PACE: PREFIX-ANCHORED CALIBRATION WITH ADAPTIVE SEGMENTATION FOR ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) offers a practical way to align smaller student models with stronger teachers on student-generated trajectories. However, token-level teacher-student discrepancies are often noisy and poorly calibrated, as they are susceptible to capability-irrelevant noise, such as lexical variation, formatting artifacts and teacher preferences rather than reflecting differences in reasoning ability. Consequently, standard OPD struggles to distinguish meaningful teacher-student divergence from stable response-level mismatch. To address this problem, we propose PACE (Prefix-Anchored Calibration with adaptive sEgmentation), a segment-level calibration objective for on-policy distillation. PACE first adaptively partitions each student generated response into locally coherent segments using change points in teacher's negative log-likelihood (NLL), together with structural boundaries. PACE computes a student-to-teacher NLL ratio from segment-averaged losses, yielding a more stable local measure of mismatch than token-level discrepancies. The mean ratio over the initial segments serves as a trajectory-specific anchor against which subsequent segment ratios are calibrated. This relative formulation helps reduce the influence of persistent teacher–student discrepancies unrelated to reasoning ability, allowing the learning objective to focus more directly on segment-specific capability gaps. It complements standard OPD without requiring additional reward models or human annotations, providing a lightweight segment-level calibration signal for LLM agents. Experiments across tool-use and coding benchmarks demonstrate consistent improvements over OPD baselines across different teacher-student model scales, supporting the effectiveness and generality of PACE.

1 INTRODUCTION

Large language models (LLMs) have demonstrated increasingly strong capabilities in multi-step reasoning, code generation, and tool use. However, deploying these enormous-size models across long-horizon interactions incurs substantial computational and serving costs. Knowledge distillation offers a practical way to transfer their capabilities to smaller, more efficient models. Conventional offline distillation trains students on teacher-generated responses, which may introduce a mismatch between training and inference. On-policy distillation (OPD) mitigates this mismatch by providing teacher supervision on responses sampled from the student's current policy(Agarwal et al., 2024). The resulting teacher–student discrepancies serve as dense token-level learning signals, enabling policy optimization without task-specific outcome rewards or a separately trained reward model.

Despite providing dense supervision, OPD does not guarantee that token-level teacher–student discrepancies faithfully measure capability gaps (Nagarajan et al., 2023). Since the teacher evaluates tokens sampled from the student policy, token-level discrepancies are inherently noisy and can be confounded by capability-irrelevant factors, such as lexical choice, formatting conventions, and teacher-specific preferences. Importantly (Armandpour et al., 2026), these factors may induce a persistent teacher–student mismatch throughout the trajectory, yielding a trajectory-specific baseline discrepancy that is not necessarily tied to reasoning ability. Consequently, an absolute discrepancy observed in a later segment may reflect this intrinsic mismatch rather than a genuine increase in the student–teacher capability gap. This motivates calibrating local discrepancies against a response-specific reference, so that the resulting signal emphasizes relative deviations beyond the capability-irrelevant baseline.

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.

Existing methods address these challenges from two complementary directions. One line aggregates supervision over sequences, steps, or segments to reduce token-level fluctuations and improve credit assignment. However, sequence-level scores can obscure where deviations occur, while finer-grained methods often require predefined boundaries, task-specific structure, or additional supervision (Liu et al., 2025c). Another line modifies the distillation divergence or adapts teacher targets to the student's learning progress (Ko et al., 2024; Shing et al., 2025). These methods can stabilize training, but they do not provide a trajectory-specific reference for interpreting local disagreement. Without such a reference, it is difficult to tell whether a discrepancy signals a capability-relevant deviation or merely continues an existing mismatch caused by style, formatting, prompt difficulty, or teacher preference. This raises a central question: *How can OPD filter out noisy token-level discrepancies while identifying local deviations that are more likely to reflect a teacher–student ability gap?*

To bridge this gap, we introduce Prefix-Anchored Calibration with adaptive sEgmentation (PACE), an auxiliary OPD objective built around two designs. First, PACE dynamically partitions each response according to changes in teacher surprisal, forming variable-length segments for aggregating token-level likelihood information. Second, it uses the first few segments as an internal anchor, taking their average student-to-teacher NLL ratio as a detached baseline against which later segments are calibrated. Dynamic segmentation provides local measurement units, while the prefix anchor uses each response's initial teacher–student discrepancy as its reference level. PACE regularizes subsequent deviations from this reference mismatch alongside basic token-level OPD, without additional reward models or training annotations. Across six benchmarks covering diverse task types and teacher–student pairs of varying model sizes, it outperforms OPD in all evaluated settings. Our contributions are summarized as following:

- We introduce an adaptive segmentation mechanism that constructs local coherent units in a signal-driven rather than predefined manner. It partitions student responses using teacher-NLL change points and structural boundaries, then aggregates token-level discrepancies into segment-level student-to-teacher NLL ratios. This yields stable local mismatch estimates while avoiding fragmented token-level supervision and fixed-granularity boundaries.
- We propose **PACE**, a prefix-anchored segment-level calibration framework for OPD in LLM agents. PACE uses the ratio-based student–teacher discrepancy in the initial segments of each response as a trajectory-specific anchor, and calibrates later discrepancies against this anchor mismatch. This relative formulation preserves teacher-guided supervision while reducing capability-irrelevant noise through segment-level calibration, without requiring task-specific rewards, additional reward models, or human annotations.
- We evaluate PACE through experiments on six agentic benchmarks covering tool use and coding under different teacher–student model scales. PACE consistently improves over standard OPD across all evaluated settings. Diagnostic analyses further show that adaptive segmentation produces semantically coherent units and that prefix-anchored calibration provides a stable response-specific reference, supporting the effectiveness and generality of PACE for on-policy distillation.

2 Related Work

2.1 Granularity and Segmentation

Recent work has recognized that both token-level and sequence-level supervision are limited for long, structured LLM generations. Token-level objectives are sensitive to lexical, syntactic, and stylistic variation, while sequence-level objectives collapse an entire trajectory into a single scalar and obscure fine-grained credit assignment. Segment-level methods address this trade-off with different boundary criteria. Some use surface structures, such as newlines, learned segmenters, or tool-call boundaries (Ma et al., 2023; Setlur et al., 2025; Yin et al., 2025; Xie et al., 2026; Liu et al., 2025b), which may not align with changes in reasoning state. Others rely on student uncertainty, entropy transitions, or forking tokens (Liu et al., 2025c; Bin et al., 2025; Wang et al., 2026; Hou et al., 2025), whose reliability depends on student calibration. Trajectory-level methods further use keypoints, subgoals, reference-policy entropy, or fixed action chunks (Fu et al., 2026; Shi et al., 2026; Gao et al., 2026; Becker et al., 2026), often requiring additional structure or supervision. Segment-level modeling remains less explored in OPD (Song & Zheng, 2026). Existing OPD methods typically aggregate reverse Kullback–Leibler(KL) uniformly across tokens or define segments using

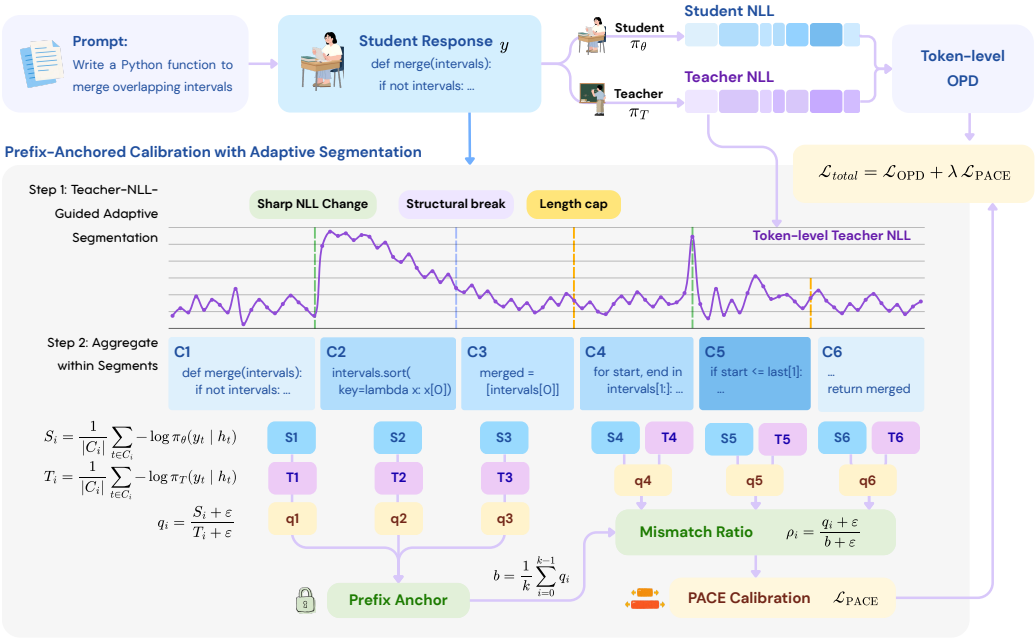


Figure 1: **Overview of PACE.** Alongside token-level OPD, PACE adaptively partitions each student-generated response based on changes in teacher token NLL, structural boundaries, and segment-length constraints. Student and teacher NLLs are averaged within each segment to compute a segment-level mismatch ratio. The first k segments form a detached prefix anchor against which subsequent segment ratios are calibrated. The deviations of later segments from this anchor are aggregated through a softly gated penalty, providing a segment-level calibration loss that complements token-level OPD.

KL thresholds, delimiters, or predefined reasoning steps (Liu et al., 2025a; He et al., 2026; Shi et al., 2026). In contrast, we identify boundaries from changes in teacher token-level surprisal with lightweight structural constraints. This produces coherent local units that reduce token-level noise while preserving finer-grained supervision than sequence-level objectives (Yin et al., 2025). Unlike student-entropy-based segmentation (Yang et al., 2026), PACE defines segments using the teacher distribution that OPD aims to match.

2.2 Mitigating Teacher Bias in Distillation

Prior work on OPD aims to transfer task-relevant knowledge while mitigating teacher bias. One line explores alternative divergences, including reverse KL, Jensen–Shannon divergence, and other f-divergences, to reduce the influence of low-probability teacher outputs (Wang et al., 2023; Agarwal et al., 2024; Ko et al., 2024). Some methods further use student–teacher deviations, perturbed or proxy-teacher losses, and assistant-distribution formulations to reduce teacher bias and stabilize distribution matching (Nagarajan et al., 2023; Kim et al., 2026; Shin et al., 2026). Another line addresses exposure bias and training–inference mismatch through student rollouts, imitation-learning-style aggregation, speculative verification, or adaptive online teachers (Niu et al., 2022; Xu et al., 2025; Zhou et al., 2026). Complementary approaches introduce curriculum-based or fine-grained supervision, including adaptive teacher interpolation (Shing et al., 2025; Li et al., 2026), token- or step-level selective distillation (Dai et al., 2025; Zhang et al., 2025; Huang et al., 2026), and rationale- or process-level objectives for transferring faithful reasoning (Wang et al., 2023; Wu et al., 2024; Lei et al., 2025; Jiang et al., 2026; Deng et al., 2026). These approaches improve the quality or granularity of supervision, but local teacher–student discrepancies can still reflect persistent differences in task difficulty, style, or surface form rather than the real capability gap. We therefore use the discrepancy in initial segments as a response-specific reference and calibrate subsequent segments against it. This within-trajectory calibration downweights persistent mismatch while highlighting capability-relevant deviations, without reward models or human annotations.

3 PRELIMINARIES

3.1 AGENT EXECUTION

Given a task prompt x, a reasoning agent interacts with an external environment through multiple rounds of reasoning, actions, and observations. At interaction step k, the agent generates a reasoning sequence $\mathbf{r}_k$ followed by an action a_k, conditioned on the current interaction history $\mathcal{H}_k$:

$$(\mathbf{r}_k, a_k) \sim \pi_\theta(\cdot \mid \mathcal{H}_k), \tag{1}$$

where π_θ denotes the agent policy. An action may invoke an external tool or provide the final answer. When a tool is invoked, the environment returns an observation o_k, and the interaction history is updated as

$$\mathcal{H}_{k+1} = \mathcal{H}_k \oplus (\mathbf{r}_k, a_k, o_k), \tag{2}$$

where $\oplus$ denotes sequence concatenation. The interaction continues until the agent produces a final answer or reaches a predefined stopping condition.

The resulting execution trajectory is represented as

$$\tau = (x, \mathbf{r}_1, a_1, o_1, \ldots, \mathbf{r}_K, a_K, o_K), \tag{3}$$

where the final action may correspond to the final answer and therefore need not produce an environment observation. We use $\mathbf{y} = (y_1, \ldots, y_L)$ to denote the complete sequence of tokens generated by the agent along the trajectory. It includes reasoning tokens, action tokens, and final-answer tokens, but excludes environment-provided observation tokens. Although observations are not generated by the agent, they are included in the context for subsequent token generation.

3.2 ON-POLICY DISTILLATION

We consider a trainable student policy π_θ and a fixed teacher policy π_{T}. In on-policy distillation, the student agent first generates an execution trajectory using its current policy:

$$\tau \sim \pi_\theta(\cdot \mid x), \qquad y_t \sim \pi_\theta(\cdot \mid h_t), \tag{4}$$

where h_t denotes the complete autoregressive context available at position t. In an agent trajectory, h_t contains the original task prompt, previously generated reasoning and action tokens, and all preceding tool observations.

The teacher evaluates the same student-generated tokens under the same contexts. Therefore, the student receives supervision on the reasoning states and interaction histories induced by its own policy. At each valid generated-token position, the student is encouraged to match the teacher by minimizing their conditional distributional discrepancy:

$$\mathcal{L}_{\mathrm{OPD}} = \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)}\left[\frac{1}{N_\tau}\sum_{t=1}^{L} m_t D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid h_t) \,\|\, \pi_{\mathrm{T}}(\cdot \mid h_t)\right)\right], \tag{5}$$

where $\mathcal{D}$ denotes the task-prompt distribution, $m_t \in \{0, 1\}$ is the generation mask, and N_τ denotes the number of valid student-generated token positions in trajectory τ. We compute the distillation loss only over student-generated tokens, masking out all other positions. By distilling the teacher on student-generated execution trajectories, on-policy distillation aligns the student with the teacher over the reasoning and interaction states that the student actually encounters during agent execution.

3.3 TEACHER NEGATIVE LOG-LIKELIHOOD

Given a student-generated execution trajectory τ, the fixed teacher policy π_{T} evaluates each valid student-generated token under the same autoregressive context h_t. The teacher negative log-likelihood (NLL) at position t is defined as

$$n_t^{\mathrm{T}} = -\log \pi_{\mathrm{T}}(y_t \mid h_t). \tag{6}$$

Teacher NLL measures the surprisal of a student-generated token under the teacher distribution: lower values indicate higher teacher-assigned probability. Across an agent trajectory, it tracks the teacher's predictive confidence over reasoning, action, and final-answer tokens. Unless otherwise specified, we compute it only at valid student-generated positions ($m_t = 1$), excluding prompt tokens, tool observations, and padding.

4 METHOD

OPD aligns the student with the teacher on student-induced execution states at the token level. However, token-level teacher–student discrepancies do not reflect capability alone. Teacher surprisal n_t^{T} can also vary with linguistic style, reasoning strategy, and how a solution is expressed. Consequently, an isolated token-level gap cannot establish whether a mismatch reflects a persistent capability-related divergence or variation in expression, nor can it identify where such divergence concentrates along the trajectory. We therefore lift the calibration to a more stable resolution through a prefix-anchored, segment-relative scheme. Each student trajectory is adaptively partitioned into contiguous segments at points of sharp change in teacher NLL (Section 4.1), yielding units over which per-token noise averages out while stage-level structure is preserved. Within each trajectory, the initial segments define a baseline for the teacher–student mismatch, against which subsequent segments are calibrated to measure relative divergence (Section 4.2). A softly gated corrective term on this signal refines the OPD objective without overriding its original learning signal (Section 4.3). This calibrates capability-related divergence that emerges during generation from both the trajectory's initial teacher–student mismatch and token-level variation.

4.1 TEACHER-NLL-GUIDED ADAPTIVE SEGMENTATION

Token-level teacher–student distributional gaps are too noisy to read a capability signal off directly: they fluctuate with differences in linguistic style, phrasing choices, and reasoning patterns between the two models, which makes per-token assessment unreliable. We therefore adopt a lightweight alternative: each trajectory is partitioned into a small number of contiguous segments driven by teacher NLL, and the gap is aggregated within each segment to obtain a robust segment-level signal. This segmentation reuses quantities already computed by the distillation objective and produces statistics at a resolution where per-token noise cancels.

Consider a student-generated trajectory τ as defined in Section 3.2. Let

$$\mathcal{I}_\tau = (t_{\tau,1}, \ldots, t_{\tau,M_\tau}) \tag{7}$$

denote the ordered sequence of valid student-generated token positions within τ, with $m_{t_{\tau,j}} = 1$ and $M_\tau = |\mathcal{I}_\tau|$. Only tokens produced by the student policy are included in $\mathcal{I}_\tau$, covering reasoning, action, and final-answer tokens in accordance with the generation mask. Prompt tokens, tool observations, padding, and all other masked positions are excluded from segmentation and from the segment-level objective. When the trajectory contains tool calls, we treat each returned tool observation as a hard segmentation barrier: the current segment is forcibly terminated immediately before the observation, and a new segment is initiated at the first student-generated token following it. Consequently, no segment is permitted to straddle a tool-observation boundary, ensuring that segment-level statistics are computed over contiguous spans of student-generated tokens that share a common intra-turn context. Excluded positions are nonetheless retained in the conditioning context, so both teacher and student score each token in $\mathcal{I}_\tau$ under an identical history.

We use the teacher NLL n_t^{T} defined in Equation (6) to identify changes in the teacher's token-level surprisal. For adjacent valid positions within the same trajectory, define

$$\Delta_{\tau,j} = \left| n^{\mathrm{T}}_{t_{\tau,j+1}} - n^{\mathrm{T}}_{t_{\tau,j}} \right|, \qquad j = 1, \ldots, M_\tau - 1. \tag{8}$$

The trajectory-specific boundary threshold is

$$\gamma_\tau = Q_p \left(\{\Delta_{\tau,j}\}_{j=1}^{M_\tau - 1} \right), \tag{9}$$

where Q_p denotes the quantile at level p of the within-trajectory NLL changes.

We scan the valid tokens of τ from left to right and close the current segment after position $t_{\tau,j}$ when either (i) its length reaches the maximum segment length $L_{\max}$, or (ii) its length is at least the minimum segment length $L_{\min}$ and $\Delta_{\tau,j} > \gamma_\tau$. The end of the trajectory closes the remaining segment. This produces an ordered partition $\mathcal{C}_\tau = (C_{\tau,0}, \ldots, C_{\tau,K_\tau - 1})$, where each $C_{\tau,i}$ comprises valid token positions from the same trajectory and K_τ is the number of segments. The final segment may contain fewer than $L_{\min}$ tokens. A trajectory with no adjacent pair of valid tokens forms a single segment.

The minimum-length constraint suppresses spurious boundaries from isolated NLL fluctuations, while the maximum-length constraint avoids overly long segments. We also use newline positions as structural cues when determining segment endpoints. Together, these criteria yield a trajectory-adaptive partition that aggregates likelihood information at a stable local resolution. We use teacher NLL to drive segmentation because abrupt changes in teacher surprisal indicate shifts in the predictive regime, providing natural segment anchors for teacher–student comparison. The validity of this choice is supported by the theoretical analysis in Appendix D and the empirical study in Appendix C.

4.2 Prefix-Calibrated Teacher–Student Mismatch

Directly comparing absolute discrepancies across trajectories risks conflating this persistent, capability-independent component with capability-driven deviations that emerge later in the trajectory. Prior work has observed that the earliest decoding steps of an autoregressive generation disproportionately reflect prompt priors and shallow stylistic continuation rather than extended reasoning (Holtzman et al., 2019; Wang & Zhou, 2024). Building on this observation, we use the opening segments of each trajectory as a trajectory-specific calibration for the intrinsic teacher–student gap. Subsequent segments are evaluated relative to this calibration, reducing capability-independent mismatch in the segment-level signal.

For each segment $C_{\tau,i}$, we compute the mean student and teacher NLL over the *same* student-generated tokens: $S_{\tau,i} = \frac{1}{|C_{\tau,i}|}\sum_{t\in C_{\tau,i}} -\log \pi_\theta(y_t \mid h_t)$ and $T_{\tau,i} = \frac{1}{|C_{\tau,i}|}\sum_{t\in C_{\tau,i}} n_t^{\mathrm{T}}$. For numerical stability, a small positive constant ε is added to the numerator and denominator of every ratio defined below. The segment-level mismatch ratio is

$$q_{\tau,i} = \frac{S_{\tau,i}+\varepsilon}{T_{\tau,i}+\varepsilon}. \tag{10}$$

Here, $q_{\tau,i}$ compares the average difficulty that the student and the teacher assign to the student-generated tokens within segment i, without requiring a teacher-generated reference generation.

Let k denote the number of prefix segments used for calibration. When $K_\tau > k$, we construct a trajectory-specific baseline from these segments:

$$b_\tau = \mathrm{sg}\left(\frac{1}{k}\sum_{i=0}^{k-1} q_{\tau,i}\right), \tag{11}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. We then measure the relative mismatch of each subsequent segment as

$$\rho_{\tau,i} = \frac{q_{\tau,i}+\varepsilon}{b_\tau+\varepsilon}, \qquad i = k, \ldots, K_\tau - 1. \tag{12}$$

A value $\rho_{\tau,i} \approx 1$ indicates that the segment's teacher–student NLL ratio remains close to the trajectory's own initial level, while $\rho_{\tau,i} > 1$ signals a larger relative student NLL than the prefix would predict. Calibrating against b_τ therefore adapts the calibration signal to each trajectory, avoiding a shared absolute threshold that could mistake baseline differences arising from the capability gap for the degradation we aim to detect.

We apply a stop-gradient to b_τ so that the loss cannot be minimized by degrading the prefix in order to lower the comparison anchor. The prefix tokens continue to contribute to the underlying OPD objective, ensuring that their learning signal is preserved. Trajectories with $K_\tau \le k$ contain no post-prefix segments over which the relative mismatch can be evaluated, and are therefore omitted from the calibration penalty while still participating in the OPD objective.

4.3 Segment-Relative Degradation Objective

We retain OPD as the base objective because it provides the teacher's original token-level supervision by aligning the student with the teacher's next-token distribution. However, this signal can be noisy and poorly calibrated at the token level. PACE therefore calibrates OPD with a segment-level relative-mismatch penalty: while OPD supplies fine-grained per-token guidance, the calibration term discourages the teacher–student gap in later segments from exceeding the trajectory-specific prefix baseline. In this way, the final objective preserves the original OPD supervision while adding a calibrated correction for trajectory-level degradation.

OPD minimizes the token-level KL divergence between the student and teacher next-token distributions on student-generated trajectories (Section 3.2), and we retain its trajectory-level loss $\mathcal{L}_{\mathrm{OPD}}(\tau)$ as the base objective in what follows.

PACE complements $\mathcal{L}_{\mathrm{OPD}}$. For every post-prefix segment ($i \geq k$), we introduce a soft gate

$$g_{\tau,i} = \sigma([\min(\rho_{\tau,i}, \rho_{\max}) - \delta_{\deg}]), \tag{13}$$

where $\sigma(\cdot)$ is the sigmoid function and $\delta_{\deg}$ is the relative-mismatch threshold. The gate concentrates the corrective signal on segments with substantial mismatch: it is near zero when $\rho_{\tau,i} \lesssim \delta_{\deg}$, suppressing noisy fluctuations, and saturates toward one when $\rho_{\tau,i} \gg \delta_{\deg}$, focusing calibration on segments where the student diverges most from the prefix baseline. The cap $\rho_{\max}$ bounds only the gate input, preventing extreme $\rho_{\tau,i}$ from dominating the aggregation while leaving the relative mismatch in the calibration unclipped.

The trajectory-level calibration loss is squared log-ratio deviations across post-prefix segments:

$$\mathcal{L}_{\mathrm{PACE}}(\tau) = \begin{cases} \dfrac{\sum_{i=k}^{K_\tau - 1} g_{\tau,i} \, (\log \rho_{\tau,i})^2}{\sum_{i=k}^{K_\tau - 1} g_{\tau,i}}, & K_\tau > k, \\ 0, & K_\tau \leq k. \end{cases} \tag{14}$$

The gate assigns greater weight to segments with larger relative mismatch, while the squared log-ratio measures their deviation from the trajectory-specific baseline. Because the gate is soft, segments below $\delta_{\deg}$ may still contribute to the loss with attenuated weight rather than being excluded outright.

The full training objective combines the base OPD loss with the calibration term:

$$\mathcal{L}_{\mathrm{total}} = \mathbb{E}_{x \sim \mathcal{D}} \, \mathbb{E}_{\tau \sim \pi_\theta(\cdot|x)} \left[\mathcal{L}_{\mathrm{OPD}}(\tau) + \lambda \, \mathcal{L}_{\mathrm{PACE}}(\tau) \right], \tag{15}$$

where λ controls the contribution of the calibration term relative to the base objective. This design preserves the teacher's original token-level supervision on all valid student-generated positions and augments it with a trajectory-calibrated signal that tracks changes in teacher–student mismatch along the generated trajectory. The calibration term reuses OPD rollouts and teacher evaluations, making it a lightweight addition that requires neither task rewards nor a reward model.

5 Experiments

5.1 Experimental Setup

Tasks and datasets. We evaluate our method on agentic tasks. For tool use, we use 4k agentic tool-use data from ToolRL (Qian et al., 2026), following its original training and test splits. We evaluate on ACEBench (Chen et al., 2025), BFCL (Patil et al., 2025), and CRMArena (Huang et al., 2025). For coding, we sample 5k training data from ReflectionSeq-GPT (R-GPT) (Ren et al., 2025) and evaluate on BigCodeBench (BCB) (Zhuo et al., 2025), MBPP (Austin et al., 2021), and test set of ReflectionSeq-GPT. There is no overlap between the coding training and test examples.

Models and evaluation benchmarks. We evaluate two teacher-student configurations: Qwen3.5-27B paired with Qwen3.5-4B, and Qwen3.5-9B paired with Qwen3.5-2B. For each configuration, we compare our method against the base model and several on-policy distillation baselines. Following the official evaluation protocol of each benchmark, we report task-completion accuracy (%), with higher values indicating better performance.

Baselines. We compare our method with on-policy distillation approaches that refine supervision at different granularities. MOPD (Yu et al., 2026) conditions the teacher on successful and failed peer rollouts for the same prompt. OPD^2 (Heo et al., 2026) distills the difference between a fine-tuned teacher and its base model. SOPD (Sun et al., 2026) provides step-level corrections along student-generated trajectories, bridging supervised fine-tuning and OPD.

Implementation Details. We set the boundary quantile to $p = 0.9$, the minimum and maximum segment lengths to $L_{\min} = 16$ and $L_{\max} = 48$, respectively, and the number of calibration segments to $k = 3$. For the soft gate, we use $\delta_{\deg} = 1.15$, $\rho_{\max} = 5.0$, and $\lambda = 0.1$. All methods are trained for a single epoch with a batch size of 12 on 8 NVIDIA H100 GPUs. Each teacher–student configuration uses 2 GPUs for teacher inference and 6 for student optimization.

Table 1: Results on tool-use and coding benchmarks (%; higher is better). Best and second-best scores within each student scale are **bolded** and underlined, respectively.

	Agentic Tool Use			Coding		
Method	**ACEBench**	**BFCL**	**CRMArena**	**BCB**	**MBPP**	**R-GPT**
4B Student with a 27B Teacher						
Base	66.95	67.10	12.80	34.00	69.20	14.23
OPD	67.56	71.90	16.00	33.50	72.20	18.76
MOPD	68.11	73.50	17.40	34.50	74.10	19.40
OPD^2	66.25	72.66	15.80	34.50	73.48	17.36
SOPD	68.44	74.80	18.20	34.00	72.84	19.80
PACE (Ours)	**69.84**	**79.30**	**21.60**	**37.50**	**75.60**	**22.36**
2B Student with a 9B Teacher						
Base	58.32	55.80	6.40	18.50	40.30	10.48
OPD	59.54	64.60	8.60	20.50	41.40	11.80
MOPD	60.09	64.20	10.00	20.00	42.30	12.24
OPD^2	58.83	62.36	8.40	19.50	41.68	10.40
SOPD	62.42	65.50	10.40	**22.50**	42.04	12.24
PACE (Ours)	**66.80**	**71.30**	**11.60**	22.00	**43.60**	**13.72**

5.2 Main Results

Agentic tool use. Table 1 shows that our method consistently achieves the best performance across all agentic tool-use benchmarks and both student scales. Compared with OPD, the improvement reaches up to 7.40 percentage points on BFCL for the 4B student and 7.26 points on ACEBench for the 2B student. Notably, our method delivers a maximum relative gain of 35.0% on CRMArena, demonstrating its effectiveness in challenging multi-step tool-use scenarios.

The baseline performance highlights the importance of supervision granularity. MOPD generally improves over OPD, but its rollout-level signal spans distinct reasoning and interaction stages, resulting in modest gains. OPD^2 isolates the effect of reasoning tuning but does not directly address action selection or observation handling. It even underperforms the OPD baseline on some tool-use tasks. SOPD is the strongest baseline in most tool-use settings, demonstrating the value of step-level correction. Nevertheless, reasoning-step boundaries do not consistently align with transitions among planning, tool invocation, observation interpretation, and plan revision. Our method instead partitions trajectories into locally coherent segments and anchors each segment's student-to-teacher NLL ratio to an early-segment baseline. This separates capability-related divergence from the initial systemic student–teacher mismatch while retaining local supervision. The consistent gains on BFCL and CRMArena, both of which require multi-step tool interaction, support the combined design.

Coding. Our method improves coding performance at both student scales. Coding trajectories often contain extended, semantically coherent spans, whereas tool-use trajectories can undergo semantic shifts both between tool calls and within individual steps. More frequent transitions create greater opportunities for segment-level supervision to identify and address localized performance gaps. The coding gains indicate that the method remains effective even when such transitions are less frequent.

5.3 Ablation Studies

We conduct ablation studies with Qwen3.5-27B as the teacher model and Qwen3.5-4B as the student model, examining the effects of calibration-term strength, segmentation strategy, baseline construction, and the gating mechanism.

Main comparison and regularization weight. Although performance decreases as λ increases, our method consistently outperforms token-level OPD across different λ settings, suggesting that segment-level calibration complements the original distribution-matching objective. Since PACE calibrates rather than replaces the teacher–student distribution gap, its benefit is strongest under moderate regularization. Excessive segment-level regularization can dominate fine-grained token-level supervision and weaken this complementary effect, thereby degrading performance.

Table 2: Ablation results on ACEBench and BFCL Multi-Step.

Variant	ACE	BFCL
Main comparison		
4B Base	66.95	67.10
4B Base + OPD	67.56	71.90
PACE(Ours)	**69.84**	**79.30**
Regularization weight		
$\lambda = 0.2$	68.12	78.10
$\lambda = 0.3$	68.46	77.60
Segmentation		
Random	59.64	65.40
Fixed window (32)	67.77	73.20
Student–teacher gap	68.04	74.80
Student NLL	67.88	74.26
Only structural boundaries	67.50	74.10
Baseline and gating		
Baseline $K = 1$	68.96	77.32
Baseline $K = 5$	68.10	74.16
Without sigmoid gate	67.82	72.80

Segmentation strategy. Teacher-NLL-based segmentation performs best across both benchmarks. **Fixed window** places boundaries every 32 tokens regardless of content, so it cannot respond to changes in predictive difficulty. **Student–teacher gap** triggers a cut when the token-level student–teacher NLL difference jumps sharply, but the resulting boundaries shift as the student policy is optimized. **Student NLL** replaces the teacher NLL in our change-point rule with the student's own NLL, inheriting the same instability from the moving student policy. **Only structural boundaries** segments purely at newlines and tool-observation breaks, capturing explicit formatting or interaction events but missing changes within those spans. **Random segmentation** places boundaries uniformly at random, and the marked drop in performance underscores the importance of where segment boundaries are placed.

Baseline and gating. The default early-segment baseline outperforms both shorter and longer alternatives. A shorter baseline is more sensitive to local fluctuations, whereas an overly long one incorporates more reasoning content, introducing signals that reflect differences in model capability. The intermediate baseline therefore provides a more robust reference. Removing the sigmoid gate also reduces performance, particularly on BFCL, indicating that downweighting small fluctuations around the baseline makes the segment-level signal more robust.

5.4 Analysis results

We further provide diagnostic analyses that empirically substantiate the key design choices underlying PACE. Appendix C.1 shows that token-level OPD signals fluctuate within coherent coding blocks, supporting teacher-NLL-based segment aggregation. Appendix C.3 shows that high-ρ segments align with failure-relevant regions: the first high-ρ segment localizes error onset with a 74.74% hit rate, and 77.55% of high-ρ segments are judged erroneous. Appendices C.5 and C.4 further support the prefix baseline: early student–teacher ratios show little separation across 2B, 4B, and 9B students, and the first 60 tokens contain 69.7% meta-nuisance content. These results suggest that PACE captures failure-relevant deviations, while the prefix anchor mainly reflects stable trajectory-level mismatch rather than substantive teacher–student capability gaps.

6 Conclusion

Token-level teacher–student discrepancies in OPD are noisy and often conflate stable stylistic differences with genuine capability degradation, making it difficult to localize where actual capability gaps between the student and the teacher. To address this problem, we propose PACE, which adaptively segments each response according to changes in teacher negative log-likelihood and uses the initial segments as a trajectory-specific anchor to calibrate the relative mismatch of subsequent segments. This design separates stable trajectory-level mismatch from in-trajectory degradation and better identifies capability gaps between the teacher and the student, without requiring any reward model or human annotation. Experiments on agentic tool-use and coding benchmarks demonstrate consistent improvements over OPD baselines. Diagnostic analyses further show that adaptive segmentation yields semantically coherent units, while prefix-anchored calibration provides a stable trajectory-specific reference.

Ethics Statement

This paper presents contributions to the field of LLM agents. We do not anticipate direct negative societal impacts from this specific work. All datasets and benchmarks employed in our experiments are publicly available and contain no sensitive or personally identifiable information.

Generative AI Use Disclosure

Generative AI tools were used solely for polishing and proofreading the manuscript to improve grammatical fluency and clarity. They were not used to generate research ideas or experimental results. All content has been reviewed and verified by the authors, who take full responsibility for this work.

Reproducibility Statement

To facilitate reproducibility, complete implementation details, including hyperparameters, model configurations, and datasets, are provided in the experimental setup section (Section 5). The source code will be released publicly upon publication.

References

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

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc N. M Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why, 2026. URL https://arxiv.org/abs/2605.10889.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Philipp Becker, Niklas Freymuth, Serge Thilges, Fabian Otto, and Gerhard Neumann. Troll: Trust regions improve reinforcement learning for large language models. In *International Conference on Learning Representations*, volume 2026, pp. 53096–53135, 2026.

Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine learning*, 34(1):177–210, 1999.

Yi Bin, Tianyi Jiang, Yujuan Ding, Kainian Zhu, Fei Ma, Jingkuan Song, Yang Yang, and Heng Tao Shen. Explore briefly, then decide: Mitigating llm overthinking via cumulative entropy regulation. *arXiv preprint arXiv:2510.02249*, 2025.

Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Yuefeng Huang, et al. Acebench: Who wins the match point in tool usage? *arXiv preprint arXiv:2501.12851*, 2025.

Kenneth Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. Capture the key in reasoning to enhance cot distillation generalization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 441–465, 2025.

Yihe Deng, I Hsu, Jun Yan, Zifeng Wang, Rujun Han, Gufeng Zhang, Yanfei Chen, Wei Wang, Tomas Pfister, Chen-Yu Lee, et al. Supervised reinforcement learning: From expert trajectories to step-wise reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 75893–75911, 2026.

Preston Fu, Kevin Frans, Oleh Rybkin, Sergey Levine, and Aviral Kumar. Long-horizon language model reinforcement learning via progressive point matching. *arXiv preprint arXiv:2609.07303*, 2026.

Heyang Gao, Zexu Sun, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Xu Chen. Solving the granularity mismatch: Hierarchical preference learning for long-horizon llm agents. In *International Conference on Learning Representations*, volume 2026, pp. 67081–67110, 2026.

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

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. Treerl: Llm reinforcement learning with on-policy tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369, 2025.

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

Kung-Hsiang Huang, Akshara Prabhakar, Sidharth Dhawan, Yixin Mao, Huan Wang, Silvio Savarese, Caiming Xiong, Philippe Laban, and Chien-Sheng Wu. Crmarena: Understanding the capacity of llm agents to perform professional crm tasks in realistic environments. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3830–3850, 2025.

Zhaoyang Jiang, Xuanqi Peng, Fei Teng, Zhizhong Fu, Yunsoo Kim, Jiacong Mi, Zicheng Li, and Honghan Wu. Better accuracies, worse reasoning: A step-level audit of medical chain-of-thought distillation. *arXiv preprint arXiv:2605.28301*, 2026.

Yeongmin Kim, Donghyeok Shin, Mina Kang, Byeonghu Na, and Il-Chul Moon. Distillation of large language models via concrete score matching. In *International Conference on Learning Representations*, volume 2026, pp. 16649–16682, 2026.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Zhenyu Lei, Zhen Tan, Song Wang, Yaochen Zhu, Zihan Chen, Yushun Dong, and Jundong Li. Learning from diverse reasoning paths with routing and collaboration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2832–2845, 2025.

Guanghao Li, Wenhao Jiang, Mingfeng Chen, Yan Li, Hao Yu, Shuting Dong, Tao Ren, Ming Tang, and Chun Yuan. Scout: Teaching pre-trained language models to enhance reasoning via flow chain-of-thought. *Advances in Neural Information Processing Systems*, 38:95340–95364, 2026.

Jun Liu, Zhenglun Kong, Peiyan Dong, Changdi Yang, Tianqi Li, Hao Tang, Geng Yuan, Wei Niu, Wenbin Zhang, Pu Zhao, et al. Structured agent distillation for large language model. *arXiv preprint arXiv:2505.13820*, 2025a.

Qiyuan Liu, Hao Xu, Xuhong Chen, Wei Chen, Yee Whye Teh, and Ning Miao. Enhancing large language model reasoning with reward models: An analytical survey. *arXiv preprint arXiv:2510.01925*, 2025b.

Yuliang Liu, Junjie Lu, Zhaoling Chen, Chaofeng Qu, Jason Klein Liu, Chonghan Liu, Zefan Cai, Yunhui Xia, Li Zhao, Jiang Bian, et al. Adaptivestep: Automatically dividing reasoning step through model confidence. *arXiv preprint arXiv:2502.13943*, 2025c.

Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. Let's reward step by step: Step-level reward model as the navigators for reasoning. *arXiv preprint arXiv:2310.10080*, 2023.

Vaishnavh Nagarajan, Aditya K Menon, Srinadh Bhojanapalli, Hossein Mobahi, and Sanjiv Kumar. On student-teacher deviations in distillation: does it pay to disobey? *Advances in Neural Information Processing Systems*, 36:5961–6000, 2023.

Yulei Niu, Long Chen, Chang Zhou, and Hanwang Zhang. Respecting transfer gap in knowledge distillation. *Advances in Neural Information Processing Systems*, 35:21933–21947, 2022.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *Advances in Neural Information Processing Systems*, 38:105523–105553, 2026.

Houxing Ren, Mingjie Zhan, Zhongyuan Wu, Aojun Zhou, Junting Pan, and Hongsheng Li. Reflectioncoder: Learning from reflection sequence for enhanced one-off code generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9999–10020, 2025.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 60808–60838, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Tianyuan Shi, Canbin Huang, Bei Li, Xin Chen, Xiaojun Quan, Jingang Wang, and Qifan Wang. Beyond trajectory imitation: Strategy-guided policy optimization for llm reasoning. *arXiv preprint arXiv:2606.24064*, 2026.

Donghyeok Shin, Yeongmin Kim, Suhyeon Jo, Byeonghu Na, and Il-Chul Moon. AMiD: Knowledge distillation for LLMs with α-mixture assistant distribution. In *International Conference on Learning Representations (ICLR)*, pp. 51469–51493, 2026.

Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. Taid: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. In *International Conference on Learning Representations*, volume 2025, pp. 92233–92256, 2025.

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

Changhui Sun, Lanbo Liu, Hang Lei, Tong Ling, Jiahang Xie, Zhiyong Zheng, Yujia Wang, Hao Liu, Feng Xiao, Lu Liu, et al. Step-level on-policy distillation: Interpolating between on-policy distillation and supervised fine-tuning. *arXiv preprint arXiv:2608.16333*, 2026.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. Scott: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5546–5558, 2023.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026.

Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems*, 37:66383–66409, 2024.

Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan Rossi, Sungchul Kim, Anup Rao, and Julian McAuley. Decot: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14073–14087, 2024.

Yutao Xie, Nathaniel Thomas, Nick Hansen, Yang Fu, Li Li, and Xiaolong Wang. Tips: Turn-level information-potential reward shaping for search-augmented llms. In *International Conference on Learning Representations*, volume 2026, pp. 156549–156584, 2026.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, volume 2025, pp. 64616–64646, 2025.

Chen Yang, Haiyuan Wan, Rengrong Xiong, Yize Chen, and Danny HK Tsang. Beyond imitation: Filtering on-policy distillation by reasoning progress. *arXiv preprint arXiv:2608.19408*, 2026.

Yueqin Yin, Shentao Yang, Yujia Xie, Ziyi Yang, Yuting Sun, Hany Awadalla, Weizhu Chen, and Mingyuan Zhou. Segmenting text and learning their rewards for improved rlhf in language model. *arXiv preprint arXiv:2501.02790*, 2025.

Weichen Yu, Xiaomin Li, Yizhou Zhao, Xiaoze Liu, Ruowang Zhang, Haixin Wang, Yinyi Luo, Chen Henry Wu, Gaurav Mittal, Matt Fredrikson, et al. Multi-rollout on-policy distillation via peer successes and failures. *arXiv preprint arXiv:2605.12652*, 2026.

Songming Zhang, Xue Zhang, Tong Zhang, Bojie Hu, Yufeng Chen, and Jinan Xu. Alignditil: Token-level language model alignment as adaptive policy distillation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19791–19807, 2025.

Yuhang Zhou, Lizhu Zhang, Yifan Wu, Mingyi Wang, Bo Peng, Jiayi Liu, Xiangjun Fan, and Zhuokai Zhao. Omniopd: Logit-free on-policy distillation via speculative verification. *arXiv preprint arXiv:2606.01476*, 2026.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, volume 2025, pp. 66602–66656, 2025.

A Limitations

Our adaptive segmentation cannot fully eliminate noise arising from teacher–student mismatch, but segment-level aggregation and trajectory-specific anchor calibration help mitigate and partially correct for its effects. A segment may still contain both incidental probability differences and genuine disagreements about the continuation. Moreover, teacher–student disagreement does not necessarily reflect a true capability gap; it may instead arise from differences in calibration, wording preferences, or generation contexts. Although our method mitigates noise arising from inherent teacher–student mismatch, it cannot eliminate it entirely.

B More Experiments

B.1 Training Setup

We implement our method using the VolcEngine Reinforcement Learning (VeRL) framework (Sheng et al., 2025). We use a learning rate of 1×10^{-6} and retain the default OPD settings for all other hyperparameters.

Table 3: Ablation results on ACEBench and BFCL multi-step using Qwen3.5-27B as the teacher and Qwen3.5-4B as the student.

Method	ACEBench	BFCL Multi-step
Base	66.95	67.10
OPD	67.56	71.90
Fixed window, $w = 16$ tokens	67.34	72.12
Fixed window, $w = 32$ tokens	67.77	73.20
Fixed window, $w = 48$ tokens	67.10	71.84
Token-level gating	68.16	73.54
Teacher-student difference	68.58	74.22
PACE	**69.84**	**79.30**

B.2 Ablation Studies

Table 3 compares PACE with standard OPD and several segmentation and gating variants. PACE performs best on both benchmarks. Fixed-window segmentation yields smaller and window-size-dependent improvements, suggesting that static windows do not capture response-specific changes in teacher surprisal as effectively as adaptive segmentation. To isolate the effect of optimization granularity, the token-level gating variant applies the same method at the token level rather than the segment level, with all other settings unchanged. Although this variant improves over standard OPD, it underperforms PACE on both benchmarks. This comparison suggests that applying the method to adaptively defined segments is more effective than applying it to individual tokens.

Segment-level ratio versus difference. We examine the choice of segment-level mismatch measure by replacing the NLL ratio $q_{\tau,i} = (S_{\tau,i} + \varepsilon)/(T_{\tau,i} + \varepsilon)$ with the NLL difference $q^{\text{diff}}_{\tau,i} = S_{\tau,i} - T_{\tau,i}$, while retaining the remaining components of PACE. As shown in Table 3, the ratio-based variant outperforms the difference-based variant. The difference measures mismatch on the absolute NLL scale, which can vary substantially across segments. Although prefix calibration accounts for the overall mismatch scale of a trajectory, it does not normalize these local scale variations. In contrast, the ratio normalizes student NLL by teacher NLL within each segment, making mismatch signals more comparable across segments. The results suggest that this local scale normalization is important for identifying relative degradation.

C Experimental Analysis

C.1 Token-Level Fluctuations

A motivation for segment-level optimization is that token-level OPD loss can fluctuate within a code block that represents a coherent programming decision. We inspect four selected trajectories containing negative token-level OPD deltas to illustrate this behavior. A negative delta reflects a local token-level signal but does not, on its own, establish that the surrounding code block is a poor programming decision. Segment-level optimization evaluates the OPD signal over a semantically coherent unit rather than letting isolated token-level fluctuations dominate the assessment.

Numerical assignment. The first semantic block pairs a comment describing the intended tolerance with its assignment:

```
# Allow for small floating-point errors
eps = 1e-9
```

The block contains one negative-delta token: the minus sign in `1e-9`. The teacher nevertheless predicts that same minus sign as its top-1 token. The local negative signal thus occurs within a numerical literal, rather than at a plausible boundary between distinct programming decisions. This case illustrates how a token-level sign change can arise inside an otherwise coherent block without indicating that the teacher prefers a different token.

Return-and-comment block. The second block comprises a return statement and an adjacent comment describing an edge case:

```
return default
# Treat empty string as "not set"
```

This trajectory passes execution. Of the six negative-delta tokens in the block, five retain teacher–student top-1 agreement. These cases occur mainly around the line transition and local comment wording, including `not`, `set`, and quotation marks. Negative signals therefore appear at several positions within a successful local continuation, despite the absence of a corresponding top-1 preference change at most of those positions. Execution success does not establish that every token is optimal, but it cautions against interpreting each negative delta as a separate consequential error.

Function-introduction block. The same pattern appears in a block introducing a function:

```
# Function to print a matrix.
def print_matrix(matrix):
```

All four negative-delta tokens—`to`, `matrix`, `print`, and `_matrix`—retain top-1 agreement with the teacher. The local sign changes span both the natural-language comment and the function header, yet none corresponds to a different teacher top-1 token. This example suggests that the sign of an individual token-level delta, by itself, is an unreliable indicator of a semantic decision boundary.

Import block. A final block provides an important qualification:

```
from xml.etree.ElementTree import fromstring, ParseError
import os
```

Among its six negative-delta tokens, three retain top-1 agreement, whereas three have a different teacher-preferred token. The block thus combines local fluctuations with substantive preference differences. Segment-level aggregation may reduce sensitivity to the former, but indiscriminately reinforcing the entire block could obscure the latter.

Taken together, these selected cases show that negative token-level OPD deltas can occur within coherent semantic blocks even when the teacher's top-1 prediction agrees with the generated token. This observation motivates using semantic blocks as optimization units, while the mixed signals in trajectory 30 caution against reinforcing every block unconditionally. Beyond these qualitative examples, our ablation study in Section 5.3 compares token-level gating, fixed-length segmentation, and adaptive segmentation; the results show that our adaptive-segment method outperforms both alternatives.

C.2 Analysis of Teacher-Student Ratio Dynamics

Experimental Setup. We analyze 300 failed trajectories selected from student on-policy rollouts on coding tasks. For each trajectory, we form segments using changes in teacher NLL and compute the ratio of mean student NLL to mean teacher NLL for each segment, denoted by $r_i = S_i/T_i$. The average ratio over the first three segments defines a trajectory-specific baseline, $b = (r_0 + r_1 + r_2)/3$. For each subsequent segment, we compute its relative degradation coefficient $\rho_i = r_i/b$. We group subsequent segments by normalized position and measure the robust standard deviation of ρ_i across trajectories in each position bin. We also report the robust standard deviation of b for comparison.

Analysis. Figure 2 shows that the early segment baseline has relatively low variability across failed trajectories. In contrast, the relative degradation coefficient exhibits greater variability, which tends to increase at later positions. This pattern suggests that early teacher and student NLL ratios capture a relatively stable component of their discrepancy, while additional, trajectory-dependent differences become more pronounced as generation proceeds.

These observations motivate using the early ratio as a within-trajectory reference. Normalizing later ratios by this reference allows the regularizer to focus on changes in teacher and student discrepancy along the trajectory, rather than its initial magnitude. The figure supports this design motivation, although variability alone does not establish that the later differences are caused by a capability gap.

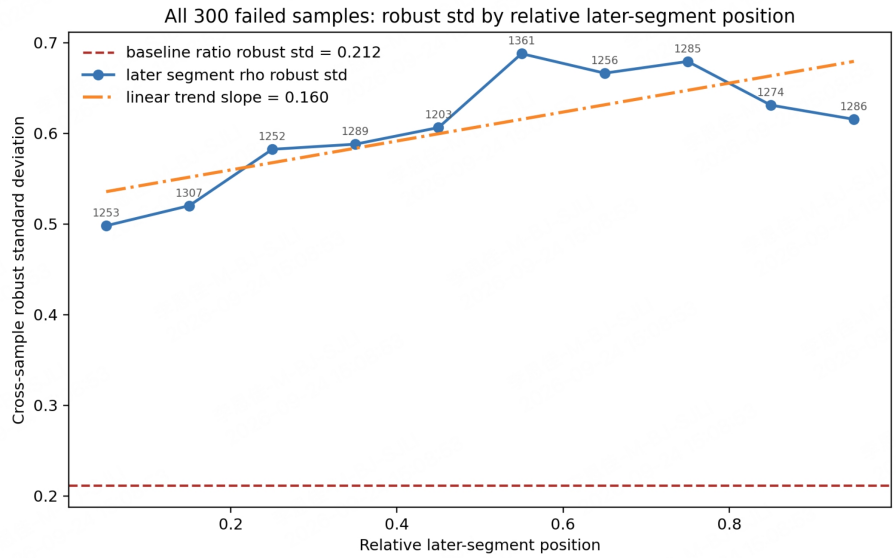


Figure 2: Variability of the student to teacher NLL ratio across selected failed trajectories. Later segments are grouped by their normalized position in the response. The horizontal reference line shows the robust standard deviation of the baseline ratio, while the fitted line summarizes the positional trend in the robust standard deviation of the relative degradation coefficient ρ_i. The baseline varies less across trajectories, whereas the variability of ρ_i generally increases toward later positions.

C.3 Localizing Errors with High Ratio Segment

Error onset localization. We first examine whether segment-level relative degradation can localize the onset of failure. We sample 300 failed student on-policy trajectories from coding tasks and use a stronger model (GPT-5.6) to annotate where each trajectory first becomes erroneous. We then map each annotated point to its corresponding segment. The segment-level method predicts the first segment with a high relative degradation coefficient ρ_i. The token-level baseline predicts the first token with a large token-level OPD discrepancy, which we map to the segment containing that token. We evaluate both predictions on the same trajectories using the same criterion. A prediction is a hit if it falls in the annotated segment or either adjacent segment. The segment-level method achieves a 74.74% hit rate, whereas the token-level baseline achieves 25.38%. Thus, the first large token-level discrepancy often falls outside the neighborhood of the actual failure onset, even after it is mapped to a segment. Aggregating discrepancies over coherent segments provides a more reliable signal for locating where the trajectory begins to fail.

Semantic relevance of high-degradation segments. We separately sample 300 trajectories without filtering by correctness and ask the annotator model to assess the content of their high-ρ segments. The annotator identifies 77.55% of these segments as erroneous. Thus, large relative degradation is frequently associated with incorrect content, including in a sample not restricted to failed trajectories. Together, the two analyses support segment-level relative degradation as a signal for failure-relevant behavior, while leaving open whether every high-ρ segment reflects a reasoning error.

C.4 The Early Segments as a Baseline

To assess whether the initial teacher-relative ratio captures differences in student model capability, we evaluate Qwen3.5 student models at three scales (2B, 4B, and 9B), using Qwen3.5-27B as a common teacher. As shown in Figure 3, the three curves remain close to $q = 1$ and show negligible separation in segment 0. Because prefix lengths vary across trajectories, the initial anchored segment (segment 0) contains approximately 60 response tokens on average. The three curves nearly overlap in this initial segment, suggesting that it provides little discriminative signal about student model size. Instead, it serves as the starting reference for the teacher-relative metric. The clearer separation in

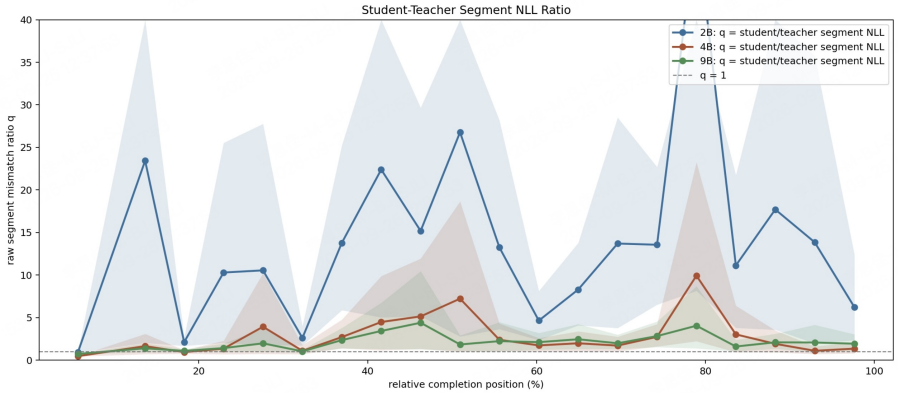


Figure 3: Teacher-relative ratios for 2B, 4B, and 9B student models evaluated against a shared 27B teacher. The horizontal axis denotes relative segment position within each response, from the beginning (0) to the end (1). The curves are computed from 300 randomly sampled trajectories. Segment 0 covers the first 60 response tokens.

subsequent segments suggests that scale-dependent differences become more apparent as the response progresses. Thus, the initial segment is more useful as a baseline for measuring later divergence than as a standalone indicator of model capability.

C.5 Content Composition of Early Segments

A central premise of PACE is that the early-segment baseline captures *style- and format-level nuisance* rather than the student's reasoning capability. To assess this directly, we analyze the content composition of the first 60 and first 100 completion tokens across 500 sampled responses. Each token span is classified by a rule-based tagger into five categories: *boilerplate* (formatted openings), *restatement* (problem paraphrasing), *planning* (generic strategy statements), *code*, and *substantive* (actual reasoning content). We denote meta-nuisance = boilerplate + restatement + planning.

Table 4: Average category share within the early tokens of each response.

Category	First 60 tokens	First 100 tokens
Boilerplate	0.205	0.178
Restatement	0.444	0.401
Planning	0.108	0.132
Code	0.102	0.122
Substantive	0.141	0.167
Boilerplate + Restatement	0.589	0.519
Meta-nuisance	**0.697**	**0.651**

Table 5: Sample-level nuisance dominance in early response tokens.

Metric	First 60 tokens	First 100 tokens
% responses with meta-nuisance > 0.60	0.670	0.598
% responses with boilerplate + restatement > 0.60	0.522	–

Early segments are dominated by non-substantive content. We examine the content of early response segments to assess whether they primarily reflect reasoning ability or trajectory-level nuisance factors. As shown in Table 4, boilerplate and problem restatement account for 58.9% of the first 60 tokens on average. Including generic planning raises the meta-nuisance share to 69.7%, whereas substantive reasoning accounts for only 14.1%. This pattern also holds at the response level:

in Table 5, 67.0% of responses have a meta-nuisance share above 0.60 in their first 60 tokens, and 52.2% exceed that threshold using boilerplate and restatement alone.

The pattern is strongest near the response boundary. When the prefix is shortened from 100 to 60 tokens, the mean meta-nuisance share increases from 0.651 to 0.697, and the fraction of responses with a meta-nuisance share above 0.60 rises from 0.598 to 0.670. Thus, the earliest tokens are particularly likely to contain formatting, restatement, and generic planning, rather than task-specific reasoning. Representative response openings include *"Below is a Python implementation that…"*, *"Problem Restatement: We are given…"*, and *"Let's go step by step."*

D Theoretical Analysis

D.1 Motivation for Teacher-Guided Segmentation Trigger

Consider a student-generated trajectory $y_{1:T} \sim \pi_\theta(\cdot \mid x)$, and let h_t denote the complete autoregressive context preceding token y_t. We evaluate each observed token under the teacher policy π_T and define its teacher-assigned negative log-likelihood (NLL) as

$$n_t^T = -\log \pi_T(y_t \mid h_t). \tag{16}$$

Because the teacher serves as the reference policy for distillation, n_t^T provides a teacher-based assessment of each student-generated continuation. We seek locations where this assessment changes sharply and use them to propose boundaries along the student trajectory.

Teacher-defined predictive association. Pointwise mutual information (PMI) provides a measure of association relative to a marginal baseline and has been used to analyze lexical associations in language (Church & Hanks, 1990). To interpret teacher NLL through this lens, consider the context-independent baseline

$$q_T(v) = \frac{1}{T}\sum_{u=1}^{T} \pi_T(v \mid h_u). \tag{17}$$

This is the token marginal induced by the teacher when a context is drawn uniformly from the observed student-generated prefixes. We introduce q_T for analysis; computing it is not required to obtain the teacher-NLL boundary scores.

The corresponding teacher-defined pointwise association score for the observed token is

$$I_t^T = \log \frac{\pi_T(y_t \mid h_t)}{q_T(y_t)}. \tag{18}$$

Under the joint distribution induced by the empirical prefix distribution and the teacher conditional policy, I_t^T has the form of pointwise mutual information between context and token. Here, we evaluate that score at a token sampled by the student. A larger I_t^T means that the teacher assigns more probability to y_t in its actual context than it does under the trajectory-level baseline.

Let $\tilde{h}_t = -\log q_T(y_t)$ denote the baseline surprisal of the observed token. The teacher NLL then admits the exact decomposition

$$n_t^T = \tilde{h}_t - I_t^T. \tag{19}$$

For $t > 1$, its adjacent difference is

$$J_t = n_t^T - n_{t-1}^T = \underbrace{(\tilde{h}_t - \tilde{h}_{t-1})}_{\Delta\tilde{h}_t} - \underbrace{(I_t^T - I_{t-1}^T)}_{\Delta I_t^T}. \tag{20}$$

A decrease in teacher-defined predictive association ($\Delta I_t^T < 0$) contributes positively to the teacher-NLL jump J_t. However, because adjacent NLLs evaluate different observed tokens, J_t also contains the change in baseline surprisal $\Delta\tilde{h}_t$. A positive jump alone therefore does not imply a decrease in predictive association.

Prior text-segmentation methods motivate the use of local changes in lexical cohesion or topicality to identify boundaries (Church & Hanks, 1990; Beeferman et al., 1999). Drawing on this change-based

perspective, we make a working assumption specific to teacher-evaluated student trajectories: at informative transitions, the change in teacher-defined predictive association dominates the concurrent variation in baseline token surprisal. When $|\Delta\tilde{h}_t| \ll |\Delta I_t^T|$, the jump decomposition in Eq. (20) gives

$$J_t \approx -\Delta I_t^T. \tag{21}$$

Under this assumption, a large absolute teacher-NLL jump reflects a pronounced change in teacher-defined predictive association. Positive and negative jumps correspond to decreases and increases in this association, respectively. We use the jump magnitude $|J_t|$ to propose segment boundaries in either direction. The resulting boundaries reflect changes in teacher-assigned surprisal and are not assumed to coincide with semantic transitions.

D.2 From Teacher-NLL Boundaries to Local Aggregation

The preceding analysis interprets teacher-NLL jumps as candidate transition signals. We now relate these signals to the subsequent averaging operation, without requiring the predictive-association assumption.

For a segment $C_{\tau,i}$ of length $m_{\tau,i}$, define its mean teacher NLL and averaging error as

$$T_{\tau,i} = \frac{1}{m_{\tau,i}} \sum_{t \in C_{\tau,i}} n_t^T, \qquad E_{\tau,i} = \frac{1}{m_{\tau,i}} \sum_{t \in C_{\tau,i}} (n_t^T - T_{\tau,i})^2. \tag{22}$$

If adjacent tokens $t-1$ and t belong to the same segment, then

$$(n_t^T - T_{\tau,i})^2 + (n_{t-1}^T - T_{\tau,i})^2 \geq \frac{(n_t^T - n_{t-1}^T)^2}{2}, \tag{23}$$

which gives

$$E_{\tau,i} \geq \frac{J_t^2}{2m_{\tau,i}}. \tag{24}$$

A pronounced jump retained within a segment therefore contributes a lower bound on the error of representing that segment by its mean. This connects jump-based boundary selection to the local aggregation used by PACE, without assuming that token NLLs are smooth within each segment.

The threshold rule admits a related interpretation. Let B contain indices at which a new segment begins, and consider

$$\mathcal{J}_\gamma(B) = \sum_{\substack{t=2 \\ t \notin B}}^{T} J_t^2 + \gamma^2 |B|. \tag{25}$$

The first term measures adjacent variation retained within segments, while the second penalizes additional boundaries. Without structural or length constraints, adding a boundary at t changes this objective by $\gamma^2 - J_t^2$, favoring a cut when $|J_t| > \gamma$. Selecting γ through a trajectory-specific quantile adapts this tradeoff to the observed variation. This provides an interpretation of the jump criterion, rather than an optimality claim for PACE's full segmentation procedure, which also incorporates structural boundaries and segment-length constraints.

D.3 Scale-Invariant Ratios and Prefix Calibration

Let $n_t^S = -\log \pi_\theta(y_t \mid h_t)$ denote the student NLL on the same observed tokens, and define

$$S_{\tau,i} = \frac{1}{m_{\tau,i}} \sum_{t \in C_{\tau,i}} n_t^S, \qquad q_{\tau,i} = \frac{S_{\tau,i}}{T_{\tau,i}}. \tag{26}$$

We omit numerical stabilizers and assume positive NLLs in the following analysis.

Aggregating before normalization. The segment ratio can be expressed as

$$q_{\tau,i} = \frac{\sum_{t\in C_{\tau,i}} n_t^S}{\sum_{t\in C_{\tau,i}} n_t^T} = \sum_{t\in C_{\tau,i}} \frac{n_t^T}{\sum_{u\in C_{\tau,i}} n_u^T} \frac{n_t^S}{n_t^T}. \tag{27}$$

Thus, PACE pools NLLs before division and yields a teacher-NLL-weighted average of token ratios. Individual tokens with small teacher NLLs receive correspondingly small weights, avoiding equal weighting of ratios with potentially unstable token-level denominators.

For a fixed partition, common positive scaling $(S_{\tau,i}, T_{\tau,i}) \mapsto (c_{\tau,i}S_{\tau,i}, c_{\tau,i}T_{\tau,i})$ leaves $q_{\tau,i}$ unchanged, whereas the difference $S_{\tau,i} - T_{\tau,i}$ scales by $c_{\tau,i}$. This motivates the ratio as a measure of the relative student–teacher NLL relationship. More generally, a statistic satisfying $F(cS, cT) = F(S, T)$ for every $c > 0$ can be written as $F(S, T) = f(S/T)$, obtained by setting $c = 1/T$.

Using the prefix as a relative reference. With segments indexed from zero, the prefix anchor and calibrated ratios are

$$b_\tau = \mathrm{sg}\left(\frac{1}{k}\sum_{j=0}^{k-1} q_{\tau,j}\right), \qquad \rho_{\tau,i} = \frac{q_{\tau,i}}{b_\tau}, \quad i \geq k. \tag{28}$$

If all segment ratios are multiplied by a common trajectory-specific factor $a_\tau > 0$, the anchor scales by the same factor and $\rho_{\tau,i}$ remains unchanged. For example, consider the decomposition

$$q_{\tau,i} = a_\tau u_{\tau,i}, \qquad \frac{1}{k}\sum_{j=0}^{k-1} u_{\tau,j} = 1. \tag{29}$$

Then $b_\tau = a_\tau$ and $\rho_{\tau,i} = u_{\tau,i}$, so calibration expresses each later segment relative to the initial ratio level.

The two normalization stages therefore serve distinct purposes. Segment ratios are invariant to common local NLL scaling, while prefix calibration removes a common trajectory-wide proportional offset. The initial segments supply the temporal reference, and stop-gradient blocks the direct auxiliary gradient through the anchor. Prefix tokens remain supervised by the base OPD objective.