacher suboptimality satisfies Equation Eq. (67).

The empirical recovery profile estimates the population continuation values targeted by Recipe II, while the state coverage condition provides a sufficient criterion for transferring improvements learned from collected student prefixes to the states encountered during later OPD.

E.4 Theoretical Analysis of Recipe III: Controlling Decoded Policy Mismatch

This subsection formalizes how Recipe III targets the decoded policy mismatch $\epsilon^{\mathrm{dec}}_{S,T}$ in Proposition E.1. Its central principle is that the alignment objective should compare the distributions that actually govern token sampling, rather than only the raw next-token distributions produced by the models. In this way, Recipe III aligns the training objective with the behavioral discrepancy appearing in the unified performance decomposition.

Our analysis proceeds in five steps. First, we show that, under distribution-dependent hard truncation, an arbitrarily small divergence between raw distributions does not necessarily imply similar decoded behavior. Second, we prove that Decoding-Aware KL provides an explicit upper bound on the decoded policy mismatch. Third, we analyze support mismatch after truncation and quantify the approximation error introduced by dynamic support expansion. Fourth, we extend the analysis to the sampled-token form of Recipe III. Finally, we state the condition under which reducing decoded mismatch tightens the complete student performance bound.

E.4.1 Raw Alignment Does Not Generally Control Behavior Under Hard Truncation

Standard OPD commonly minimizes a divergence between the raw next-token distributions of the student and teacher. Under distribution-dependent hard truncation, however, a small raw-distribution divergence does not generally imply a small discrepancy between the resulting sampling distributions.

Let $p_t(\cdot \mid s)$ and $q_t(\cdot \mid s)$ denote the raw next-token distributions of the student and teacher at state s. Let $\mathcal{G}_S$ and $\mathcal{G}_T$ denote the respective decoding operators used by the student and teacher. These operators may include temperature scaling, top-k filtering, top-p filtering, and subsequent renormalization.

The corresponding sampling distributions are

$$p^{\mathrm{dec}}_t(\cdot \mid s) = \mathcal{G}_S\big(p_t(\cdot \mid s)\big), \qquad q^{\mathrm{dec}}_t(\cdot \mid s) = \mathcal{G}_T\big(q_t(\cdot \mid s)\big). \tag{76}$$

These decoded distributions govern the token-sampling probabilities during generation. The decoded policy mismatch appearing in Proposition E.1 is

$$\epsilon^{\mathrm{dec}}_{S,T} = \frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d^t_S}\left[D_{\mathrm{TV}}\left(p^{\mathrm{dec}}_t(\cdot \mid s_t), q^{\mathrm{dec}}_t(\cdot \mid s_t)\right)\right], \tag{77}$$

where d^t_S is the state distribution induced by the student and

$$D_{\mathrm{TV}}(P, Q) = \frac{1}{2}\sum_{v} |P(v) - Q(v)| \tag{78}$$

is the total variation distance.

The following proposition shows that raw KL does not provide a uniform control of decoded mismatch under top-1 truncation.

Proposition E.5 (Small Raw KL Does Not Imply Small Decoded Mismatch)**.** *For any $\varepsilon > 0$, there exist two raw distributions p and q such that*

$$\mathrm{KL}(p\|q) < \varepsilon, \tag{79}$$

while their top-1 decoded distributions satisfy

$$D_{\mathrm{TV}}\left(\mathcal{G}(p), \mathcal{G}(q)\right) = 1. \tag{80}$$

Proof. Consider a vocabulary with two tokens and define

$$p = \left(\frac{1}{2} + \eta, \frac{1}{2} - \eta\right), \qquad q = \left(\frac{1}{2} - \eta, \frac{1}{2} + \eta\right), \tag{81}$$

where $0 < \eta < 1/2$.

Their reverse KL divergence is

$$\begin{aligned}\mathrm{KL}(p\|q) &= \left(\frac{1}{2} + \eta\right) \log \frac{\frac{1}{2} + \eta}{\frac{1}{2} - \eta} \\ &\quad + \left(\frac{1}{2} - \eta\right) \log \frac{\frac{1}{2} - \eta}{\frac{1}{2} + \eta} \\ &= 2\eta \log \frac{\frac{1}{2} + \eta}{\frac{1}{2} - \eta}.\end{aligned} \tag{82}$$

As $\eta \to 0$,

$$\mathrm{KL}(p\|q) \to 0. \tag{83}$$

Therefore, for any $\varepsilon > 0$, one can choose a sufficiently small η such that $\mathrm{KL}(p\|q) < \varepsilon$.

Under top-1 decoding, however, the first token has the highest probability under p, while the second token has the highest probability under q. Hence,

$$\mathcal{G}(p) = (1, 0), \qquad \mathcal{G}(q) = (0, 1). \tag{84}$$

Their total variation distance is

$$D_{\mathrm{TV}}\left(\mathcal{G}(p), \mathcal{G}(q)\right) = 1. \tag{85}$$

□

Proposition E.5 shows that raw-distribution similarity does not guarantee similarity between the policies that generate tokens after hard truncation. Top-1 decoding is a special case of top-k decoding, so the example rules out a uniform raw-KL guarantee for hard top-k decoding in general. Analogous boundary effects may also occur under other distribution-dependent truncation rules, including top-p, although the proposition above is stated specifically for top-1.

The issue arises because the retained support depends on the input distribution. Small changes near a selection boundary can therefore produce discontinuous changes in the retained token set. This differs from applying a fixed stochastic transformation to both distributions. Consequently, under hard distribution-dependent truncation, minimizing raw KL does not generally upper-bound the decoded mismatch $\epsilon^{\mathrm{dec}}_{S,T}$.

E.4.2 Decoding-Aware KL Upper-Bounds the Decoded Policy Mismatch

Recipe III instead computes the divergence after applying the decoding transformations. This produces an objective that explicitly upper-bounds the decoded-policy mismatch appearing in the unified performance decomposition.

At the population level, define the normalized Decoding-Aware KL objective as

$$\mathcal{L}_{\mathrm{DAKL}} = \frac{1}{H} \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[\mathrm{KL}\left(p_t^{\mathrm{dec}}(\cdot \mid s_t) \,\middle\|\, q_t^{\mathrm{dec}}(\cdot \mid s_t)\right)\right]. \tag{86}$$

For the moment, assume that the decoded teacher distribution assigns positive probability to every token in the support of the decoded student distribution, so that the KL divergence is finite. Support mismatch is analyzed in the next subsection.

The following proposition establishes the direct connection between the DAKL objective and the alignment quantity in the unified decomposition.

Proposition E.6 (Decoded Policy Mismatch Bound)**.** *The decoded policy mismatch is upper-bounded by*

$$\boxed{\epsilon_{S,T}^{\mathrm{dec}} \le \sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{DAKL}}}.} \tag{87}$$

Proof. Pinsker's inequality gives, at every student-visited state,

$$D_{\mathrm{TV}}\left(p_t^{\mathrm{dec}}, q_t^{\mathrm{dec}}\right) \le \sqrt{\frac{1}{2}\mathrm{KL}\left(p_t^{\mathrm{dec}} \| q_t^{\mathrm{dec}}\right)}. \tag{88}$$

Substituting Equation Eq. (88) into Equation Eq. (77) yields

$$\epsilon_{S,T}^{\mathrm{dec}} \le \frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[\sqrt{\frac{1}{2}\mathrm{KL}\left(p_t^{\mathrm{dec}} \| q_t^{\mathrm{dec}}\right)}\right]. \tag{89}$$

Since the square-root function is concave, Jensen's inequality gives

$$\begin{aligned}\epsilon_{S,T}^{\mathrm{dec}} &\le \sqrt{\frac{1}{2H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[\mathrm{KL}\left(p_t^{\mathrm{dec}} \| q_t^{\mathrm{dec}}\right)\right]} \\ &= \sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{DAKL}}}.\end{aligned} \tag{90}$$

□

Proposition E.6 provides the direct theoretical connection for Recipe III. A smaller population Decoding-Aware KL produces a smaller explicit upper bound on the decoded-policy mismatch $\epsilon_{S,T}^{\mathrm{dec}}$. This does not by itself imply that every reduction in the upper bound must produce a strict reduction in the realized mismatch.

Combining Proposition E.6 with the unified decomposition gives

$$\Delta_S \le \Delta_T + H \cdot \overline{\kappa}_{T|S} \cdot \sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{DAKL}}}. \tag{91}$$

Since Section E.5 establishes $\overline{\kappa}_{T|S} \le 1$, a looser consequence is

$$\Delta_S \le \Delta_T + H \cdot \sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{DAKL}}}. \tag{92}$$

These inequalities are population statements. They show that the DAKL objective targets an explicit upper bound on the decoded-policy mismatch used in our performance decomposition. They do not imply that every finite-sample optimization run must reduce the population objective, the realized mismatch, or the final student error.

E.4.3 Handling Support Mismatch Through Dynamic Support Expansion

Decoding transformations such as top-k and top-p filtering commonly produce sparse distributions with different, and potentially only partially overlapping, supports. If the student assigns positive probability to a token that receives zero probability under the decoded teacher distribution, the reverse KL in Equation Eq. (86) becomes infinite.

Recipe III addresses this support mismatch by dynamically expanding the teacher support before renormalization. We first analyze the form used when aligning the full retained student support.

At a state s, define the supports as

$$\mathcal{S}_S(s) = \operatorname{supp}\left(p_t^{\mathrm{dec}}(\cdot \mid s)\right), \qquad \mathcal{S}_T(s) = \operatorname{supp}\left(q_t^{\mathrm{dec}}(\cdot \mid s)\right). \tag{93}$$

The expanded teacher support is the union

$$\overline{\mathcal{S}}_T(s) = \mathcal{S}_T(s) \cup \mathcal{S}_S(s). \tag{94}$$

Let $r_t^T(\cdot \mid s)$ denote the teacher distribution after continuous transformations, such as temperature scaling, but before hard truncation. Define the teacher probability mass retained by its original support as

$$a_t(s) = \sum_{v \in \mathcal{S}_T(s)} r_t^T(v \mid s), \tag{95}$$

and the additional teacher probability mass introduced by the student support as

$$b_t(s) = \sum_{v \in \mathcal{S}_S(s) \setminus \mathcal{S}_T(s)} r_t^T(v \mid s). \tag{96}$$

The original decoded teacher distribution is

$$q_t^{\mathrm{dec}}(v \mid s) = \frac{r_t^T(v \mid s)\mathbb{I}\left[v \in \mathcal{S}_T(s)\right]}{a_t(s)}. \tag{97}$$

The expanded teacher distribution is

$$\overline{q}_t(v \mid s) = \frac{r_t^T(v \mid s)\mathbb{I}\left[v \in \overline{\mathcal{S}}_T(s)\right]}{a_t(s) + b_t(s)}. \tag{98}$$

The expansion gives positive teacher probability to every student-retained token whose pretruncation teacher probability is positive. Under this condition, the reverse KL with respect to $\overline{q}_t$ is finite.

When the student support contains tokens outside the original teacher support, the expansion changes the normalized teacher distribution. The following lemma quantifies this change exactly.

Lemma E.7 (Support Expansion Distortion)**.** *The total variation distance between the original decoded teacher distribution and the expanded teacher distribution is*

$$\boxed{D_{\mathrm{TV}}\left(\overline{q}_t, q_t^{\mathrm{dec}}\right) = \frac{b_t}{a_t + b_t}.} \tag{99}$$

Proof. For readability, we omit the state argument. By the definition of total variation,

$$\begin{aligned} 2D_{\mathrm{TV}}\left(\overline{q}_t, q_t^{\mathrm{dec}}\right) &= \sum_{v \in \mathcal{S}_T} r_t^T(v) \left| \frac{1}{a_t + b_t} - \frac{1}{a_t} \right| \\ &\quad + \sum_{v \in \mathcal{S}_S \setminus \mathcal{S}_T} \frac{r_t^T(v)}{a_t + b_t}. \end{aligned} \tag{100}$$

The first term equals

$$\begin{aligned} \sum_{v \in \mathcal{S}_T} r_t^T(v) \left(\frac{1}{a_t} - \frac{1}{a_t + b_t} \right) &= a_t \left(\frac{1}{a_t} - \frac{1}{a_t + b_t} \right) \\ &= \frac{b_t}{a_t + b_t}. \end{aligned} \tag{101}$$

The second term is

$$\frac{b_t}{a_t + b_t}. \tag{102}$$

Therefore,

$$2D_{\mathrm{TV}}\left(\overline{q}_t, q_t^{\mathrm{dec}}\right) = \frac{2b_t}{a_t + b_t}, \tag{103}$$

which proves Equation Eq. (99). □

Lemma E.7 shows that the distortion is exactly determined by the teacher probability mass added to the expanded support. When b_t is small relative to a_t, the expanded teacher distribution remains close to the original decoded teacher distribution. When $b_t = 0$, no support expansion occurs and the two distributions are identical.

Define the support-expanded Decoding-Aware KL objective as

$$\mathcal{L}_{\mathrm{DAKL}}^{\mathrm{exp}} = \frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[\mathrm{KL}\left(p_t^{\mathrm{dec}}(\cdot \mid s_t) \,\middle\|\, \overline{q}_t(\cdot \mid s_t)\right)\right], \tag{104}$$

and define the average support distortion as

$$\delta_{\mathrm{sup}} = \frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[\frac{b_t(s_t)}{a_t(s_t) + b_t(s_t)}\right]. \tag{105}$$

The following proposition bounds the original decoded mismatch using the support-expanded objective.

Proposition E.8 (Decoded Mismatch Under Support Expansion)**.** *The decoded policy mismatch satisfies*

$$\boxed{\epsilon_{S,T}^{\mathrm{dec}} \le \sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{DAKL}}^{\mathrm{exp}}} + \delta_{\mathrm{sup}}.} \tag{106}$$

Proof. At each student-visited state, the triangle inequality for total variation gives

$$D_{\mathrm{TV}}\left(p_t^{\mathrm{dec}}, q_t^{\mathrm{dec}}\right) \le D_{\mathrm{TV}}\left(p_t^{\mathrm{dec}}, \overline{q}_t\right) + D_{\mathrm{TV}}\left(\overline{q}_t, q_t^{\mathrm{dec}}\right). \tag{107}$$

Pinsker's inequality and Lemma E.7 imply

$$D_{\mathrm{TV}}\left(p_t^{\mathrm{dec}}, q_t^{\mathrm{dec}}\right) \le \sqrt{\frac{1}{2}\mathrm{KL}\left(p_t^{\mathrm{dec}} \| \overline{q}_t\right)} + \frac{b_t}{a_t + b_t}. \tag{108}$$

Taking expectations over student states, averaging over positions, and applying Jensen's inequality to the square-root term yields Equation Eq. (106). □

Combining Proposition E.8 with the unified decomposition yields

$$\Delta_S \le \Delta_T + H \cdot \overline{\kappa}_{T|S} \cdot \left[\sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{DAKL}}^{\mathrm{exp}}} + \delta_{\mathrm{sup}}\right]. \tag{109}$$

Equation Eq. (109) separates two effects. The first term inside the brackets is controlled by the support-expanded DAKL objective. The second term quantifies the discrepancy between the expanded teacher target and the teacher distribution that would otherwise govern decoding.

The support distortion is small when the added student tokens collectively receive little teacher probability mass before truncation. If an inserted token has exactly zero pretruncation teacher probability, support expansion alone cannot make the KL finite, and an additional smoothing mechanism is required. Under the idealized real-valued softmax model without structural masking, all pretruncation probabilities are positive. Finite-precision implementations may nevertheless require numerical smoothing.

E.4.4 SAMPLED-TOKEN SUPPORT EXPANSION

Some scalable OPD implementations use sampled-token objectives rather than full-support alignment. In this form of Recipe III, the teacher support is expanded only with the token sampled by the student. Because the expanded teacher distribution depends on the sampled token, the resulting expected log ratio is not a KL divergence between two fixed distributions and must be analyzed separately.

For a sampled token $y \in \mathcal{S}_S(s)$, define the single-token expanded support as

$$\mathcal{S}_T^{(y)}(s) = \mathcal{S}_T(s) \cup \{y\}. \tag{110}$$

The corresponding teacher distribution is

$$q_t^{(y)}(v \mid s) = \frac{r_t^T(v \mid s)\mathbb{I}\left[v \in \mathcal{S}_T^{(y)}(s)\right]}{a_t(s) + r_t^T(y \mid s)\mathbb{I}\left[y \notin \mathcal{S}_T(s)\right]}. \tag{111}$$

Define the population sampled-token objective at state s as

$$\ell_{\mathrm{ST}}(s) = \mathbb{E}_{y \sim p_t^{\mathrm{dec}}(\cdot \mid s)}\left[\log p_t^{\mathrm{dec}}(y \mid s) - \log q_t^{(y)}(y \mid s)\right]. \tag{112}$$

Because the teacher target $q_t^{(y)}$ depends on the sampled token, $\ell_{\mathrm{ST}}(s)$ is a sampled log-ratio objective rather than a KL divergence and need not be nonnegative.

Let $\overline{q}_t$ be the full union-support distribution in Equation Eq. (98). The following lemma connects the sampled-token objective to the KL against this fixed expanded distribution.

Lemma E.9 (Sampled-Token Objective and Union-Support KL)**.** *At every state,*

$$\mathrm{KL}\left(p_t^{\mathrm{dec}} \| \overline{q}_t\right) \le \ell_{\mathrm{ST}}(s) + \log \frac{a_t(s) + b_t(s)}{a_t(s)}. \tag{113}$$

Proof. For every $y \in \mathcal{S}_S(s)$, decompose the log ratio as

$$\log \frac{p_t^{\mathrm{dec}}(y \mid s)}{\overline{q}_t(y \mid s)} = \log \frac{p_t^{\mathrm{dec}}(y \mid s)}{q_t^{(y)}(y \mid s)} + \log \frac{q_t^{(y)}(y \mid s)}{\overline{q}_t(y \mid s)}. \tag{114}$$

If $y \in \mathcal{S}_T(s)$, then

$$\frac{q_t^{(y)}(y \mid s)}{\overline{q}_t(y \mid s)} = \frac{a_t(s) + b_t(s)}{a_t(s)}. \tag{115}$$

If $y \notin \mathcal{S}_T(s)$, then

$$\frac{q_t^{(y)}(y \mid s)}{\overline{q}_t(y \mid s)} = \frac{a_t(s) + b_t(s)}{a_t(s) + r_t^T(y \mid s)} \le \frac{a_t(s) + b_t(s)}{a_t(s)}. \tag{116}$$

Therefore, in both cases,

$$\log \frac{q_t^{(y)}(y \mid s)}{\overline{q}_t(y \mid s)} \le \log \frac{a_t(s) + b_t(s)}{a_t(s)}. \tag{117}$$

Taking the expectation of Equation Eq. (114) over $y \sim p_t^{\mathrm{dec}}(\cdot \mid s)$ proves the result. □

Define the normalized sampled-token objective as

$$\mathcal{L}_{\mathrm{DAST}} = \frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[\ell_{\mathrm{ST}}(s_t)\right], \tag{118}$$

and define the support normalization term as

$$\xi_{\mathrm{sup}} = \frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[\log \frac{a_t(s_t) + b_t(s_t)}{a_t(s_t)}\right]. \tag{119}$$

Although $\mathcal{L}_{\mathrm{DAST}}$ need not be nonnegative by itself, Lemma E.9 implies that $\mathcal{L}_{\mathrm{DAST}} + \xi_{\mathrm{sup}}$ upper-bounds a nonnegative average KL divergence.

Proposition E.10 (Sampled-Token Decoded Mismatch Bound)**.** *The decoded policy mismatch satisfies*

$$\boxed{\epsilon^{\mathrm{dec}}_{S,T} \le \sqrt{\frac{1}{2}\left(\mathcal{L}_{\mathrm{DAST}} + \xi_{\mathrm{sup}}\right)} + \delta_{\mathrm{sup}}.} \tag{120}$$

Proof. Averaging Equation Eq. (113) over student-visited states and token positions gives

$$\frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[\mathrm{KL}\left(p_t^{\mathrm{dec}} \| \overline{q}_t\right)\right] \le \mathcal{L}_{\mathrm{DAST}} + \xi_{\mathrm{sup}}. \tag{121}$$

At each state, the triangle inequality gives

$$D_{\mathrm{TV}}\left(p_t^{\mathrm{dec}}, q_t^{\mathrm{dec}}\right) \le D_{\mathrm{TV}}\left(p_t^{\mathrm{dec}}, \overline{q}_t\right) + D_{\mathrm{TV}}\left(\overline{q}_t, q_t^{\mathrm{dec}}\right). \tag{122}$$

Applying Pinsker's inequality to the first term, Lemma E.7 to the second term, and Jensen's inequality after averaging yields

$$\begin{aligned}\epsilon^{\mathrm{dec}}_{S,T} &\le \sqrt{\frac{1}{2H}\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[\mathrm{KL}\left(p_t^{\mathrm{dec}} \| \overline{q}_t\right)\right]} + \delta_{\mathrm{sup}} \\ &\le \sqrt{\frac{1}{2}\left(\mathcal{L}_{\mathrm{DAST}} + \xi_{\mathrm{sup}}\right)} + \delta_{\mathrm{sup}}.\end{aligned} \tag{123}$$

□

The additional term ξ_{sup} appears because single-token expansion produces a different normalized teacher distribution for each sampled token. When the full student support is used for expansion, as in Equation Eq. (104), the teacher target is fixed at each state and the tighter bound in Equation Eq. (106) applies directly.

For the sampled-token form, reducing $\mathcal{L}_{\mathrm{DAST}}$ alone is not sufficient to guarantee a tighter mismatch bound if ξ_{sup} or δ_{sup} increases. The bound tightens when the complete right-hand side of Equation Eq. (120) decreases.

E.4.5 When Does Recipe III Tighten the Student Guarantee?

Recipe III leaves the teacher policy unchanged, so the teacher suboptimality Δ_T is unchanged. Its primary role is to control the decoded policy mismatch, or an explicit upper bound on that mismatch. Changing the student objective may nevertheless change both the student policy and the state distribution on which recovery sensitivity is evaluated.

Let $\pi_S^{(0)}$ denote a student trained with the standard objective, and let $\pi_S^{(D)}$ denote a student trained with Recipe III. Since both use the same teacher, their theoretical upper bounds are

$$\mathcal{B}_0 = \Delta_T + H \cdot \overline{\kappa}_{T|S^{(0)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(0)},T}, \tag{124}$$

and

$$\mathcal{B}_D = \Delta_T + H \cdot \overline{\kappa}_{T|S^{(D)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(D)},T}. \tag{125}$$

Therefore, Recipe III produces a tighter complete upper bound whenever

$$\boxed{\overline{\kappa}_{T|S^{(D)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(D)},T} < \overline{\kappa}_{T|S^{(0)}} \cdot \epsilon^{\mathrm{dec}}_{S^{(0)},T}.} \tag{126}$$

A simple sufficient case occurs when recovery sensitivity does not increase and Recipe III reduces the decoded mismatch:

$$\overline{\kappa}_{T|S^{(D)}} \le \overline{\kappa}_{T|S^{(0)}}, \qquad \epsilon^{\mathrm{dec}}_{S^{(D)},T} < \epsilon^{\mathrm{dec}}_{S^{(0)},T}. \tag{127}$$

More generally, the reduction in decoded mismatch must be large enough to offset any increase in recovery sensitivity associated with the changed student policy and its induced state distribution.

The theoretical advantage of Recipe III is not that every optimization run must satisfy Equation Eq. (126). Rather, Recipe III replaces an objective that does not generally control decoded mismatch under hard truncation with one that explicitly upper-bounds the alignment quantity appearing in the unified performance decomposition.

E.4.6 SUMMARY

The theoretical role of Recipe III can be summarized by three main statements.

First, under hard distribution-dependent truncation, raw-distribution alignment does not generally control decoded behavior:

$$\boxed{\mathrm{KL}(p_t \| q_t) \to 0 \quad \not\Rightarrow \quad D_{\mathrm{TV}}\left(p_t^{\mathrm{dec}}, q_t^{\mathrm{dec}}\right) \to 0.} \tag{128}$$

Second, Decoding-Aware KL upper-bounds the decoded policy mismatch:

$$\boxed{\epsilon_{S,T}^{\mathrm{dec}} \le \sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{DAKL}}}.} \tag{129}$$

Third, dynamic support expansion makes the reverse KL finite under the stated positivity condition while introducing an explicit approximation term:

$$\boxed{\epsilon_{S,T}^{\mathrm{dec}} \le \sqrt{\frac{1}{2}\mathcal{L}_{\mathrm{DAKL}}^{\mathrm{exp}}} + \delta_{\mathrm{sup}}.} \tag{130}$$

The sampled-token form admits a related bound with the additional normalization term ξ_{sup}. Thus, Recipe III replaces raw-distribution alignment, which does not generally control decoded behavior under hard truncation, with an objective that explicitly upper-bounds the decoded-policy mismatch appearing in Proposition E.1. When the resulting decrease in decoded mismatch is large enough to offset any increase in recovery sensitivity, the corresponding upper bound on student suboptimality becomes tighter.

E.5 PROOF OF THE UNIFIED OPD PERFORMANCE DECOMPOSITION

This subsection provides the complete mathematical proof of Proposition E.1. The central conclusion is that the final student suboptimality can be separated into the intrinsic suboptimality of the teacher and a transfer term that depends on two factors: the mismatch between the actual student and teacher sampling policies, and the sequence-level consequence of that local mismatch.

Proof strategy. The derivation proceeds in three steps. First, we express the global performance difference between the teacher and student as a telescoping sum of local continuation losses over states visited by the student. Second, we normalize each positive local continuation loss by the magnitude of the decoded policy mismatch, yielding a bounded effective recovery sensitivity. Third, we combine the resulting transfer term with the intrinsic teacher suboptimality to obtain

$$\Delta_S \le \Delta_T + H \cdot \overline{\kappa}_{T|S} \cdot \epsilon_{S,T}^{\mathrm{dec}}. \tag{131}$$

E.5.1 STEP 1: LOCALIZING PERFORMANCE DIFFERENCE ON STUDENT-VISITED STATES

Our derivation begins by connecting global sequence-level outcomes to local token-level decisions. The first step is to rewrite the sequence-level performance difference $J(\pi_T) - J(\pi_S)$ as a sum of local differences evaluated on the state distribution induced by the student.

Consider a generation process with maximum response length H. A prompt $x \sim \mathcal{D}$ defines the initial state. At step t, the state is

$$s_t = \left(x, y_{<t}\right), \tag{132}$$

and the action $a_t = y_t$ is the next token. The next state is obtained by appending the selected token:

$$s_{t+1} = \big(x, y_{\le t}\big). \tag{133}$$

Variable-length responses can be included by treating the end-of-sequence token as an action and introducing an absorbing state after termination.

Let $p_t^{\mathrm{dec}}(\cdot \mid s_t)$ and $q_t^{\mathrm{dec}}(\cdot \mid s_t)$ denote the student and teacher distributions that actually govern token sampling. Let d_S^t denote the distribution over states at step t when the prompt is sampled from $\mathcal{D}$ and the preceding tokens are generated by the student.

Recall that the sequence-level reward satisfies $R(x, y) \in [0, 1]$. For a state s at step t and a candidate token a, define the teacher action value as

$$Q_t^T(s, a) = \mathbb{E}\left[R(x, y) \,|\, s_t = s,\ a_t = a,\ a_{t+1:H} \sim \pi_T\right]. \tag{134}$$

Thus, $Q_t^T(s, a)$ is the expected final reward when token a is selected at state s and the teacher generates the remaining trajectory.

The teacher value at state s is

$$V_t^T(s) = \mathbb{E}_{a \sim q_t^{\mathrm{dec}}(\cdot|s)}\left[Q_t^T(s, a)\right]. \tag{135}$$

This quantity is the expected final reward when the teacher controls both the current token and all remaining tokens.

We also define the teacher continuation value after a student action:

$$C_t^{T|S}(s) = \mathbb{E}_{a \sim p_t^{\mathrm{dec}}(\cdot|s)}\left[Q_t^T(s, a)\right]. \tag{136}$$

This is the expected final reward when the student selects the current token but the teacher controls the remaining trajectory.

Since $R(x, y) \in [0, 1]$, all associated continuation values satisfy

$$0 \le Q_t^T(s, a),\ V_t^T(s),\ C_t^{T|S}(s) \le 1. \tag{137}$$

The following lemma rewrites the teacher-student performance difference using these continuation values.

Lemma E.11 (Performance Difference on Student-Visited States)**.** *The teacher-student performance difference satisfies*

$$\boxed{J(\pi_T) - J(\pi_S) = \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t}\left[V_t^T(s_t) - C_t^{T|S}(s_t)\right].} \tag{138}$$

Proof. Define the terminal teacher value as

$$V_{H+1}^T(s_{H+1}) = R(x, y). \tag{139}$$

At the initial state, $s_1 = x$, the expected performance of the teacher is

$$J(\pi_T) = \mathbb{E}_{s_1}\left[V_1^T(s_1)\right]. \tag{140}$$

For a trajectory generated entirely by the student, the terminal teacher value is the realized sequence reward. Therefore,

$$J(\pi_S) = \mathbb{E}_{\tau \sim \pi_S}\left[V_{H+1}^T(s_{H+1})\right]. \tag{141}$$

Subtracting Equation Eq. (141) from Equation Eq. (140) gives

$$J(\pi_T) - J(\pi_S) = \mathbb{E}_{\tau \sim \pi_S}\left[V_1^T(s_1) - V_{H+1}^T(s_{H+1})\right]. \tag{142}$$

The difference inside the expectation can be expanded as a telescoping sum:

$$V_1^T(s_1) - V_{H+1}^T(s_{H+1}) = \sum_{t=1}^{H} \left[V_t^T(s_t) - V_{t+1}^T(s_{t+1}) \right]. \tag{143}$$

Substituting Equation Eq. (143) into Equation Eq. (142) yields

$$J(\pi_T) - J(\pi_S) = \sum_{t=1}^{H} \mathbb{E}_{\tau \sim \pi_S} \left[V_t^T(s_t) - V_{t+1}^T(s_{t+1}) \right]. \tag{144}$$

Conditioned on state s_t, the student samples

$$a_t \sim p_t^{\text{dec}}(\cdot \mid s_t). \tag{145}$$

By the definition of the teacher action value,

$$Q_t^T(s_t, a_t) = \mathbb{E}\left[V_{t+1}^T(s_{t+1}) \,\middle|\, s_t, a_t \right]. \tag{146}$$

Taking the expectation over the student action gives

$$\begin{aligned} \mathbb{E}\left[V_{t+1}^T(s_{t+1}) \mid s_t \right] &= \mathbb{E}_{a_t \sim p_t^{\text{dec}}(\cdot \mid s_t)} \left[Q_t^T(s_t, a_t) \right] \\ &= C_t^{T|S}(s_t). \end{aligned} \tag{147}$$

Applying the law of total expectation to Equation Eq. (144) and substituting Equation Eq. (147) gives

$$J(\pi_T) - J(\pi_S) = \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[V_t^T(s_t) - C_t^{T|S}(s_t) \right]. \tag{148}$$

□

Lemma E.11 shows that the sequence-level performance difference can be evaluated entirely on states induced by the student. At each such state, the local quantity $V_t^T(s_t) - C_t^{T|S}(s_t)$ measures the change in expected final reward caused by allowing the student, rather than the teacher, to select the current token.

E.5.2 Step 2: Factorizing the Local Transfer Error

Having localized the performance difference, the second step separates the magnitude of the decoded policy mismatch from the sequence-level consequence associated with that mismatch.

At state s, define the local decoded policy mismatch as

$$e_t^{S,T}(s) = D_{\text{TV}}\left(p_t^{\text{dec}}(\cdot \mid s), q_t^{\text{dec}}(\cdot \mid s) \right), \tag{149}$$

where

$$D_{\text{TV}}(P, Q) = \frac{1}{2} \sum_a |P(a) - Q(a)|. \tag{150}$$

Because $V_t^T(s) - C_t^{T|S}(s)$ may be negative when the student-induced action distribution yields a larger expected teacher continuation value than the teacher's own action distribution, we retain only its positive part:

$$g_t^{T|S}(s) = \left[V_t^T(s) - C_t^{T|S}(s) \right]_+, \tag{151}$$

where $[z]_+ = \max(z, 0)$.

When $e_t^{S,T}(s) > 0$, define the local recovery sensitivity as

$$\kappa_t^{T|S}(s) = \frac{g_t^{T|S}(s)}{e_t^{S,T}(s)}. \tag{152}$$

When $e_t^{S,T}(s) = 0$, we define

$$\kappa_t^{T|S}(s) = 0. \tag{153}$$

This definition is well posed. If $e_t^{S,T}(s) = 0$, then the student and teacher sampling distributions are identical at state s. Consequently,

$$C_t^{T|S}(s) = V_t^T(s), \tag{154}$$

and therefore $g_t^{T|S}(s) = 0$.

By definition, the positive local transfer gap can be written exactly as

$$g_t^{T|S}(s) = \kappa_t^{T|S}(s) \cdot e_t^{S,T}(s). \tag{155}$$

This factorization is definitional rather than an independence assumption. In particular, it does not assert that recovery sensitivity and decoded mismatch vary independently. Its utility is that the normalized consequence term $\kappa_t^{T|S}(s)$ has a direct continuation-value interpretation and, as shown below, is bounded by the span of the teacher action values.

The quantity $\kappa_t^{T|S}(s)$ is an effective recovery sensitivity under the current student policy. It measures the positive sequence-level loss associated with one unit of decoded policy mismatch at state s.

We next show that this sensitivity is bounded using a standard property of total variation distance.

Lemma E.12 (Expectation Difference Under Total Variation)**.** *Let P and Q be probability distributions over a discrete action space, and let f be a bounded real-valued function. Then*

$$|\mathbb{E}_{a\sim P}[f(a)] - \mathbb{E}_{a\sim Q}[f(a)]| \le \mathrm{span}(f) \cdot D_{\mathrm{TV}}(P, Q), \tag{156}$$

where

$$\mathrm{span}(f) = \max_a f(a) - \min_a f(a). \tag{157}$$

Proof. Let

$$M = \max_a f(a), \qquad m = \min_a f(a), \tag{158}$$

and define the midpoint

$$c = \frac{M + m}{2}. \tag{159}$$

Since

$$\sum_a [P(a) - Q(a)] = 0, \tag{160}$$

we have

$$\mathbb{E}_{a\sim P}[f(a)] - \mathbb{E}_{a\sim Q}[f(a)] = \sum_a [P(a) - Q(a)]\,[f(a) - c]\,. \tag{161}$$

Taking absolute values gives

$$|\mathbb{E}_P[f] - \mathbb{E}_Q[f]| \le \sum_a |P(a) - Q(a)|\,|f(a) - c|\,. \tag{162}$$

By the definition of c,

$$|f(a) - c| \le \frac{M - m}{2} = \frac{\mathrm{span}(f)}{2}. \tag{163}$$

Therefore,

$$\begin{aligned} |\mathbb{E}_P[f] - \mathbb{E}_Q[f]| &\le \frac{\mathrm{span}(f)}{2} \sum_a |P(a) - Q(a)| \\ &= \mathrm{span}(f) \cdot D_{\mathrm{TV}}(P, Q). \end{aligned} \tag{164}$$

□

Apply Lemma E.12 with

$$P = q_t^{\mathrm{dec}}(\cdot \mid s), \qquad Q = p_t^{\mathrm{dec}}(\cdot \mid s), \tag{165}$$

and

$$f(a) = Q_t^T(s, a). \tag{166}$$

Using the definitions of V_t^T and $C_t^{T|S}$, we obtain

$$\left| V_t^T(s) - C_t^{T|S}(s) \right| \le \operatorname{span}_a Q_t^T(s, a) \cdot e_t^{S,T}(s). \tag{167}$$

Because $Q_t^T(s, a) \in [0, 1]$, its span satisfies

$$\operatorname{span}_a Q_t^T(s, a) \le 1. \tag{168}$$

Combining Equations Eq. (151), Eq. (152), and Eq. (167) gives the following bounds on local recovery sensitivity:

$$0 \le \kappa_t^{T|S}(s) \le \operatorname{span}_a Q_t^T(s, a) \le 1. \tag{169}$$

Equation Eq. (169) provides a direct operational interpretation of recovery sensitivity. If the teacher achieves similar final rewards after different local actions, then the span of $Q_t^T(s, \cdot)$ is small, and local student-teacher mismatch has limited sequence-level consequences.

We now aggregate the local quantities over token positions and student-visited states. Define the average decoded policy mismatch as

$$\epsilon_{S,T}^{\mathrm{dec}} = \frac{1}{H} \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[e_t^{S,T}(s_t) \right]. \tag{170}$$

If $\epsilon_{S,T}^{\mathrm{dec}} > 0$, define the mismatch-weighted average recovery sensitivity as

$$\overline{\kappa}_{T|S} = \frac{\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[\kappa_t^{T|S}(s_t) \cdot e_t^{S,T}(s_t) \right]}{\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[e_t^{S,T}(s_t) \right]}. \tag{171}$$

If $\epsilon_{S,T}^{\mathrm{dec}} = 0$, we define

$$\overline{\kappa}_{T|S} = 0. \tag{172}$$

Since every local recovery sensitivity lies in $[0, 1]$, its mismatch-weighted average also satisfies

$$0 \le \overline{\kappa}_{T|S} \le 1. \tag{173}$$

By construction,

$$\begin{aligned} H \cdot \overline{\kappa}_{T|S} \cdot \epsilon_{S,T}^{\mathrm{dec}} &= \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[\kappa_t^{T|S}(s_t) \cdot e_t^{S,T}(s_t) \right] \\ &= \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[g_t^{T|S}(s_t) \right]. \end{aligned} \tag{174}$$

Thus, the recovery-related transfer term is exactly the accumulated positive continuation loss over states visited by the student.

E.5.3 Step 3: Synthesizing Teacher and Transfer Errors

With the positive local transfer gaps defined, we now combine them with the intrinsic teacher error.

Recall that

$$\Delta_S = J^\star - J(\pi_S), \qquad \Delta_T = J^\star - J(\pi_T). \tag{175}$$

Adding and subtracting $J(\pi_T)$ gives the exact identity

$$\begin{aligned}\Delta_S &= J^\star - J(\pi_S) \\ &= J^\star - J(\pi_T) + J(\pi_T) - J(\pi_S) \\ &= \Delta_T + J(\pi_T) - J(\pi_S).\end{aligned} \tag{176}$$

Lemma E.11 gives

$$J(\pi_T) - J(\pi_S) = \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[V_t^T(s_t) - C_t^{T|S}(s_t) \right]. \tag{177}$$

For any scalar z, we have $z \le [z]_+$. Therefore,

$$\begin{aligned}J(\pi_T) - J(\pi_S) &\le \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[\left[V_t^T(s_t) - C_t^{T|S}(s_t) \right]_+ \right] \\ &= \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_S^t} \left[g_t^{T|S}(s_t) \right] \\ &= H \cdot \overline{\kappa}_{T|S} \cdot \epsilon_{S,T}^{\mathrm{dec}},\end{aligned} \tag{178}$$

where the last equality follows from Equation Eq. (174).

Substituting Equation Eq. (178) into Equation Eq. (176) gives

$$\Delta_S \le \Delta_T + H \cdot \overline{\kappa}_{T|S} \cdot \epsilon_{S,T}^{\mathrm{dec}}. \tag{179}$$

This concludes the proof of Proposition E.1.

Relation to classical imitation learning formulations. Since $\overline{\kappa}_{T|S} \le 1$, the unified decomposition immediately implies the coarser bound

$$\Delta_S \le \Delta_T + H \cdot \epsilon_{S,T}^{\mathrm{dec}}. \tag{180}$$

This form is consistent with classical on-policy imitation learning analyses, where learner error scales linearly with the decision horizon and the policy discrepancy evaluated on learner-induced states (Ross et al., 2011). The refined bound in Proposition E.1 additionally isolates the effective consequence of local policy mismatch through $\overline{\kappa}_{T|S}$.

The performance difference identity used above is also structurally related to standard policy performance difference analyses in reinforcement learning (Schulman et al., 2015). The distinguishing feature of our formulation is that the local values are evaluated under teacher continuations while the state distribution is induced by the student, which makes the decomposition directly applicable to OPD.

E.5.4 Theoretical Scope and Limitations

The performance-difference identity in Lemma E.11 and the factorization $g_t^{T|S} = \kappa_t^{T|S} \cdot e_t^{S,T}$ are exact. Proposition E.1 becomes an upper bound only when each signed local continuation gap $V_t^T - C_t^{T|S}$ is replaced by its positive part in Equation Eq. (178).

One-sided transfer error. The positive-part construction retains harmful student deviations but discards beneficial deviations for which $C_t^{T|S}(s) > V_t^T(s)$. The resulting bound may therefore be conservative when the student-induced action distribution outperforms the teacher action distribution on a non-negligible set of states.

Assumptions of the formalism. Proposition E.1 compares a fixed student policy with a fixed teacher policy. During OPD, the student changes over time, so the bound can be applied to any training snapshot or to the final trained policy, provided that d_S^t, $\epsilon_{S,T}^{\mathrm{dec}}$, and $\overline{\kappa}_{T|S}$ are evaluated under the policy being analyzed.

The reference performance $J^\star$ is defined over a policy class containing both policies under comparison, ensuring that Δ_S and Δ_T are nonnegative. The derivation assumes a maximum response length H and a bounded sequence-level reward $R(x, y) \in [0, 1]$. If the reward lies in $[R_{\min}, R_{\max}]$, the same derivation applies after rescaling. Without rescaling, the upper bound on the recovery sensitivities becomes $R_{\max} - R_{\min}$.

The policies in the decomposition are the distributions that govern token sampling after the decoding transformations. Deterministic decoding is covered by representing the selected token as a degenerate distribution, assuming a fixed rule for resolving ties.

Interpretation of the transfer dynamics. The notation $\overline{\kappa}_{T|S}$ emphasizes that recovery sensitivity is not solely an intrinsic property of the teacher. It depends on the student actions, the states induced by the student, and the direction of the student-teacher mismatch. Consequently, a teacher may exhibit different recovery sensitivity for different students.

The decomposition identifies the primary factor targeted by each recipe, but these factors are coupled. Teacher adaptation may change both decoded mismatch and recovery sensitivity. Recovery training may also change the intrinsic teacher suboptimality or the teacher sampling policy. Decoding-Aware KL may alter the student states visited during generation.

Generalization and optimization conditions. The theoretical analyses in Sections E.2 through E.4 identify objectives and sufficient conditions associated with tighter bounds, but they do not guarantee that an arbitrary nonconvex optimization run must satisfy those conditions.

Recipe II additionally requires a coverage condition relating the state distribution used for recovery training to the student-state distribution encountered during subsequent OPD.

A tighter theoretical upper bound does not deterministically imply that the realized student performance must improve by the same amount. Rather, the decomposition identifies sufficient mechanisms through which the recipes can improve OPD and clarifies the theoretically relevant quantities that can guide empirical analysis.

Subject to these conditions, Proposition E.1 provides a unified theoretical account of the OPD Cookbook. The bound highlights three factors relevant to effective OPD: the intrinsic quality of the teacher, its continuation reliability on student-induced states, and the mismatch between the student and teacher policies that govern generation.