JUNCTIONS AND BASINS: TRAJECTORY-SENSITIVE ON-POLICY DISTILLATION FOR LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) is an effective paradigm for transferring knowledge between language models, where a student learns from dense token-level supervision along its own trajectories. However, existing OPD methods typically apply uniform supervision across token positions, overlooking the substantial variation in their downstream consequences. We find that student trajectories exhibit two characteristic regimes: *Junctions*, where local variations strongly influence subsequent generation, and *Basins*, where plausible alternatives induce only limited changes to the future trajectory. To identify these regimes, we introduce a lightweight gradient-based future-sensitivity estimator that measures the directional sensitivity of the sampled future trajectory at each token position, together with a two-component mixture model that adaptively captures low- and high-sensitivity regions. Building on this structure, we propose Junction–Basin On-Policy Distillation (JB-OPD), which applies mode-seeking distillation at Junction-like positions while using mode-covering supervision with relational preservation at Basin-like positions. Across six mathematical reasoning benchmarks, JB-OPD consistently outperforms the state-of-the-art OPD baseline across Qwen3-0.6B, 1.7B, and 4B students, with average Pass@8 gains of 3.65%, 4.46%, and 4.39%, respectively. Further analysis reveals clear heterogeneity in downstream influence and shows that adapting distillation according to the inferred Junction–Basin roles consistently improves reasoning performance, supporting future trajectory sensitivity as an effective signal for allocating on-policy supervision.

1 INTRODUCTION

Knowledge distillation (Hinton et al., 2015; Gou et al., 2021) is a widely used approach for transferring the capabilities of large language models (LLMs) to smaller and more efficient students. Conventional distillation typically trains the student on fixed teacher- or expert-generated trajectories (Yang et al., 2026a; Zhang et al., 2026). For autoregressive generation, however, the student conditions on prefixes induced by its own previous predictions at inference time. Deviations from the reference trajectories seen during training can therefore move the model into poorly supervised states, where local errors may accumulate over subsequent generation (Mroueh et al., 2026).

On-policy distillation (OPD) alleviates this train–inference state mismatch by sampling trajectories from the student and querying the teacher on the states it actually visits (Li et al., 2026a; Zhao et al., 2026; Agarwal et al., 2024). In this way, OPD combines on-policy state coverage with dense token-level teacher feedback, providing fine-grained supervision while reducing the discrepancy between training and inference. Recent OPD methods have mainly focused on how the student should imitate the teacher at each visited state, for example through reverse-KL objectives or adaptive strategies that modify supervision according to properties of the current predictive distribution (Li et al., 2026b; Jin et al., 2026). However, these approaches primarily characterize each position through its local predictive behavior, without explicitly considering its influence on the subsequent reasoning trajectory (Sun et al., 2026; Jang et al., 2026; Jin et al., 2026).

In this paper, we examine on-policy distillation from the perspective of token-wise downstream influence along student reasoning trajectories. Our analysis reveals substantial heterogeneity across positions: local token variations can have markedly different effects on subsequent generation. For example, replacing a connective such as ‘therefore” with ‘thus” may leave the subsequent derivation

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.

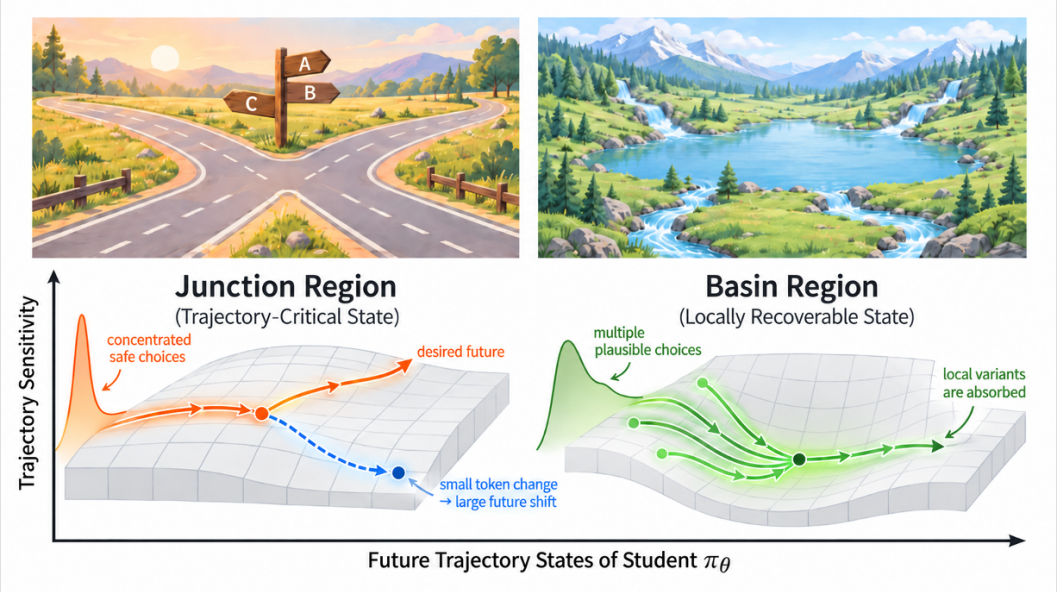


Figure 1: **Junction and Basin regions in student on-policy trajectories.** Junctions denote trajectory-critical positions with high downstream influence, where local token variations can substantially affect subsequent reasoning. Basins denote locally tolerant regions with lower downstream influence, where plausible local alternatives have limited impact on the future trajectory. This distinction motivates position-adaptive supervision according to the downstream role of each token.

nearly unchanged, while modifying an operator, sign, variable substitution, or intermediate quantity can propagate through multiple later steps and redirect the reasoning process. This difference is not simply a matter of local confidence or uncertainty, but rather of how strongly the future trajectory responds to a local token variation.

Based on this observation, we introduce a *Junction–Basin* view of student on-policy trajectories, as illustrated in Fig. 1. A **Junction** denotes a trajectory-critical position with high downstream influence, where a local variation can strongly influence subsequent generation. A **Basin**, in contrast, denotes a locally tolerant position with lower downstream influence, where different plausible alternatives can be accommodated with limited effect on the future trajectory. Junction-like and Basin-like positions can be interleaved throughout a reasoning trajectory, revealing fine-grained differences in the downstream roles of individual token positions. This suggests that applying the same form of supervision uniformly across positions may overlook meaningful variation in their influence on future reasoning.

Motivated by this view, we propose **Junction–Basin On-Policy Distillation (JB-OPD)**, which adapts token-level supervision according to the downstream sensitivity of each position. A key challenge is to estimate such sensitivity efficiently along long reasoning trajectories. Explicitly replacing tokens and independently evaluating their alternative continuations at every position would incur substantial computational overhead. To address this, we develop a lightweight gradient-based future-sensitivity estimator that measures how the likelihood of the sampled future trajectory responds to a local movement toward a plausible alternative-token direction. In this way, token-level future-sensitivity scores can be obtained directly from the sampled on-policy trajectory without separately rolling out alternative continuations for every position. We then model the resulting future-sensitivity score distribution with a two-component mixture model and use the posterior probability as a continuous estimate of the Junction–Basin role at each position. This probabilistic characterization naturally motivates asymmetric supervision. At Junction-like positions, where local deviations are more likely to propagate into larger trajectory-level changes, we applies mode-seeking supervision to keep the student closely aligned with dominant teacher modes. At Basin-like positions, where multiple plausible local alternatives may lead to similar future trajectories, we instead favor mode-covering distribution matching together with lightweight relational preservation, avoiding unnecessary over-concentration on a single local choice. The mixture posterior smoothly interpolates

between these two forms of supervision, allowing future trajectory sensitivity to directly modulate the form and strength of token-level distillation without requiring a hard Junction–Basin assignment.

In summary, our contributions are as follows: (1) we reveal substantial token-wise heterogeneity in the downstream sensitivity of student on-policy reasoning trajectories and introduce the *Junction–Basin* view to characterize this asymmetry; (2) we develop a lightweight gradient-based future-sensitivity estimator together with a probabilistic mixture model to derive continuous Junction–Basin posteriors; (3) we propose JB-OPD, which adaptively combines mode-seeking supervision at Junction-like positions with mode-covering, relation-preserving supervision at Basin-like positions; and (4) extensive experiments across six mathematical reasoning benchmarks and multiple model scales demonstrate consistent improvements over existing OPD methods.

2 Related Work

2.1 Knowledge Distillation.

Knowledge distillation (KD) aims to transfer the capabilities of a large teacher model into a more compact student model, typically through distribution matching, representation alignment, or sequence-level imitation (Hinton et al., 2015; Kim & Rush, 2016). Beyond these basic formulations, recent studies have explored alternative discrepancy measures, selective distillation, contrastive objectives, and importance-aware weighting over training examples or token positions (Ko et al., 2024; 2025; He et al., 2025; Guo et al., 2026). Such techniques have substantially expanded the flexibility of KD for compressing large language models.

However, most conventional KD procedures optimize the student on externally provided or teacher-induced contexts rather than on states generated by the student itself. During autoregressive inference, the student must condition on its own previous outputs, causing the training and inference state distributions to diverge and potentially amplifying errors over long generations (Bengio et al., 2015; Ranzato et al., 2015).

2.2 On-Policy Distillation

On-policy distillation (OPD) reduces this state-distribution mismatch by collecting trajectories from the current student and evaluating those visited prefixes with a teacher model (Agarwal et al., 2024; Li et al., 2026a). In this way, teacher feedback is delivered directly on the states induced by the student policy, providing dense supervision throughout the generated trajectory. This training paradigm has recently become increasingly important for post-training reasoning models and has also been adopted in large-scale reasoning systems and self-distillation settings (Yang et al., 2025; Xu et al., 2026a; Zhao et al., 2026; Ye et al., 2026). Recent research has improved OPD from several perspectives. Some studies examine whether the teacher and student exhibit compatible reasoning behavior, and investigate teacher selection or complementary supervision across models (Fu et al., 2026; Li et al., 2026a; Jiang et al., 2026). Others adapt the supervision signal according to uncertainty, teacher–student disagreement, teachability, or token-level reliability (Jin et al., 2026; Xu et al., 2026b; Wang et al., 2026). Another line of work focuses on reducing the computational burden of distillation through representation-level alignment, cached or offline teacher feedback, and reasoning-efficient training strategies (Yang et al., 2026b; Wu et al., 2026; Ziheng et al., 2026).

Nevertheless, most existing OPD methods decide how to supervise a position mainly from information available at or near the current prediction step. They therefore provide only limited modeling of how a local token variation propagates into later reasoning states. In contrast, our work explicitly measures downstream future influence along student trajectories and uses this signal to adapt token-level distillation.

3 METHODOLOGY

3.1 Preliminaries

Let π_θ denote the student policy and π_T a fixed teacher policy. Given an input prompt x, the student generates an on-policy response $y = (y_1, \ldots, y_L) \sim \pi_\theta(\cdot \mid x)$, where L denotes the response

length. At generation step t, we denote the student-visited prefix by $s_t = (x, y_{<t})$. The next-token distributions of the student and teacher are respectively denoted by $p_t(a) = \pi_\theta(a \mid s_t)$ and $q_t(a) = \pi_T(a \mid s_t)$ for $a \in \mathcal{V}$.

Standard on-policy distillation (Agarwal et al., 2024) supervises the student directly on states visited by its own policy. Following the commonly used reverse-KL formulation, we define the token-level OPD objective as:

$$\mathcal{L}_t^{\mathrm{OPD}} = D_{\mathrm{KL}}(p_t \| q_t) = \sum_{a \in \mathcal{V}} p_t(a) \log \frac{p_t(a)}{q_t(a)}. \tag{1}$$

The overall objective is:

$$\mathcal{L}_{\mathrm{OPD}} = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[\sum_{t=1}^{L} \mathcal{L}_t^{\mathrm{OPD}} \right]. \tag{2}$$

Reverse KL tends to concentrate the student distribution around the teacher's dominant modes, providing mode-seeking supervision at each visited state (Minka et al., 2005). In practice, this objective can be approximated with a single-sample Monte Carlo estimate on student-generated trajectories and optimized with a policy-gradient-style update. The teacher–student log-probability difference of each sampled token provides the token-level learning signal, while importance weighting and PPO-style clipping can be used to stabilize training (Jin et al., 2026). However, applying the same objective uniformly across a trajectory overlooks substantial differences in downstream influence between token positions. Local variation at some positions can strongly affect subsequent reasoning, whereas other positions remain relatively tolerant to plausible alternatives. Motivated by this asymmetry, we propose **Junction–Basin On-Policy Distillation (JB-OPD)**, which estimates token-wise future sensitivity, probabilistically models the resulting sensitivity regimes, and adapts standard OPD supervision accordingly.

3.2 Gradient-Based Future Sensitivity Estimation

The first challenge is to quantify how strongly each token affects the subsequent student trajectory. One possible approach is to replace a sampled token with an alternative token and recompute the continuation. Applying this procedure to every position, however, requires repeated forward passes and introduces substantial overhead during on-policy training. We instead propose a gradient-based estimator of *local future sensitivity*. Under causal autoregressive modeling, the gradient of the trajectory objective with respect to a token embedding captures how changes at that position affect subsequent predictions. This property allows us to estimate the future sensitivity of all positions jointly with a single backward pass. For each sampled token y_t, we first select the highest-probability alternative token under the student distribution:

$$y_t' = \arg \max_{a \in \mathcal{V},\, a \neq y_t} p_t(a). \tag{3}$$

The alternative y_t' is used to specify a token-dependent direction in the student's embedding space. Let $E_\theta : \mathcal{V} \to \mathbb{R}^d$ denote the student input embedding function, where d is the embedding dimension, and let $e_t = E_\theta(y_t)$. We define the normalized direction from y_t toward y_t' as

$$v_t = \frac{E_\theta(y_t') - E_\theta(y_t)}{\|E_\theta(y_t') - E_\theta(y_t)\|_2 + \varepsilon_v}, \tag{4}$$

where $\varepsilon_v > 0$ is a small constant for numerical stability. Normalization reduces the effect of varying embedding distances across token pairs, so that v_t primarily captures the direction toward an alternative that is likely under the student distribution.

We next measure how the subsequent continuation depends on the token representation at each position. Let $\ell_k = -\log p_k(y_k)$ denote the negative log-likelihood of the sampled token at position k. We define the trajectory objective as: $\mathcal{J}_{\mathrm{traj}} = \sum_{k=1}^{L} \ell_k$. Although $\mathcal{J}_{\mathrm{traj}}$ is evaluated over the complete response, causal masking ensures that e_t can affect only predictions after position t. Therefore, its gradient with respect to e_t can be written as:

$$g_t = \frac{\partial \mathcal{J}_{\mathrm{traj}}}{\partial e_t} = \sum_{k=t+1}^{L} \frac{\partial \ell_k}{\partial e_t}, \qquad g_t \in \mathbb{R}^d. \tag{5}$$

Thus, g_t aggregates the local dependence of the subsequent continuation on the representation at position t. Importantly, the gradients $\{g_t\}_{t=1}^{L-1}$ for all positions are obtained simultaneously by backpropagating the same trajectory objective. The gradient g_t describes local sensitivity in the embedding space, but its magnitude alone does not indicate whether this sensitivity is aligned with a likely alternative token. We therefore project g_t onto the token-dependent direction v_t and define the future-sensitivity score as:

$$C_t = \left| g_t^\top v_t \right|, \qquad t = 1, \ldots, L-1. \tag{6}$$

The absolute value retains the magnitude of the response regardless of whether moving along v_t locally increases or decreases the trajectory objective. The score C_t has a direct interpretation as a directional derivative. Consider the local path $e_t(\alpha) = e_t + \alpha v_t$, where α is a scalar and $\alpha = 0$ corresponds to the original token representation. By the chain rule:

$$C_t = \left| \left. \frac{\partial \mathcal{J}_{\mathrm{traj}}(e_t + \alpha v_t)}{\partial \alpha} \right|_{\alpha=0} \right| = \left| g_t^\top v_t \right|. \tag{7}$$

Hence, C_t measures how rapidly the trajectory objective changes locally when the representation at position t moves in the embedding direction toward its highest-probability alternative token. All token-wise scores $\{C_t\}_{t=1}^{L-1}$ can be obtained jointly from a single backward pass and are detached before the subsequent probabilistic Junction–Basin modeling stage.

3.3 Probabilistic Junction–Basin Modeling

The future-sensitivity score C_t provides a token-level measure of how strongly each position affects the subsequent student trajectory. We therefore use this score to characterize whether a position exhibits more Junction-like or Basin-like behavior. However, directly separating the two regimes with a fixed threshold is undesirable because the scale and distribution of C_t may vary across models, rollout batches, and training stages. We instead model the score distribution probabilistically and infer a continuous Junction–Basin role for each position. Since C_t is non-negative and typically skewed, we first transform it as $u_t = \log(C_t + \varepsilon_s)$, where $\varepsilon_s > 0$ is a small constant for numerical stability. Let $c_t \in \mathcal{C} = \{\mathrm{B}, \mathrm{J}\}$ denote the latent role of position t, corresponding to Basin and Junction, respectively. We fit a two-component Gaussian mixture model:

$$p_\Gamma(u_t) = \sum_{c \in \mathcal{C}} \pi_c \mathcal{N}(u_t; \mu_c, \sigma_c^2), \tag{8}$$

where $\Gamma = \{\mu_c, \sigma_c^2, \pi_c\}_{c \in \mathcal{C}}$ contains the component mean, variance, and mixture weight. The parameters are estimated using Expectation–Maximization (Feng & Ge, 2026), and the components are identified according to their means. Rather than making a hard assignment, we retain the posterior probability of the Junction component:

$$\zeta_t = p_\Gamma(c_t = \mathrm{J} \mid u_t) = \frac{\pi_{\mathrm{J}} \mathcal{N}(u_t; \mu_{\mathrm{J}}, \sigma_{\mathrm{J}}^2)}{\sum_{c \in \mathcal{C}} \pi_c \mathcal{N}(u_t; \mu_c, \sigma_c^2)}. \tag{9}$$

The resulting $\zeta_t \in [0, 1]$ provides a continuous Junction–Basin weight: larger values indicate higher future sensitivity, while smaller values indicate greater local tolerance.

3.4 Adaptive Junction–Basin Distillation

The posterior ζ_t characterizes how strongly position t exhibits the higher-sensitivity Junction-like regime. We use this signal to adapt the token-level supervision: positions with larger ζ_t retain more of the mode-seeking OPD objective, whereas positions with smaller ζ_t receive broader teacher supervision that preserves plausible alternatives. For Junction-like positions, we retain the standard OPD objective $\mathcal{L}_t^{\mathrm{OPD}}$ defined in Eq. (2). Since this objective uses reverse KL, it tends to concentrate the student distribution around the teacher's dominant modes (Minka et al., 2005), which is desirable at high-sensitivity positions where local variation can more strongly affect the subsequent trajectory. For Basin-like positions, we instead reverse the KL direction and use forward-KL supervision:

$$\mathcal{L}_t^{\mathrm{FKL}} = D_{\mathrm{KL}}(q_t \| p_t) = \sum_{a \in \mathcal{V}} q_t(a) \log \frac{q_t(a)}{p_t(a)}. \tag{10}$$

Compared with reverse KL, forward KL encourages broader coverage of the teacher distribution and is therefore better suited to locally tolerant regions where multiple teacher-supported alternatives may remain plausible (Minka et al., 2005). In practice, we approximate this term using the teacher's top-K support for computational efficiency.

We further preserve the teacher's relative preference among these alternatives. Let $\mathcal{S}_t^K = \mathrm{TopK}(q_t)$ and $\mathcal{P}_t = \{(i,j) : i,j \in \mathcal{S}_t^K,\ i \neq j\}$. We define:

$$\mathcal{L}_t^{\mathrm{rel}} = \frac{1}{|\mathcal{P}_t|} \sum_{(i,j)\in\mathcal{P}_t} \left[\log \frac{p_t(i)}{p_t(j)} - \log \frac{q_t(i)}{q_t(j)} \right]^2. \tag{11}$$

This term complements forward-KL matching by preserving the teacher's local preference structure within the selected support. We therefore define the Basin supervision as:

$$\mathcal{L}_t^{\mathrm{B}} = \mathcal{L}_t^{\mathrm{FKL}} + \lambda_{\mathrm{R}} \mathcal{L}_t^{\mathrm{rel}}, \tag{12}$$

where λ_{R} controls the strength of relational preservation. Rather than assigning each position to a discrete regime, JB-OPD interpolates between standard OPD supervision and Basin supervision using ζ_t:

$$\mathcal{L}_t^{\mathrm{JB}} = \zeta_t \mathcal{L}_t^{\mathrm{OPD}} + (1-\zeta_t)\mathcal{L}_t^{\mathrm{B}}. \tag{13}$$

Thus, larger ζ_t places greater weight on the original mode-seeking OPD objective, whereas smaller ζ_t shifts supervision toward the mode-covering Basin objective.

Since the representation of the final response token cannot affect any subsequent prediction, no future-sensitivity score is defined for $t = L$. We therefore retain standard OPD supervision at this position, equivalently setting $\zeta_L = 1$. The overall training objective is:

$$\mathcal{L}_{\text{JB-OPD}} = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{L} \mathcal{L}_t^{\mathrm{JB}} \right]. \tag{14}$$

JB-OPD therefore generalizes standard OPD by adapting token-level supervision according to future sensitivity, without requiring additional future rollouts or alternative continuations.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and training data. We evaluate JB-OPD using Qwen3-8B (Yang et al., 2025) as the teacher model and Qwen3-0.6B-Base, Qwen3-1.7B-Base, and Qwen3-4B-Base as student models. Additional results on other model families are provided in the Appendix. We use MATH (Hendrycks et al., 2021) dataset for training the 0.6B and 1.7B students, and DAPO (Yu et al., 2026) dataset for training the 4B student. Unless otherwise specified, all competing methods use the same training prompts, rollout budget, optimizer, and number of optimization steps to ensure a controlled comparison.

Evaluation benchmarks and metrics. We evaluate mathematical reasoning performance on six benchmarks: MATH500 (Hendrycks et al., 2021), AMC23 (Zheng et al., 2026), AIME24 (Dai et al., 2026), AIME25 (Dai et al., 2026), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). During evaluation, we sample eight responses for each problem using temperature 1.0, top-p $= 0.8$, and a maximum generation length of 8192 tokens. We report Avg@8 and Pass@8. Avg@8 denotes the average correctness across the eight sampled responses, whereas Pass@8 measures whether at least one of the eight responses reaches the correct final answer.

Implementation details. We implement off-policy knowledge distillation with DistillKit (Jin et al., 2026), while both on-policy distillation and GRPO are built on the `verl` framework (Sheng et al., 2024). Unless otherwise specified, training uses a global batch size of 128 and a mini-batch size of 32, corresponding to four optimization updates for each collected batch. For Basin supervision, the teacher distribution is restricted to its top-K support with $K = 16$, and the relational regularization weight is fixed to $\lambda_{\mathrm{R}} = 0.1$ for all experiments. For numerical stability, we set $\varepsilon_v = 10^{-8}$ in the directional normalization and $\varepsilon_s = 10^{-8}$ before the logarithmic transformation of the sensitivity scores. Additional implementation details are provided in the Appendix. All experiments are run on four NVIDIA A100 80GB GPUs.

Table 1: Main results (accuracy %) on six mathematical reasoning benchmarks. Results are reported with Avg@8 and Pass@8 for three student scales. **Bold** denotes the best performance and underline denotes the second-best performance under each setting.

Student	Method	MATH500		AMC23		Minerva		OlympiadBench		AIME24		AIME25	
		Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8
Qwen3-0.6B-Base (MATH)	KD	47.80	69.60	23.43	52.50	18.61	36.03	18.15	36.59	2.19	6.67	0.83	6.67
	GRPO	51.83	74.40	25.25	55.00	20.08	35.66	18.91	34.96	4.58	10.00	0.83	10.00
	OPD	50.09	73.20	24.69	57.50	19.98	34.93	20.15	37.19	2.50	10.00	1.25	6.67
	EOPD	52.02	76.00	27.81	55.00	20.82	**37.13**	19.52	39.56	4.17	13.33	1.25	6.67
	JB-OPD	**53.35**	**79.20**	**29.15**	**61.00**	**22.05**	36.90	**21.45**	**42.50**	**5.83**	**16.67**	**1.67**	**13.33**
Qwen3-1.7B-Base (MATH)	KD	63.80	84.20	38.11	70.00	28.14	44.10	27.66	48.40	10.06	20.00	3.36	16.67
	GRPO	68.83	84.60	40.62	70.00	29.46	48.16	29.89	50.81	9.17	20.00	4.58	16.67
	OPD	67.76	84.80	39.06	70.00	29.83	47.06	30.09	51.56	8.33	20.00	6.25	16.67
	EOPD	68.73	87.60	41.88	75.00	30.15	50.74	**30.28**	51.11	10.42	23.33	5.83	16.67
	JB-OPD	**70.25**	**90.40**	**43.75**	**80.00**	**31.45**	**53.50**	30.15	**54.00**	**11.67**	**30.00**	**7.50**	**23.33**
Qwen3-4B-Base (DAPO-Math-14k)	KD	74.73	92.20	48.13	85.00	34.65	55.15	37.33	62.52	12.50	30.00	12.08	23.33
	GRPO	80.47	91.00	58.44	75.00	**40.81**	55.51	43.37	60.74	14.85	30.00	12.66	26.67
	OPD	78.81	90.80	57.33	80.00	40.08	54.00	42.10	58.80	18.33	26.67	12.08	30.00
	EOPD	80.20	93.00	60.94	87.50	39.71	56.99	43.24	63.11	17.92	36.67	**17.50**	33.33
	JB-OPD	**81.80**	**95.20**	**62.50**	**92.50**	40.65	**59.60**	**44.70**	**66.30**	**20.00**	**43.33**	17.30	**40.00**

4.2 Main Results

We compare JB-OPD with supervised knowledge distillation (KD) (Kim & Rush, 2016), GRPO (Shao et al., 2024), standard on-policy distillation (OPD) (Agarwal et al., 2024), and EOPD (Jin et al., 2026), a recent adaptive OPD method. Table 1 presents the results across six mathematical reasoning benchmarks and three student scales. JB-OPD consistently improves over standard OPD and generally outperforms EOPD across different model capacities. On Qwen3-1.7B, for example, JB-OPD improves Avg@8 and Pass@8 over EOPD by 1.52% and 2.80% on MATH500, respectively, and by 1.87% and 5.00% on AMC23. The advantage becomes more pronounced on the challenging AIME benchmarks, where Pass@8 increases by 6.67% on AIME24 and 6.66% on AIME25 compared with EOPD. Consistent gains are also observed at both smaller and larger student scales; notably, JB-OPD improves Pass@8 over EOPD on five of the six benchmarks for Qwen3-0.6B and on all six benchmarks for Qwen3-4B. Overall, these results indicate that downstream trajectory sensitivity provides a useful signal for adapting on-policy supervision beyond existing distribution-based adaptive strategies.

4.3 Analyzing the Junction–Basin Structure

Sensitivity distribution. The central motivation of JB-OPD is that token positions along student reasoning trajectories differ substantially in their downstream sensitivity. We therefore examine whether the proposed gradient-based future-sensitivity signal reveals a meaningful Junction–Basin structure. Figure 2 visualizes the empirical distribution of u_t on MATH500 and AMC23 together with the fitted two-component Gaussian mixture. The sensitivity scores exhibit clear heterogeneity, with distinct regions of relatively lower and higher future sensitivity. We associate the lower-mean component with Basin-like tokens and the higher-mean component with Junction-like tokens according to their relative sensitivity levels. Rather than introducing a manually specified threshold, the mixture posterior provides a continuous estimate of the Junction–Basin role at each position, allowing the supervision strength to vary smoothly with the inferred sensitivity. Overall, the observed distributions show that token positions along student reasoning trajectories can differ substantially in their downstream sensitivity, providing empirical motivation for the proposed role-adaptive distillation.

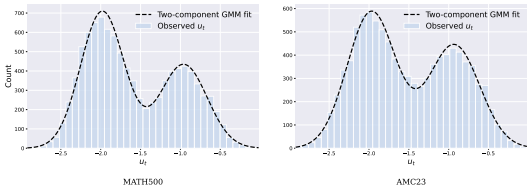


Figure 2: Illustration of the Junction–Basin sensitivity distribution.

Token-level trajectory visualization. Figure 3 visualizes the token-wise future sensitivity along representative mathematical reasoning trajectories. Higher sensitivity frequently appears around operators, intermediate quantities, variable substitutions, equation transformations, and transitions

between major reasoning steps, where local token variations can more strongly influence the subsequent trajectory. In contrast, connective expressions, formatting tokens, and locally interchangeable textual realizations tend to exhibit lower future sensitivity, indicating greater local tolerance. The visualization further shows that high- and low-sensitivity positions are interleaved throughout a response rather than forming long contiguous regions, revealing substantial fine-grained heterogeneity in downstream influence within a single reasoning trajectory.

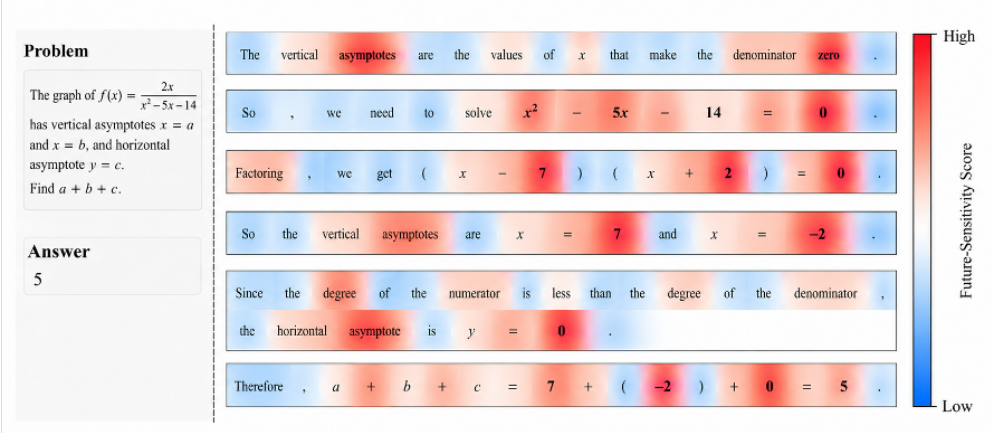


Figure 3: Token-level visualization of the future-sensitivity score along representative mathematical reasoning trajectories.

4.4 Ablation Studies

Effect of Role-Specific Distillation Strategy. We compare different RKL/FKL assignments for Junction-like and Basin-like positions to examine whether the two regions require different forms of supervision. As shown in Table 2, applying RKL to Junctions and FKL to Basins consistently gives the strongest overall performance across the four benchmarks. In contrast, using the same KL direction for both regions is less effective, while reversing the proposed assignment causes a clear degradation. This suggests that high-sensitivity positions benefit from mode-seeking supervision, whereas low-sensitivity regions are better served by preserving a broader set of teacher-supported alternatives.

Table 2: Effect of different distillation strategies for Junction-like and Basin-like positions using Qwen3-1.7B-Base. Relational regularization is disabled to isolate the effect of the KL direction.

Junction	Basin	MATH500		AMC23		AIME24		AIME25	
		Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8
RKL	RKL	67.76	84.80	39.06	70.00	8.33	20.00	6.25	16.67
FKL	FKL	67.20	85.00	38.44	67.50	9.17	23.33	5.00	13.33
FKL	RKL	66.91	83.80	37.88	67.50	8.75	20.00	5.42	13.33
RKL	FKL	**69.68**	**88.80**	**42.81**	**77.50**	**10.83**	**26.67**	**7.08**	**20.00**

Table 3: Comparison of uniform, hard, and soft Junction–Basin routing using Qwen3-1.7B-Base.

Routing Strategy	MATH500		AMC23		AIME24		AIME25	
	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8
Uniform OPD	67.76	84.80	39.06	70.00	8.33	20.00	6.25	16.67
Hard GMM Assignment	69.31	88.20	42.38	77.50	10.83	26.67	7.08	20.00
Soft GMM Posterior	**70.25**	**90.40**	**43.75**	**80.00**	**11.67**	**30.00**	**7.50**	**23.33**

Table 4: Effect of relational preservation in Basin supervision using Qwen3-1.7B-Base.

Variant	MATH500		AMC23		AIME24		AIME25	
	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8
w/o Relational Regularization	69.18	88.60	41.88	77.50	10.42	26.67	6.67	20.00
JB-OPD	**70.25**	**90.40**	**43.75**	**80.00**	**11.67**	**30.00**	**7.50**	**23.33**

Effect of Soft Junction–Basin Routing. We evaluate whether the posterior uncertainty from the two-component mixture should be retained during optimization. Table 3 shows that hard GMM assignment already improves over uniform OPD, confirming the benefit of separating high- and low-sensitivity positions. Replacing the hard labels with the posterior Junction probability further improves performance across all four benchmarks, with larger gains on AIME24 and AIME25. Soft routing allows positions near the mixture boundary to interpolate between Junction and Basin supervision, avoiding the abrupt objective changes introduced by binary assignment.

Effect of Relational Preservation. We examine whether forward-KL matching alone is sufficient for Basin supervision. As shown in Table 4, adding the relational regularizer consistently improves performance over FKL alone, while the gain is smaller than that from the role-specific KL assignment. This indicates that forward KL provides the primary signal for preserving teacher-supported alternatives, whereas relational preservation further maintains their relative preference structure and provides complementary supervision.

Computational Overhead and Efficiency Analysis. Compared with standard OPD, JB-OPD requires neither additional teacher forward passes nor extra autoregressive rollouts. Its additional training cost mainly comes from future-sensitivity estimation and the forward-KL and relational terms used for Basin supervision. The future-sensitivity scores for all token positions are obtained jointly from the factual on-policy trajectory through a single trajectory-level backward pass, followed by lightweight directional projections and GMM fitting. Overall, JB-OPD adds 10.6 seconds per training step over the 47.8-second OPD baseline, corresponding to a 22.2% overhead, of which 1.4 seconds is attributable to future-sensitivity estimation and 9.2 seconds to the additional distillation terms. Since student on-policy generation itself takes approximately 37.7 seconds per step, rollout generation remains the dominant training cost. In terms of memory, future-sensitivity estimation and Basin supervision introduce approximately 360 MB and 1506.5 MB of additional memory per microbatch, respectively, for a total of about 1.87 GB, with an additional 0.6 GB bf16 token-embedding cache used to construct alternative-token directions. As the sensitivity estimator and Junction–Basin weighting are used only during training and do not alter the student architecture, JB-OPD introduces no additional model parameters or inference-time overhead.

5 Conclusion

In this work, we identify a downstream-sensitivity asymmetry in student on-policy trajectories, where different token positions can play markedly different roles in shaping subsequent generation. We capture this behavior with a Junction–Basin view that separates trajectory-critical positions (Junctions) from locally tolerant regions (Basins). Junctions favor mode-seeking supervision to constrain consequential deviations, whereas Basins benefit from mode-covering supervision that preserves multiple teacher-supported alternatives. Based on this insight, we develop JB-OPD, a lightweight on-policy distillation framework that estimates token-level future sensitivity and adaptively combines the two supervision behaviors through a probabilistic Junction–Basin assignment. Empirical results across multiple student scales and mathematical reasoning benchmarks show that JB-OPD consistently improves over existing distillation baselines, with particularly clear gains in Pass@8.

Ethics Statement

This study uses publicly available mathematical reasoning datasets and pretrained language models released for research use. No private, personally identifiable, clinical, or human-subject data are

involved, and therefore no additional ethical approval or informed consent is required. We follow the licenses and intended use of all datasets and models employed in this work. The distilled models may inherit limitations or biases from their teacher models and training data, which should be considered when applying them outside the evaluated reasoning tasks.

Reproducibility Statement

The proposed JB-OPD framework is described in detail in the main paper, including the future-sensitivity estimator, Junction–Basin modeling, and role-specific distillation objectives. Model configurations, training data, evaluation benchmarks, decoding settings, and implementation details are provided in the experimental section and appendix. Additional ablations and efficiency analyses are included to facilitate reproduction and verification of the method.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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.

Ibtihel Amara, Nazanin Sepahvand, Brett H Meyer, Warren J Gross, and James J Clark. Bd-kd: Balancing the divergences for online knowledge distillation. *arXiv preprint arXiv:2212.12965*, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Runpeng Dai, Linfeng Song, Haolin Liu, Zhenwen Liang, Dian Yu, Haitao Mi, Zhaopeng Tu, Rui Liu, Tong Zheng, Hongtu Zhu, et al. Cde: Curiosity-driven exploration for efficient reinforcement learning in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 59123–59144, 2026.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Wei Feng and Zongyuan Ge. Generalized category discovery under domain shift: A frequency domain perspective. *Advances in Neural Information Processing Systems*, 38:111721–111749, 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. *arXiv preprint arXiv:2603.25562*, 2026.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

Yiju Guo, Wenkai Yang, Zexu Sun, Ning Ding, Zhiyuan Liu, and Yankai Lin. Learning to focus: Causal attention distillation via gradient-guided token pruning. *Advances in Neural Information Processing Systems*, 38:24921–24948, 2026.

Changyi He, Yifu Ding, Jinyang Guo, Ruihao Gong, Haotong Qin, and Xianglong Liu. Da-kd: Difficulty-aware knowledge distillation for efficient large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen 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. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, 2026.

Yuxuan Jiang, Runchao Li, Shubhashis Roy Dipta, Dawei Li, and Zhao Yang. Cornerstones or stumbling blocks? deciphering the rock tokens in on-policy distillation. *arXiv preprint arXiv:2605.09253*, 2026.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. Todi: Token-wise distillation via fine-grained divergence control. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8089–8102, 2025.

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.

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

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. Distillm-2: A contrastive approach boosts the distillation of llms. *arXiv preprint arXiv:2503.07067*, 2025.

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. In *Advances in neural information processing systems*, 2022.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026a.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint arXiv:2606.02684*, 2026b.

Tom Minka et al. Divergence measures and message passing. 2005.

Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jarret Ross, and Jesus Rios. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. In *International Conference on Learning Representations*, volume 2026, pp. 110991–111007, 2026.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

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

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

Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Yilin Cheng, Bichuan Feng, Pengfei Liu, Junfeng Fang, and Xiang Wang. Simct: Recovering lost supervision for cross-tokenizer on-policy distillation. *arXiv preprint arXiv:2605.07711*, 2026.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback-leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5737–5755, 2025.

Yecheng Wu, Song Han, and Hai Cai. Lightning opd: Efficient post-training for large reasoning models with offline on-policy distillation. *arXiv preprint arXiv:2604.13010*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026b.

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.

Matthew Yang, Hao Bai, Ian Wu, Gene Yang, Amrith Setlur, and Aviral Kumar. Int: Self-proposed interventions enable credit assignment in llm reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 85054–85091, 2026a.

Shenzhi Yang, Guangcheng Zhu, Bowen Song, Haobo Wang, Mingxuan Xia, Xing Zheng, Yingfan Ma, Zhongqi Chen, Weiqiang Wang, Junbo Zhao, et al. Oprd: On-policy representation distillation. *arXiv preprint arXiv:2606.06021*, 2026b.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. In *International Conference on Learning Representations*, volume 2026, pp. 120693–120726, 2026.

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. *arXiv preprint arXiv:2601.18734*, 2026.

Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, He Xing, Runpeng Dai, Rui Liu, Huiwen Bao, Chengsong Huang, Heng Huang, et al. Parallel-r1: Towards parallel thinking via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 121144–121166, 2026.

Zhou Ziheng, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation. *arXiv preprint arXiv:2605.27028*, 2026.

A Additional Implementation Details

For the supervised KD baseline, we generate one teacher response for each training prompt and optimize the student on the resulting fixed teacher contexts. OPD, EOPD, JB-OPD, and GRPO are trained under the same on-policy framework, where fresh student trajectories are collected during training. We use a training batch size of 128 and a mini-batch size of 32 for all on-policy methods. OPD, EOPD, and JB-OPD sample one rollout per prompt, whereas GRPO uses eight rollouts for group-relative optimization. For adaptive distillation, OPD, EOPD and JB-OPD use the teacher's top-K support with $K = 16$, while JB-OPD additionally sets the relational regularization weight to $\lambda_{\mathrm{R}} = 0.1$. The complete training configuration is summarized in Table 5.

Table 5: Training configurations for off-policy KD and on-policy methods. Method-specific settings are indicated explicitly.

Training Regime	**Hyperparameter**	**Setting**
Off-policy KD	Learning rate	1×10^{-5}
	LR scheduler	Cosine
	Optimizer	AdamW
	CE loss weight	0.5
	KL loss weight	0.5
	Training batch size	128
	Training epochs	3
	Cutoff length / Top-K	4096 / 16
OPD / EOPD / JB-OPD	Learning rate	3×10^{-6}
	LR scheduler	Cosine
	Optimizer	AdamW
	Training batch / mini-batch	128 / 32
	Samples per prompt	1
	Top-p	1.0 (Qwen), 0.8 (Llama)
	Maximum response length	4096
	Training temperature	1.0
	Training epochs	3 (MATH), 2 (DAPO)
GRPO	Learning rate	3×10^{-6}
	LR scheduler	Cosine
	Optimizer	AdamW
	Training batch / mini-batch	128 / 32
	Samples per prompt	8
	Top-p	1.0 (Qwen), 0.8 (Llama)
	Maximum response length / temperature	4096 / 1.0
	Training epochs	3 (MATH), 2 (DAPO)

A.1 Chat Templates

Prompt formatting can materially affect the form of student-generated reasoning trajectories. To avoid confounding the comparison with differences in conversational formatting, we keep the template fixed within each training paradigm. For knowledge-distillation-based methods, we use the same teacher-compatible format across KD, OPD, EOPD, and JB-OPD:

```
<|im_start|>user
{query}
<|im_end|>
<|im_start|>assistant
<think>

</think>
```

For GRPO, we follow the default Qwen3-Base format:

```
<|im_start|>user
{query}
<|im_end|>
<|im_start|>assistant
```

At evaluation time, all methods receive the same problem statement and answer instruction. We do not introduce method-specific prompts during evaluation, ensuring that the reported differences reflect the learned policy rather than prompt engineering.

A.2 Evaluation Protocol.

We use a unified decoding setup for all generation-based evaluations, with temperature 1.0, top-$p = 0.8$, and a maximum generation budget of 8192 tokens. For mathematical reasoning and GPQA-Diamond (Rein et al., 2023), model outputs are produced with the Qwen2.5-Math evaluation toolkit (Yang et al., 2025). Mathematical reasoning tasks are evaluated in a zero-shot setting, and each problem is followed by an instruction requiring step-by-step reasoning and a final answer enclosed in \boxed{}. We report Avg@8 and Pass@8 for these generation-based benchmarks.

MMLU-Pro (Wang et al., 2024) is evaluated separately under its standard 5-shot configuration, using demonstrations sampled from the validation set. The model is prompted to reason before returning the final option in the form "the answer is (X)." For instruction-following evaluation, we use AlpacaEval 2.0 (Dubois et al., 2024) with its official pairwise comparison protocol. Performance is measured by win rate (WR) and length-controlled win rate (LC-WR), using `weighted_alpaca_eval_gpt4_turbo` as the evaluator and GPT-4 Turbo (Achiam et al., 2023) as the reference response.

B Out-of-Domain Generalization

We further evaluate whether the benefits of JB-OPD extend beyond the mathematical reasoning distribution used for training. Specifically, we test the Qwen3-1.7B-Base student on GPQA-Diamond and MMLU-Pro for general reasoning, as well as AlpacaEval 2.0 for instruction following. As shown in Table 6, JB-OPD consistently achieves the best performance across all evaluated metrics. On GPQA-Diamond, JB-OPD improves Avg@8 and Pass@8 over EOPD by 1.62% and 3.04%, respectively, while MMLU-Pro Pass@1 increases by 1.66%. The gains also transfer to instruction-following evaluation: JB-OPD improves LC-WR over EOPD by 1.29% and exceeds the strongest baseline in WR by 1.16%. These results suggest that the benefits of future-sensitivity-aware supervision are not limited to the mathematical training distribution, but also transfer to unseen reasoning and instruction-following tasks.

C Generalization to the Llama Model Family

We further evaluate JB-OPD with a Llama teacher–student pair to study its behavior beyond the Qwen family used in the main experiments. Specifically, Llama-3.1-8B-Instruct serves as the teacher and Llama-3.2-3B-Instruct as the student, with all methods trained under the same protocol.

As reported in Table 7, JB-OPD maintains consistent improvements across all four benchmarks. Compared with EOPD, it gains 1.45%/2.00% in Avg@8/Pass@8 on MATH500 and 1.56%/2.50% on AMC23. On AIME24 and AIME25, the Pass@8 improvements further increase to 3.34% and 3.33%, respectively. These results show that the advantage of allocating supervision according to

Table 6: Out-of-domain evaluation of Qwen3-1.7B-Base on general reasoning and instruction-following benchmarks.

Method	GPQA-Diamond		MMLU-Pro	AlpacaEval 2.0	
	Avg@8	Pass@8	Pass@1	LC-WR	WR
KD	27.01	75.20	37.54	19.30	23.63
GRPO	27.86	77.83	41.46	16.86	20.55
OPD	30.08	77.78	42.26	22.86	<u>29.92</u>
EOPD	<u>31.50</u>	<u>81.31</u>	<u>43.20</u>	<u>23.83</u>	29.54
JB-OPD	**33.12**	**84.35**	**44.86**	**25.12**	**31.08**

downstream future sensitivity persists under a different model family, supporting the transferability of the proposed Junction–Basin formulation across teacher–student architectures.

Table 7: Results on mathematical reasoning benchmarks using the Llama model family. Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct are used as the student and teacher, respectively. **Bold** and <u>underline</u> denote the best and second-best results, respectively.

Method	MATH500		AMC23		AIME24		AIME25	
	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8
KD	42.60	65.80	18.43	<u>52.50</u>	1.66	<u>13.33</u>	1.25	6.67
GRPO	<u>45.01</u>	66.00	20.31	50.00	<u>2.91</u>	10.00	<u>2.08</u>	<u>10.00</u>
OPD	43.60	64.80	19.06	47.50	2.08	10.00	1.25	6.67
EOPD	44.90	<u>67.20</u>	<u>22.18</u>	<u>52.50</u>	<u>2.91</u>	<u>13.33</u>	<u>2.08</u>	<u>10.00</u>
JB-OPD	**46.35**	**69.20**	**23.74**	**55.00**	**3.75**	**16.67**	**2.92**	**13.33**

D Additional Ablations

D.1 Sensitivity to Top-K Selection

The top-K support determines how many teacher-supported alternatives are retained in the Basin objective. A small K may fail to preserve sufficient probability mass from the teacher distribution, while a larger K includes more low-probability tokens and increases the cost of forward-KL and relational matching. We therefore vary $K \in \{1, 8, 16, 128\}$ while keeping all other training settings fixed. Results for Qwen3-1.7B-Base on MATH500 and AMC23 are reported in Table 8.

JB-OPD exhibits relatively stable performance across different values of K, with the best overall results obtained at $K = 16$. When K is too small, performance decreases because the retained support does not sufficiently cover plausible teacher-supported alternatives in Basin-like regions. Increasing K beyond 16 does not provide further gains, despite retaining a substantially larger portion of the low-probability tail. This suggests that a moderate support size is sufficient to capture the useful structure of the teacher distribution while avoiding unnecessary computation. We therefore use $K = 16$ in all main experiments.

Table 8: Sensitivity of JB-OPD to the teacher top-K support on Qwen3-1.7B-Base. Bold and underline indicate the best and second-best results, respectively.

K	MATH500		AMC23	
	Avg@8	Pass@8	Avg@8	Pass@8
1	68.92	87.80	40.94	72.50
8	69.63	89.20	42.81	77.50
16	**70.25**	<u>90.40</u>	<u>43.75</u>	**80.00**
128	<u>70.18</u>	**90.60**	**43.88**	<u>79.50</u>

D.2 Sensitivity to the Relational Weight

We further study the sensitivity of JB-OPD to the relational regularization coefficient λ_{R}, which controls the contribution of preference preservation within the Basin objective. We vary $\lambda_{\mathrm{R}} \in \{0, 0.05, 0.1, 0.5, 1.0\}$ while keeping the Junction–Basin modeling procedure and all other training settings unchanged. Results are reported in Table 9.

JB-OPD remains stable across a broad range of λ_{R}, indicating that the method is not highly sensitive to the precise strength of the relational constraint. Introducing relational preservation consistently improves over the setting without this term, while the differences among moderate non-zero coefficients remain relatively small. The strongest overall performance is obtained at $\lambda_{\mathrm{R}} = 0.1$, suggesting that a modest relational constraint is sufficient to complement forward distribution matching in Basin-like regions. Accordingly, we use $\lambda_{\mathrm{R}} = 0.1$ for all main experiments.

Table 9: Sensitivity of JB-OPD to the relational regularization coefficient λ_{R} on Qwen3-1.7B-Base. Bold and underline indicate the best and second-best results, respectively.

λ_{R}	MATH500		AMC23	
	Avg@8	Pass@8	Avg@8	Pass@8
0	69.18	88.60	41.88	75.00
0.05	69.79	89.80	42.81	77.50
0.10	**70.25**	**90.40**	**43.75**	**80.00**
0.50	<u>70.02</u>	<u>90.00</u>	<u>43.31</u>	<u>77.50</u>
1.0	69.86	89.60	43.06	77.50

D.3 Comparison with Adaptive KL-Based Distillation

Table 10: Comparison with adaptive KL-based distillation baselines using Qwen3-1.7B-Base. **Bold** and <u>underline</u> indicate the best and second-best performance, respectively.

Method	MATH500		AMC23		AIME24		AIME25	
	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8
BD-KD	67.45	86.80	38.75	72.50	9.17	20.00	4.58	<u>16.67</u>
AKL	66.43	85.20	37.81	70.00	7.50	16.67	4.58	13.33
ToDi	67.43	86.80	38.75	<u>75.00</u>	8.33	20.00	<u>5.83</u>	<u>16.67</u>
EOPD	<u>68.73</u>	<u>87.60</u>	<u>41.88</u>	<u>75.00</u>	<u>10.42</u>	<u>23.33</u>	<u>5.83</u>	<u>16.67</u>
JB-OPD	**70.25**	**90.40**	**43.75**	**80.00**	**11.67**	**30.00**	**7.50**	**23.33**

JB-OPD adapts the distillation objective according to downstream trajectory sensitivity, whereas several existing adaptive distillation methods modify KL matching using local teacher–student distribution statistics. To examine whether future sensitivity contributes information beyond such local criteria, we additionally compare with BD-KD (Amara et al., 2022), AKL (Wu et al., 2025), ToDi (Jung et al., 2025), and EOPD (Jin et al., 2026). All methods use the same Qwen3-8B teacher, Qwen3-1.7B student, training prompts, and optimization budget. JB-OPD achieves the best overall performance among the compared adaptive distillation methods across all four benchmarks. These results indicate that incorporating downstream trajectory sensitivity provides an effective signal for adapting token-level supervision and captures complementary information beyond local distributional statistics such as entropy or teacher–student discrepancy.