TRAJPD: CONSTRUCTING TRAJECTORY-AWARE TARGETS FOR ON-POLICY DISTILLATION IN TOOL-INTEGRATED REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transferring tool-integrated reasoning (TIR) capabilities to small language models (SLMs) benefits from supervision adapted to their multi-turn interactions. On-policy distillation (OPD) provides dense teacher guidance on trajectories generated by the student. In multi-turn TIR, student responses and tool feedback shape subsequent prediction contexts, and the tokens within each complete response jointly form an action or answer. This motivates constructing targets that remain close to the student's current predictions while retaining a specified teacher contribution across each response. We propose Trajectory-Aware Policy Distillation (TrajPD), which constructs targets jointly within each complete response. TrajPD selects targets that minimize the KL divergence from the targets to the student's predictions, averaged over positions within the response, while satisfying a response-level constraint on the retained teacher contribution. At the sampled contexts and the start of an update, the weighted distillation gradient preserves the prescribed component along the direct teacher gradient in logit space, aggregated across the response. The targets are obtained by adjusting the student's log probabilities along directions derived from differences between teacher and student probabilities, using a single scale for each response. A weight determined by the average divergence between teacher and student predictions within the response determines this scale and also weights the distillation loss. Evaluation on four benchmarks spanning mathematical reasoning, scientific reasoning, and code generation shows improved aggregate performance at both student scales, with gains across multiple tasks and evaluation metrics.

1 INTRODUCTION

Tool-integrated reasoning (TIR) enables language models to solve tasks through reasoning, tool use, and environment feedback (Cai et al., 2024), such as mathematical reasoning (Gou et al., 2024), code generation (Bai et al., 2026), and search (Ma et al., 2025). Unlike single-pass generation, TIR unfolds through a sequence of model responses and environment observations. Each tool call returns an observation, which becomes part of the context for subsequent reasoning (Lin & Xu, 2025). Therefore, transferring TIR capability to small language models (SLMs) requires supervision that adapts to the changing contexts created by the student's own interactions (Lu et al., 2025).

Reinforcement learning (RL) can train tool-using agents from final task outcomes without requiring labels for every intermediate action (Chen et al., 2025). However, a final reward provides little direct information about which reasoning steps or tool calls contributed to the results (Feng et al., 2025; Xie et al., 2026). For SLMs, sampled trajectories may also receive similar rewards (Choi et al., 2025; He et al., 2026), leaving less variation for optimization methods that compare rewards within a group (Liu et al., 2025). On-policy distillation (OPD) provides additional guidance by querying a teacher on trajectories generated by the student (Agarwal et al., 2024).

The model's responses and tool feedback become part of the context for subsequent predictions (Schick et al., 2023). During on-policy training, direct teacher matching uses the teacher distribution at each visited context as the supervision target (Gu et al., 2024). In multi-turn TIR, these contexts change as the interaction proceeds, and the difference between teacher and student predictions can

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.

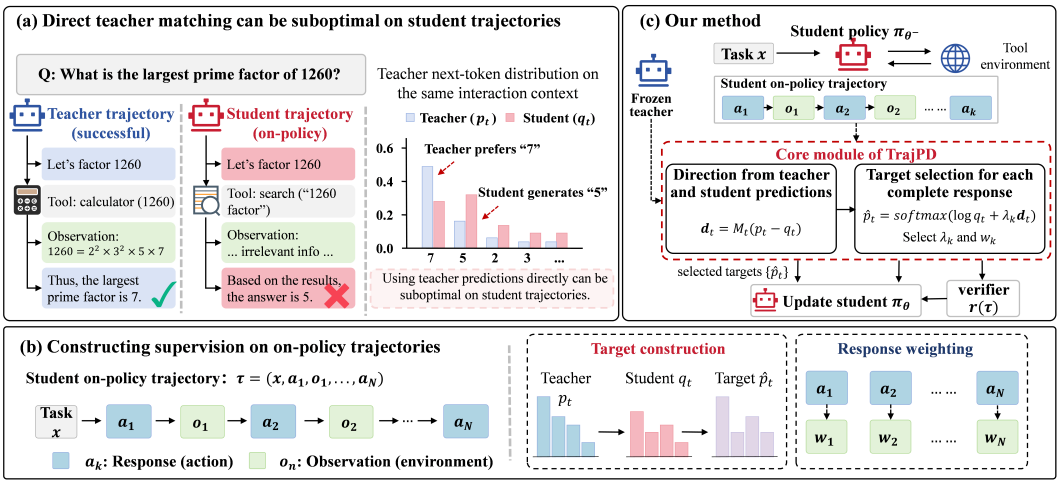

Figure 1: Motivation and overview of TrajPD. (a) Even under the same interaction context, directly using the teacher distribution at each token position can be suboptimal. (b) In multi-turn TIR, student responses and tool feedback continually change the contexts for subsequent predictions, and the difference between teacher and student predictions can vary across responses. (c) TrajPD constructs targets from teacher guidance on trajectories generated by the student. The verifier score of the completed interaction is used in the policy update.

vary across positions within a complete response (Xu et al., 2025). This motivates constructing targets for the complete response, which we refer to as a decision. The targets remain close to the student's current predictions while retaining a specified contribution from the teacher across the response.

We introduce **Traj**ectory-Aware **P**olicy **D**istillation (TrajPD), which constructs the targets for all positions in a response together. TrajPD starts from the student's current predictions and adjusts them using the difference between the teacher and student probabilities. One scale is shared by all positions in the response. TrajPD computes a weight from the average Jensen–Shannon divergence over the response. Given this weight, it selects the scale shared by all positions in that response. At the start of an update, the gradient from the constructed targets preserves the specified component of the gradient from direct teacher matching when the positions in a response are considered together. Among all targets that preserve this teacher contribution, TrajPD minimizes the average KL divergence from the constructed targets to the student's current predictions.

Our contributions are as follows:

- We propose **TrajPD**, which constructs distillation targets for all positions in each model response together. The targets are derived from the student's current predictions and retain a specified contribution from the teacher across the response.
- We establish a minimum-KL characterization of the targets constructed by TrajPD. At the start of an update, their gradient preserves a specified component of the gradient from direct teacher matching in full-vocabulary logit space when the positions in a response are considered.
- We evaluate TrajPD with Qwen3-0.6B and Qwen3-1.7B students and a Qwen3-4B teacher on AIME 2024, AIME 2025, GPQA-Diamond, and LiveCodeBench. Across all benchmark and model size settings, the constructed targets have lower KL divergence from the current student predictions. TrajPD improves scientific reasoning and code generation compared with SOD and achieves the highest four-benchmark mean Avg@8 among the evaluated student methods.

2 Related Work

Reinforcement Learning for Tool-Integrated Reasoning. Reinforcement learning (RL) is used to optimize LLM reasoning and agentic decision-making (Deng et al., 2026), with PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) providing a wide range of optimization frameworks. In multi-turn tool-integrated settings, existing methods address unstable training and the limited feedback provided by final rewards. SimpleTIR uses trajectory filtering to stabilize training (Xue et al., 2026), while STAPO identifies anomalous steps using entropy-based signals (Qi et al., 2026). Another line of work provides supervision at each turn. MatchTIR (Qu et al., 2026) and AT^{2}PO (Zong et al., 2026) operate at the turn level, whereas CW–GRPO uses process evaluations to reweight trajectory-level advantages (Wang et al., 2026).

On-Policy Distillation for Language Models and Agents. Knowledge distillation transfers predictive behavior from a teacher to a student model (Hinton et al., 2015). For autoregressive models, the histories used during offline training can differ from those encountered by the student at inference time. Learning from histories produced by the learner is a central idea in imitation learning (Ross et al., 2011). On-policy distillation (OPD) applies this idea by obtaining teacher supervision on trajectories sampled from the current student policy (Song & Zheng, 2026), while GKD provides a general framework for distillation on sequences produced by the student (Agarwal et al., 2024). For multi-turn agents, SOD changes the distillation strength according to the divergence between teacher and student predictions at each step (Zhong et al., 2026). ATOD combines an evolving schedule between OPD and RL with reweighting across turns (Tan et al., 2026), and TurnOPD allocates rollout and loss budgets across different stages of an interaction (Zhou et al., 2026).

Distillation Target Construction. Another line of work studies the information and predictions used to form distillation targets. Privileged information (PI) provides additional supervision during training (Vapnik & Vashist, 2009). For example, π-Distill trains a teacher conditioned on PI together with a student that does not receive this context (Penaloza et al., 2026). EDGE-OPD uses privileged evidence to select token positions for distillation updates (Lazaridis et al., 2026). Veto constructs targets from teacher and student predictions to emphasize agreement between two models (Jang et al., 2026). TrajPD also constructs the distillation target from the current teacher and student predictions, but performs this construction over all positions in a complete model response.

3 Preliminaries

3.1 Problem Definition

Let $\mathcal{D}$ denote the task distribution, and let $x \sim \mathcal{D}$ include a task prompt as well as descriptions of the available tools. A student policy π_θ with trainable parameters θ generates responses conditioned on x and the preceding interaction. When the student issues a tool call, the environment executes it and returns an observation, which is included in the context for subsequent generation. The interaction ends when the student returns a final answer or reaches the interaction budget. The resulting trajectory τ records the student's responses and the environment observations.

During training, a teacher policy π_{ϕ^*} with fixed parameters ϕ^* provides supervision on the same contexts visited by the student. A verifier assigns each completed trajectory a score $r(\tau) \in [0, 1]$. The learning objective is to improve the student's expected task performance under its own tool interactions. At evaluation time, the student generates responses and tool calls without teacher supervision.

3.2 Trajectories and Decisions

A decision is one complete response from the student. It may contain reasoning and can end with a tool call or a final answer. It ends when the student yields control or finishes the task. Let N be the number of decisions and index them by $k \in \{1, \ldots, N\}$. Let a_k denote the response at decision k. For $k < N$, o_k denotes the observation returned after that response. An observation may be empty. The trajectory is

$$\tau = (x, a_1, o_1, \ldots, a_{N-1}, o_{N-1}, a_N). \tag{1}$$

Let $\mathcal{I}_k$ contain the positions of the tokens generated by the student in decision k, with $|\mathcal{I}_k|$ denoting its size. These sets are disjoint, and $\mathcal{T} = \bigcup_{k=1}^{N} \mathcal{I}_k$ contains all positions at which the student generates tokens in the trajectory. At position t, h_t denotes the context available before the student samples token y_t. It contains the task, previous responses, observations, control messages, and the current response prefix. An end-of-sequence token sampled by the student is also included in $\mathcal{T}$. Observations and forced control tokens enter the history and are excluded from the loss.

3.3 On-Policy Teacher Supervision

At the start of each iteration, θ^- is set to the current student parameters and kept fixed throughout the iteration. The policy π_{θ^-} generates one trajectory for each task using the sampling procedure used for training. The teacher evaluates the same contexts in these trajectories and provides its next-token probabilities. The teacher and student models use the same tokenizer and vocabulary $\mathcal{V}$, with v denoting a token.

For each context h_t, let $\mathcal{K}_t \subset \mathcal{V}$ contain the 16 tokens with the highest probabilities under the teacher, and let $U_t = \mathcal{V} \setminus \mathcal{K}_t$ contain all remaining tokens. The shared partition $\mathcal{P}_t$ for the teacher and student contains one element for each token in $\mathcal{K}_t$ and one additional element for U_t. The tokens in U_t are grouped into one element. This preserves their total probability mass while reducing the number of entries used to construct the target.

$$\begin{aligned} p_t(C) &= \sum_{v \in C} \pi_{\phi^*}(v \mid h_t), \\ q_t(C) &= \sum_{v \in C} \pi_{\theta^-}(v \mid h_t), \\ q_{\theta,t}(C) &= \sum_{v \in C} \pi_{\theta}(v \mid h_t). \end{aligned} \tag{2}$$

p_t denotes the teacher distribution, q_t is the student distribution before the update, and $q_{\theta,t}$ is the student distribution used for optimization. All three are defined on $\mathcal{P}_t$. At the start of the update, $q_{\theta,t} = q_t$. The partition keeps the probabilities of the selected tokens and the total probability assigned to all remaining tokens. Probabilities such as $\pi_\theta(y_t \mid h_t)$ refer to the individual sampled token y_t in the full vocabulary rather than to an element of $\mathcal{P}_t$.

For distributions u and v over the same finite set $\mathcal{S}$, the forward Kullback–Leibler divergence is denoted by $D_{\mathrm{KL}}(u|v)$. All logarithms are natural. Directly matching the teacher at position t gives the loss

$$\ell_t^{\mathrm{direct}}(\theta) = D_{\mathrm{KL}}(p_t \| q_{\theta,t}). \tag{3}$$

TrajPD uses both p and q to construct a new target. Each collected batch is used for one optimizer step starting from $\theta = \theta^-$, followed by fresh trajectories. After the update, the student generates a new set of trajectories for the next iteration.

4 Trajectory-Aware Policy Distillation

4.1 Overview

TrajPD treats each complete response as one decision, since its generated tokens together determine the model's actions before the next environment response. The student's predictions can vary substantially across positions within the same response. Applying the same fraction of the teacher correction at every position can introduce more change to the student's predictions than necessary. Instead, TrajPD selects the targets for the complete response under a requirement that the response retain the teacher correction. Among the targets satisfying this requirement, TrajPD chooses those with the smallest mean KL divergence from the student's current predictions. Tool observations can influence the predictions in subsequent responses, while TrajPD constructs targets separately

for each complete response. The response weight also enters the distillation loss, so the teacher correction used to construct the targets is preserved when updating the student.

TrajPD constructs supervision in three stages. First, it forms candidate targets from the teacher and student predictions. It then selects an individual target scale for all token positions in each complete response as well as a response weight. Finally, it updates the student using the selected targets together with verified task scores. The full training procedure can be found in Algorithm 1 in Appendix B.

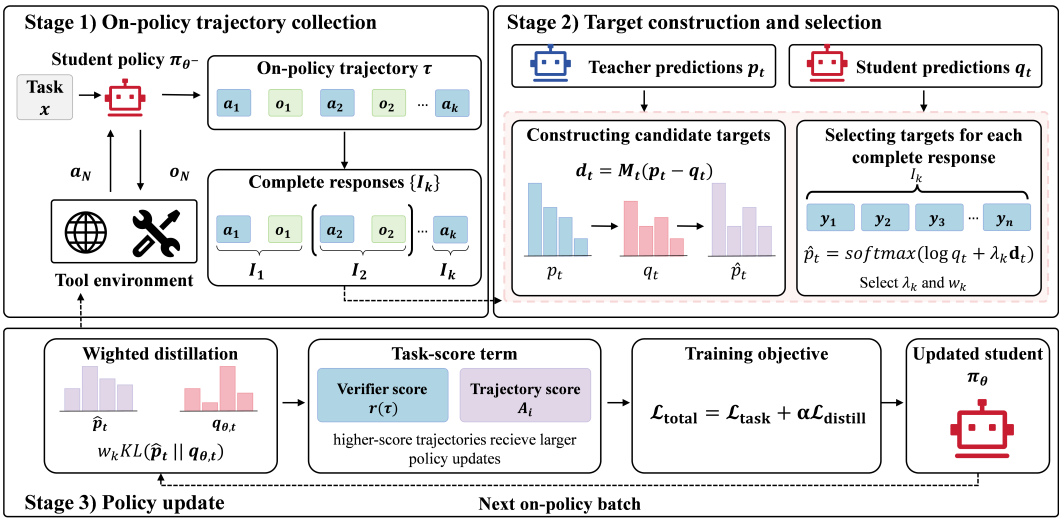

Figure 2: Overview of TrajPD. Stage 1 collects on-policy trajectories through tool interaction. Stage 2 constructs targets and determines a weight for each complete response. Stage 3 updates the student using the selected targets and verified task scores.

4.2 Target Construction from Teacher and Student Predictions

At each sampled position, $p_t - q_t$ gives the difference between the teacher prediction p_t and the student prediction q_t. This difference is represented in full-vocabulary logit space using $\mathbf{M}_t = \mathrm{diag}(1, \ldots, 1, c_t)$, where the final entry corresponds to the grouped unselected tokens. Here c_t is the sum of squared student probabilities after normalization over the unselected tokens. It accounts for how the unselected probability mass is distributed when these tokens are grouped into one element. Appendix A.1 derives this metric. We define $\mathbf{d}_t$ and G_k as follows, with $\top$ denoting transpose,

$$\mathbf{d}_t = \mathbf{M}_t(p_t - q_t), \qquad G_k = \frac{1}{|\mathcal{I}_k|} \sum_{t \in \mathcal{I}_k} (p_t - q_t)^\top \mathbf{d}_t. \tag{4}$$

The target selection in the next section preserves the same component along the direct teacher gradient as direct teacher matching in full-vocabulary logit space. Appendix A.1 derives this relation.

For $\lambda \geq 0$, we define a candidate target $Q_t(\lambda)$. The value λ_k selected for decision k determines the final target $\widehat{p}_t$.

$$Q_t(\lambda) = \mathrm{softmax}(\log q_t + \lambda \mathbf{d}_t), \qquad \widehat{p}_t = Q_t(\lambda_k). \tag{5}$$

The logarithm acts entrywise, and $\lambda = 0$ recovers q_t. These candidates specify possible changes. The next section selects λ_k so that the weighted targets retain the prescribed teacher contribution for decision k.

4.3 Target Selection for Complete Responses

For decision k, we set $w_k = 1 + \delta_k$, where δ_k is the average Jensen–Shannon divergence between p_t and q_t across the sampled positions in the response, divided by $\log 2$. Thus $w_k \in [1, 2]$ depends on the average difference between the teacher and student predictions, rather than the number of tokens in the response. Jensen–Shannon divergence is the mean of the two KL divergences to the midpoint distribution $(p_t + q_t)/2$. Given w_k, we select targets whose weighted gradient retains the same component along the direct teacher gradient as direct teacher matching.

Let Δ_t be the probability simplex on $\mathcal{P}_t$ and $u_t \in \Delta_t$ a candidate target. We select targets by

$$\begin{aligned} \underset{\{u_t \in \Delta_t\}}{\text{minimize}} \quad & \frac{1}{|\mathcal{I}_k|} \sum_{t \in \mathcal{I}_k} D_{\mathrm{KL}}(u_t \| q_t) \\ \text{subject to} \quad & \frac{w_k}{|\mathcal{I}_k|} \sum_{t \in \mathcal{I}_k} (u_t - q_t)^\top \mathbf{d}_t \geq G_k. \end{aligned} \tag{6}$$

The solution has the exponential form in Eq. (5). For $G_k > 0$, bisection gives the unique $\lambda_k > 0$ for which the constraint in Eq. (6) holds with equality. For $G_k = 0$, set $\lambda_k = 0$. The resulting targets allow different positions to contribute different amounts while satisfying the constraint in Eq. (6). Appendix A.2.8 provides a two-position example. The objective averages $D_{\mathrm{KL}}(u_t \| q_t)$ over the sampled positions in the response.

The target $q_t + (p_t - q_t)/w_k$ gives the same initial weighted gradient as direct teacher matching. The solution of Eq. (6) retains the same component along the direct teacher gradient while minimizing the mean KL divergence from the student predictions. At the starting logits, the difference between the two gradients is orthogonal to the direct teacher gradient. Therefore, a smaller target KL does not imply a smaller weighted gradient norm. Increasing w_k changes the selected targets while preserving the same component along the direct teacher gradient. Appendix A.2.7 derives the KL improvement and distinguishes this logit property from parameter updates.

4.4 Policy Update with Targets and Task Scores

After selecting the targets, TrajPD updates the student using the distillation loss and the verifier score assigned to each completed trajectory. The weight w_k used in target selection also weights the corresponding distillation loss. At the start of the update, this weighted loss preserves the gradient condition used in target selection.

Let $B \geq 2$ be the batch size and $i \in \{1, \ldots, B\}$ index trajectories. Let $\mathcal{T}_i$ contain the sampled positions generated by the student, and let $k_i(t)$ denote the response containing position t. We define $Z_{\mathrm{act}} = \sum_i |\mathcal{T}_i|$ as the total number of these positions in the batch. Each trajectory τ_i receives a verifier score $r_i = r(\tau_i)$. For trajectory i, we define $A_i = r_i - \frac{1}{B-1} \sum_{j \neq i} r_j$, which compares its verifier score with the mean score of the remaining trajectories in the batch. We treat A_i as fixed during differentiation.

For a sampled token $y_{i,t}$ conditioned on $h_{i,t}$ and a coefficient $\alpha > 0$, the training objective is

$$\begin{aligned} \mathcal{L}_{\mathrm{distill}}(\theta) &= \frac{1}{Z_{\mathrm{act}}} \sum_{i=1}^{B} \sum_{t \in \mathcal{T}_i} w_{i,k_i(t)} D_{\mathrm{KL}}(\widehat{p}_{i,t} \| q_{\theta,i,t}), \\ \mathcal{L}_{\mathrm{task}}(\theta) &= -\frac{1}{Z_{\mathrm{act}}} \sum_{i=1}^{B} \sum_{t \in \mathcal{T}_i} A_i \log \pi_\theta(y_{i,t} \mid h_{i,t}), \\ \mathcal{L}_{\mathrm{total}}(\theta) &= \mathcal{L}_{\mathrm{task}}(\theta) + \alpha \mathcal{L}_{\mathrm{distill}}(\theta). \end{aligned} \tag{7}$$

Both losses are averaged over the same sampled student positions. The model inputs, targets, weights, and advantages remain fixed during each AdamW update. After the update, the student collects a new batch of trajectories and constructs new targets. Appendix A.3 derives the parameter gradient.

5 EXPERIMENTS

5.1 EXPERIMENT SETUPS

We use Open-AgentRL-SFT-3K and Open-AgentRL-30K released by Yu et al. (2025) for training, with Open-AgentRL-30K used to collect on-policy trajectories. Baselines consist of Vanilla, SFT, GRPO, On-Policy Distillation (OPD), Entropy-Aware On-Policy Distillation (EOPD) (Jin et al., 2026), and Step-wise On-Policy Distillation (SOD) (Zhong et al., 2026). Evaluation covers AIME 24/25, GPQA-Diamond, and LiveCodeBench. We generate eight responses per question at temperature 1.0 and top-p 0.6, and report average binary correctness (Avg@8) and the recorded Pass@8. Training is implemented with VeRL (Sheng et al., 2024), with SGLang for generation and Sandbox Fusion for Python execution. Additional implementation details are provided in Appendix B.

5.2 OVERALL PERFORMANCE

As shown in Table 1, TrajPD achieves the highest mean Avg@8 at both student sizes and ranks first or second in 17 of the 18 reported Avg@8, Pass@8, and mean entries. For the 1.7B student, TrajPD ranks first or second in every reported entry. Compared with SOD, the mean Avg@8 increases from 21.14% to 23.81% at 0.6B and from 37.40% to 38.29% at 1.7B. The largest Avg@8 gains appear on GPQA, with improvements of 9.85 and 4.67 percentage points, while LiveCodeBench also improves at both sizes. Additional Pass@k results are provided in Appendix C.2.

Table 1: Performance on four benchmarks (%). Bold indicates the best performance and underline indicates second-best, within each student size.

	AIME 2024		AIME 2025		GPQA		LiveCodeBench		Mean
Method	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8
Qwen3-4B Teacher	71.67	82.22	65.83	78.14	58.71	79.05	66.43	88.37	65.66
Qwen3-0.6B Student									
Vanilla	1.25	5.28	0.42	2.17	**28.35**	55.62	18.79	38.84	12.20
SFT	9.58	24.60	6.25	16.18	26.45	<u>60.97</u>	5.43	17.05	11.93
GRPO	2.50	5.58	1.67	7.47	27.02	60.49	20.50	38.77	12.92
OPD	6.25	20.51	12.08	26.70	<u>27.40</u>	60.64	21.21	45.48	16.74
EOPD	10.00	23.89	15.83	27.30	26.58	58.28	<u>29.21</u>	<u>58.15</u>	20.41
SOD	**18.33**	<u>36.01</u>	**20.00**	<u>35.95</u>	17.36	39.96	28.86	56.05	<u>21.14</u>
TrajPD (ours)	<u>17.08</u>	**36.23**	<u>19.17</u>	**38.44**	27.21	**62.86**	**31.79**	**62.41**	**23.81**
Qwen3-1.7B Student									
Vanilla	2.92	5.51	2.08	9.66	27.27	50.60	21.86	37.52	13.53
SFT	12.08	29.53	5.83	20.65	14.90	45.23	4.57	17.11	9.35
GRPO	8.75	26.21	9.58	21.65	30.05	56.77	22.71	33.49	17.77
OPD	27.50	42.78	21.25	34.82	32.26	57.57	41.14	**68.98**	30.54
EOPD	27.92	38.81	25.83	41.73	<u>33.46</u>	**59.94**	40.57	63.40	31.95
SOD	**44.58**	<u>61.42</u>	**34.17**	**51.31**	29.36	56.45	<u>41.50</u>	67.62	<u>37.40</u>
TrajPD (ours)	<u>41.67</u>	**65.50**	<u>33.33</u>	<u>45.30</u>	**34.03**	<u>59.62</u>	**44.14**	<u>67.70</u>	**38.29**

5.3 COMPONENT CONTRIBUTIONS AND SENSITIVITY

Table 2 compares components and three values of α at 0.6B, alongside the scales solved during training.

Target construction improves performance beyond direct teacher targets. Replacing the selected targets with the teacher distribution reduces six of the eight benchmark-metric results and lowers the mean Avg@8 from 23.81% to 22.16%. Over the final ten updates, the mean KL from the selected targets to the student predictions is 0.1174, compared with 0.2564 for the teacher distribution.

Response weights and task scores both contribute to performance. Removing the response weights reduces the mean Avg@8 from 23.81% to 20.41%, while removing the task scores reduces it to 20.37%.

$\alpha = 1$ **performs best among the tested coefficients.** The mean Avg@8 increases from 18.86% at $\alpha = 0.1$ to 20.19% at $\alpha = 0.5$ and 23.81% at $\alpha = 1$. The mean task return over the final ten updates follows the same trend, increasing from 0.1002 to 0.1917 and 0.2480, while $\bar{\lambda}$ remains between 4.222 and 4.355.

Table 2: Components and coefficient sensitivity at 0.6B (%). Bold indicates the best performance and underline indicates second-best. $\bar{\lambda}$ is the mean logged scale over updates 91–100; – means the construction is not used.

	AIME 2024		AIME 2025		GPQA		LiveCodeBench		$\bar{\lambda}$
Variant	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	Avg@8	Pass@8	
TrajPD ($\alpha = 1$)	<u>17.08</u>	<u>36.23</u>	**19.17**	**38.44**	27.21	**62.86**	**31.79**	<u>62.41</u>	4.222
w/o target construction	**18.75**	**45.48**	<u>18.33</u>	<u>33.75</u>	25.00	52.42	26.57	56.96	–
w/o response weights	11.67	26.18	15.00	33.70	<u>28.35</u>	58.04	26.64	54.46	4.695
w/o task scores	9.58	24.06	14.58	27.53	**28.47**	<u>60.33</u>	28.86	60.73	4.183
$\alpha = 0.1$	5.00	20.78	12.50	26.89	26.52	56.34	<u>31.43</u>	**67.08**	4.355
$\alpha = 0.5$	14.17	29.23	14.17	28.52	25.13	58.87	27.29	58.51	4.247

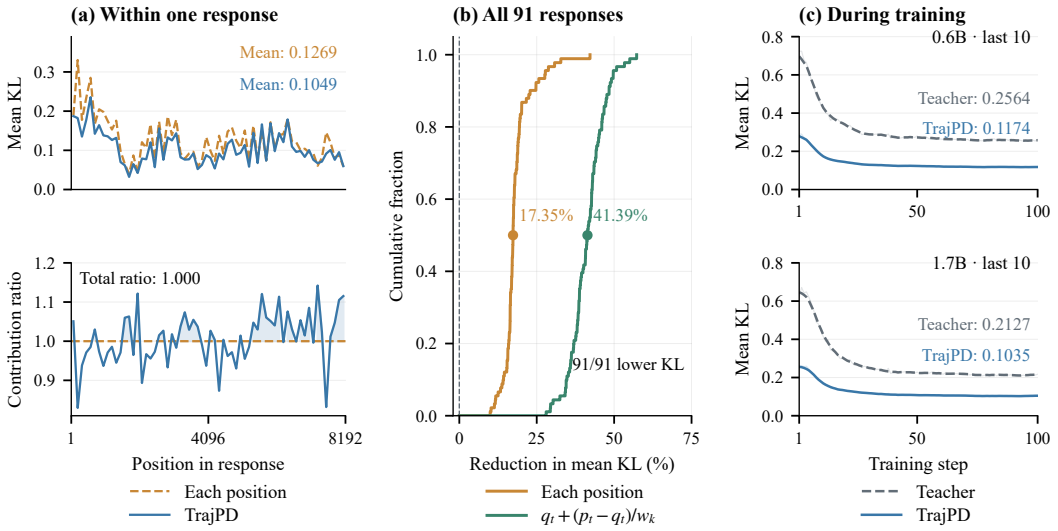

Figure 3: Analysis of TrajPD target construction. (a) KL divergence and contribution ratios across 64 consecutive bins of one response; Mean and Total ratio summarize the full response. (b) KL reduction across 91 responses; dots denote medians and the vertical line denotes zero. (c) Mean KL from the selected targets and teacher distributions to the current student predictions during training; thin lines show raw values and bold lines show seven-step averages. Labels marked last 10 give means over updates 91–100.

5.4 Behavioral and Mechanistic Analysis

Figure 3 analyzes the selected targets at three levels: within a response, across 91 recorded responses, and throughout training. For the recorded responses, all methods use the same teacher and student predictions and satisfy the same prescribed contribution, while the alternatives either determine the target independently at each position or use $q_t + (p_t - q_t)/w_k$.

TrajPD improves response completion compared with SOD. As shown in Figure 4, TrajPD achieves higher completion rates than SOD across all benchmark and model-size groups. On GPQA, this improvement is accompanied by higher overall accuracy, while the accuracy among completed responses remains within two percentage points of the corresponding accuracy for SOD. Representative trajectories and additional error analysis are provided in Appendices D and I.3.

Response-level target selection reduces the required change from the student predictions. A single λ_k allows different positions within a response to change by different amounts while satisfying the constraint in Eq. (6). Across all 91 recorded responses, the selected targets have lower mean KL than both comparison targets. Relative to determining targets independently at each position, the median KL reduction is 17.35%, and the reduction is observed in every benchmark and model-size group.

The same target property is maintained throughout training. Across 100 updates at both model sizes, the selected targets remain closer to the student predictions than the teacher distributions, while the mean relative residual of the constraint remains below 10^{-15}. Additional analysis of the solved λ_k is provided in Appendix C.4.

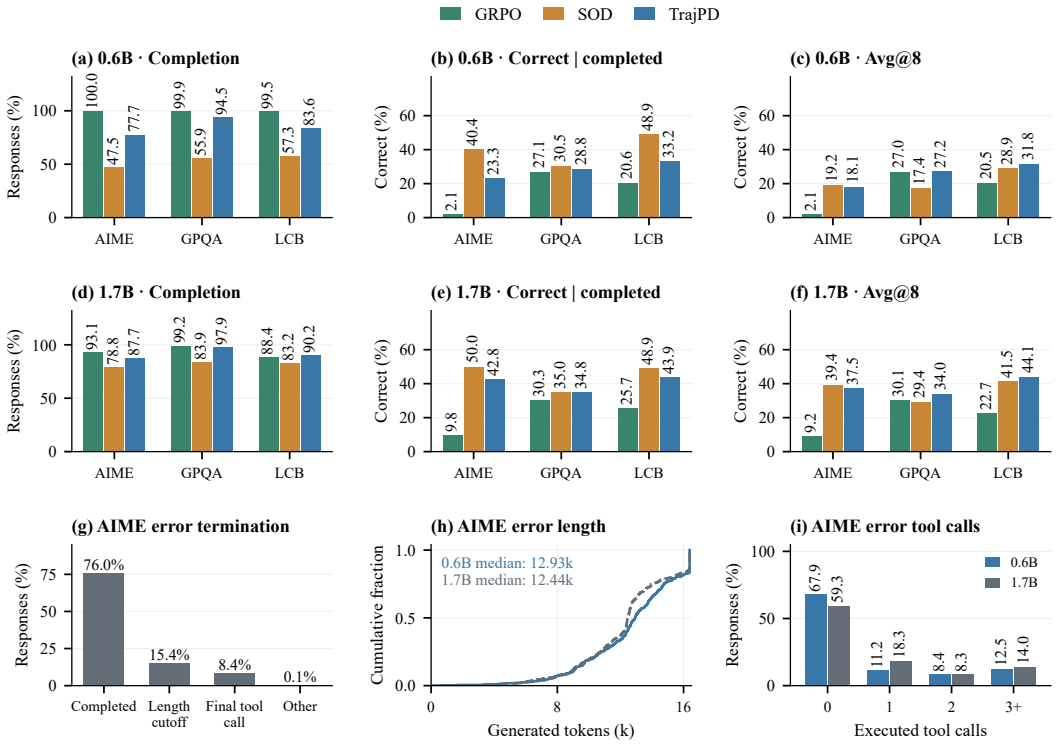

Figure 4: Response behavior across methods. The first two rows compare completion rate, correctness among completed responses, and Avg@8, with AIME 2024 and 2025 pooled. The last row summarizes termination, response length, and tool usage for 693 incorrect TrajPD responses on AIME24/25; blue and gray denote 0.6B and 1.7B, respectively.

6 Conclusion

We introduce TrajPD for on-policy distillation over tool-interaction trajectories. TrajPD constructs response-level targets that satisfy the prescribed teacher contribution while remaining close to the current student predictions, and combines these targets with response weights and task scores for policy updates. Theoretical analysis and recorded predictions support the target construction, while experiments across mathematics, science, and code show improved overall performance at both student scales.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, including grammar and spelling checks and sentence-level polishing, to improve the readability of the manuscript. We also used these tools to assist with code debugging. We did not use generative AI tools to format references. We reviewed all work produced with AI assistance. Specifically, we checked that the language edits preserved the intended meaning and reviewed AI-assisted code changes for correctness. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide all the details to reproduce our results in the appendix to ensure that the results presented in this paper are reproducible. All code and datasets are available in an anonymous repository for replication and verification (https://anonymous.4open.science/r/experiment-E7BF). The experimental settings are also provided in the appendix and in the configuration files included with the code.

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 B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Fei Bai, Yingqian Min, Beichen Zhang, Zhipeng Chen, Xin Zhao, Lei Fang, Zheng Liu, Zhongyuan Wang, and Hongteng Xu. Towards effective code-integrated reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30022–30030, 2026.

Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. Large language models as tool makers. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 54067–54089, 2024.

Fan Chen, Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Outcome-based online reinforcement learning: Algorithms and fundamental limits. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 106095–106133. Curran Associates, Inc., 2025. doi: 10.52202/085713-3541.

ChangSu Choi, Hoyun Song, Dongyeon Kim, WooHyeon Jung, Minkyung Cho, Sunjin Park, NohHyeob Bae, Seona Yu, and KyungTae Lim. Mentor: A reinforcement learning framework for model enhancement via teacher-optimized rewards in small models. *arXiv e-prints*, pp. arXiv–2510, 2025.

Yihe Deng, I-Hung Hsu, Jun Yan, Zifeng Wang, Rujun Han, Gufeng Zhang, Yanfei Chen, Wei Wang, Tomas Pfister, and Chen-Yu Lee. Supervised reinforcement learning: From expert trajectories to step-wise reasoning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 75893–75911, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Minlie Huang, Nan Duan, Weizhu Chen, et al. Tora: A tool-integrated reasoning agent for mathematical problem solving. In *International Conference on Learning Representations*, volume 2024, pp. 48362–48395, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Xixiang He, Qiyao Sun, Ao Cheng, Xingming Li, Xuanyu Ji, Hailun Lu, Runke Huang, and Qingyong Hu. Advantage collapse in group relative policy optimization: Diagnosis and mitigation. In *Forty-third International Conference on Machine Learning*, 2026.

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 Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.2094.

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. In *Forty-third International Conference on Machine Learning*, 2026.

Aristotelis Lazaridis, Dylan Bates, Aman Sharma, Brian King, Vincent Lu, and Jack FitzGerald. Edge-opd: Internalizing privileged context with evidence guided on-policy distillation, 2026.

Heng Lin and Zhongwen Xu. Understanding tool-integrated reasoning, 2025.

Jiacai Liu, Chaojie Wang, Chris Liu, Liang Zeng, Rui Yan, Yiwen Sun, and Yang Liu. Dapo : Improving multi-step reasoning abilities of large language models with direct advantage-based policy optimization. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 71445–71477. Curran Associates, Inc., 2025. doi: 10.52202/085713-2401.

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Wei Liu, Jian Luan, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. Demystifying small language models for edge deployment. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14747–14764, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.718.

Zexiong Ma, Chao Peng, Qunhong Zeng, Pengfei Gao, Yanzhen Zou, and Bing Xie. Tool-integrated reinforcement learning for repo deep search. *arXiv preprint arXiv:2508.03012*, 2025.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. In *The 1st Workshop on Scaling Post-training for LLMs*, 2026.

Qiuyi Qi, Tian Liang, Mutian Bao, Jinjian Zhang, Dongnan Liu, Wei Zhou, Linjian Mo, Ming Kong, Jie Liu, Feng Zhang, and Qiang Zhu. STAPO: Selective trajectory-aware policy optimization for LLM agent training. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28371–28392, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1308.

Changle Qu, Sunhao Dai, Hengyi Cai, Jun Xu, Shuaiqiang Wang, and Dawei Yin. MatchTIR: Fine-grained supervision for tool-integrated reasoning via bipartite matching. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11953–11968, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.549.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

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

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. *arXiv preprint arXiv: 2409.19256*, 2024.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models, 2026.

Qitai Tan, Zefang Zong, Mo Li, Yipeng Shi, Yang Li, and Peng Chen. Atod: Annealed turn-aware on-policy distillation for multi-turn agentic tasks, 2026.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2009.06.042. Advances in Neural Networks Research: IJCNN2009.

Junzhe Wang, Zhiheng Xi, Yajie Yang, Hao Luo, Shihan Dou, Tao Gui, and Qi Zhang. Enhancing LLM-based search agents via contribution weighted group relative policy optimization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31704–31718, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1462.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *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 Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 64616–64646, 2025.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun MA, and Bo An. SimpleTIR: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zhaochen Yu, Ling Yang, Jiaru Zou, Shuicheng Yan, and Mengdi Wang. Demystifying reinforcement learning in agentic reasoning. *arXiv preprint arXiv:2510.11701*, 2025.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. Sod: Step-wise on-policy distillation for small language model agents, 2026.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. Turnopd: Making on-policy distillation turn-aware for efficient long-horizon agent training, 2026.

Zefang Zong, Dingwei Chen, Yang Li, Qi Yi, Bo Zhou, Chengming Li, BO Qian, Peng Chen, and Jie Jiang. AT²PO: Agentic turn-based policy optimization via tree search. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24120–24143, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1106.

A THEORETICAL DERIVATIONS

The following three parts derive the probability metric, target construction across a response, and student update in Sections 4.2–4.4.

A.1 CANDIDATE TARGET CONSTRUCTION

A.1.1 SHARED PROBABILITIES AND FIXED QUANTITIES

We use the notation of Section 3 and restore the trajectory index i throughout this appendix. All quantities at each position carry the same trajectory index, including the partition $\mathcal{P}_{i,t}$, selected set $\mathcal{K}_{i,t}$, and unselected set $U_{i,t}$. For trajectory τ_i, let N_i be its number of decisions, $\mathcal{I}_{i,k}$ the sampled positions in decision k, and $\mathcal{T}_i = \bigcup_{k=1}^{N_i} \mathcal{I}_{i,k}$. The history and sampled token at position t are $h_{i,t}$ and $y_{i,t}$.

The partition and grouped probabilities follow Section 3. We assume positive teacher and student probabilities and use $0 \log 0 = 0$ for candidate targets.

The KL chain rule decomposes full-vocabulary divergence into divergence between partition masses and a nonnegative conditional divergence within each element. Thus grouping preserves supervision of the selected tokens and the total unselected mass while omitting within-element conditional divergence.

Teacher probabilities and the fixed student's probabilities define the targets in Sections 4.2–4.3. To derive their effect on learning, we first differentiate a fixed grouped target and then recover the metric $\mathbf{M}_{i,t}$ from the resulting logit gradients.

A.1.2 TOKEN GRADIENTS

Fix a sampled history and suppress i, t temporarily. Let z_v be the current student's logit for token v, let $\pi_\theta(v)$ abbreviate $\pi_\theta(v \mid h)$ at this history h, and let $C(v)$ be the partition element containing v. The grouped student probability is $q_\theta(C) = \sum_{v \in C} \pi_\theta(v)$. Here q_θ and $\widehat{p}$ abbreviate the grouped student distribution and target at the fixed history. For a fixed grouped target $\widehat{p}$, write $\ell = D_{\mathrm{KL}}(\widehat{p} \| q_\theta)$. Its derivative is

$$\frac{\partial \ell}{\partial z_v} = \pi_\theta(v) - \widehat{p}(C(v)) \frac{\pi_\theta(v)}{q_\theta(C(v))}. \tag{8}$$

For a selected token, this is $\pi_\theta(v) - \widehat{p}(\{v\})$. For an unselected token, its conditional student probability distributes the correction to the total unselected mass.

Differentiating the grouped softmax mass and using target normalization gives this identity.

A.1.3 THE PROBABILITY METRIC

At $\theta = \theta^-$, define $\rho_{i,t}(v \mid C) = \pi_{\theta^-}(v \mid h_{i,t}) / q_{i,t}(C)$ as the fixed student's conditional probability within C. The probability metric is

$$c_{i,t} = \sum_{v \in U_{i,t}} \left(\frac{\pi_{\theta^-}(v \mid h_{i,t})}{q_{i,t}(U_{i,t})} \right)^2, \qquad \mathbf{M}_{i,t} = \mathrm{diag}(1, \ldots, 1, c_{i,t}). \tag{9}$$

To see the resulting inner-product relation, define a matrix $\mathbf{R}_{i,t}$ with rows indexed by vocabulary tokens and columns by partition elements. Its entry at (v, C) is $\rho_{i,t}(v \mid C)$ if $v \in C$, and zero otherwise. The disjoint partition elements give $\mathbf{R}_{i,t}^\top \mathbf{R}_{i,t} = \mathbf{M}_{i,t}$. For any grouped target u, its full-vocabulary logit gradient at θ^- is $\mathbf{R}_{i,t}(q_{i,t} - u)$. Thus, for grouped difference vectors a and b,

$$(\mathbf{R}_{i,t} a)^\top (\mathbf{R}_{i,t} b) = a^\top \mathbf{M}_{i,t} b. \tag{10}$$

Each selected token contributes a diagonal entry of one. The unselected tokens contribute the sum of squared conditional probabilities. This derives the metric used in Section 4.2 directly from the shared partition.

A.2 Target Construction for Complete Responses

A.2.1 Response Weight and Target Objective

The weight defined in Section 4.3 lies in $[1, 2]$ because $0 \leq \mathrm{JS} \leq \log 2$. It is fixed during target construction and differentiation. We now derive the optimum across the response for this weight.

Fix a decision k in trajectory i. Let $\Delta_{i,t}$ be the probability simplex on $\mathcal{P}_{i,t}$, and let $u_t \in \Delta_{i,t}$ be a candidate target. The direction $\mathbf{d}_{i,t}$ and teacher reference $G_{i,k}$ are defined in Section 4.2. Let $F(\{u_t\})$ denote the mean KL from the targets to the student predictions. The target construction problem is

$$\begin{aligned} \underset{\{u_t \in \Delta_{i,t}\}}{\text{minimize}} \quad & F(\{u_t\}) = \frac{1}{|\mathcal{I}_{i,k}|} \sum_{t \in \mathcal{I}_{i,k}} D_{\mathrm{KL}}(u_t \| q_{i,t}), \\ \text{subject to} \quad & \frac{w_{i,k}}{|\mathcal{I}_{i,k}|} \sum_{t \in \mathcal{I}_{i,k}} (u_t - q_{i,t})^\top \mathbf{d}_{i,t} \geq G_{i,k}. \end{aligned} \tag{11}$$

The requirement fixes the teacher contribution, and the objective selects the closest targets in KL among those satisfying it.

A.2.2 Deriving the Exponential Solution

Write $n = |\mathcal{I}_{i,k}|$ for this derivation. Let $\mu_{i,k} \geq 0$ multiply the correction requirement and let η_t multiply the normalization equation at position t. The Lagrangian $\mathcal{J}$ is

$$\begin{aligned} \mathcal{J} = & \frac{1}{n} \sum_{t \in \mathcal{I}_{i,k}} D_{\mathrm{KL}}(u_t \| q_{i,t}) \\ & + \mu_{i,k} \left[G_{i,k} - \frac{w_{i,k}}{n} \sum_{t \in \mathcal{I}_{i,k}} (u_t - q_{i,t})^\top \mathbf{d}_{i,t} \right] \\ & + \frac{1}{n} \sum_{t \in \mathcal{I}_{i,k}} \eta_t \left[\sum_{C \in \mathcal{P}_{i,t}} u_t(C) - 1 \right]. \end{aligned} \tag{12}$$

At an interior stationary point, differentiating with respect to each target entry gives

$$\log \frac{u_t(C)}{q_{i,t}(C)} + 1 - \mu_{i,k} w_{i,k} d_{i,t}(C) + \eta_t = 0. \tag{13}$$

Set $\lambda_{i,k} = w_{i,k}\mu_{i,k}$. Normalization eliminates η_t and gives the candidate u_t^* with normalizing constant $Z_{i,t} = Z_{i,t}(\lambda_{i,k})$,

$$\begin{aligned} u_t^*(C) &= \frac{q_{i,t}(C) \exp\big(\lambda_{i,k} d_{i,t}(C)\big)}{Z_{i,t}}, \\ Z_{i,t} &= \sum_{C \in \mathcal{P}_{i,t}} q_{i,t}(C) \exp\big(\lambda_{i,k} d_{i,t}(C)\big). \end{aligned} \tag{14}$$

When $G_{i,k} > 0$, at least one position has positive JS divergence, so $w_{i,k} > 1$. Choosing $u_t = p_{i,t}$ gives weighted correction $w_{i,k} G_{i,k} > G_{i,k}$. The positive teacher distribution is therefore a strictly feasible interior point. The feasible set is compact and convex, and F is strictly convex, so its minimizer exists and is unique. The requirement holds with equality at the minimizer. Otherwise, moving a small distance toward $q_{i,t}$ would preserve feasibility and reduce F. Stationarity gives Eq. (14) with a shared positive scale, establishing $\widehat{p}_{i,t} = u_t^*$.

If $G_{i,k} = 0$, positivity of $\mathbf{M}_{i,t}$ implies $p_{i,t} = q_{i,t}$ at every position. Taking $\lambda_{i,k} = 0$ gives $\widehat{p}_{i,t} = q_{i,t} = p_{i,t}$.

A.2.3 Proof of the Minimum-KL Property

For any feasible candidate $\{u_t\}$, substitute Eq. (14) into the difference of the objectives,

$$F(\{u_t\}) - F(\{u_t^*\}) = \frac{1}{|\mathcal{I}_{i,k}|} \sum_{t \in \mathcal{I}_{i,k}} D_{\mathrm{KL}}(u_t \| u_t^*) + \frac{\lambda_{i,k}}{|\mathcal{I}_{i,k}|} \sum_{t \in \mathcal{I}_{i,k}} (u_t - u_t^*)^\top \mathbf{d}_{i,t} \geq 0. \tag{15}$$

The first sum is nonnegative by the KL inequality. The second is nonnegative because $\{u_t\}$ is feasible and $\{u_t^*\}$ satisfies the requirement with equality. Equality in the objective requires $u_t = u_t^*$ at every position.

A.2.4 Scalar Solution

For a candidate scale $\lambda \geq 0$, define the target distribution $Q_{i,t}(\lambda)$ and the mean unweighted correction $f_{i,k}(\lambda)$,

$$\begin{aligned} Q_{i,t}(\lambda) &= \mathrm{softmax}(\log q_{i,t} + \lambda \mathbf{d}_{i,t}), \\ f_{i,k}(\lambda) &= \frac{1}{|\mathcal{I}_{i,k}|} \sum_{t \in \mathcal{I}_{i,k}} \big(Q_{i,t}(\lambda) - q_{i,t}\big)^\top \mathbf{d}_{i,t}, \\ f_{i,k}(\lambda_{i,k}) &= G_{i,k}/w_{i,k}. \end{aligned} \tag{16}$$

For a distribution Q on a partition and a scalar function d of its elements, let $\mathrm{Var}_{C \sim Q}[d(C)]$ denote the variance of the scalar entries $d(C)$ when C is drawn from Q. Differentiation gives

$$f'_{i,k}(\lambda) = \frac{1}{|\mathcal{I}_{i,k}|} \sum_{t \in \mathcal{I}_{i,k}} \mathrm{Var}_{C \sim Q_{i,t}(\lambda)}[d_{i,t}(C)].$$

For $G_{i,k} > 0$, at least one direction has nonconstant entries. Indeed, a constant direction would have zero inner product with $p_{i,t} - q_{i,t}$, whereas the positive diagonal matrix gives a positive inner product whenever the distributions differ. At finite λ, $Q_{i,t}(\lambda)$ has positive entries, so $f'_{i,k}(\lambda) > 0$. Also $f_{i,k}(0) = 0$. The limit as λ grows is the largest achievable correction along these directions, which exceeds $G_{i,k}/w_{i,k}$ because the teacher is strictly feasible. Hence the positive root is finite and unique.

For fixed teacher and student predictions, increasing $w_{i,k}$ decreases the right-hand side and therefore decreases $\lambda_{i,k}$. The derivative of the mean $D_{\mathrm{KL}}(Q_{i,t}(\lambda) \| q_{i,t})$ is $\lambda f'_{i,k}(\lambda)$. The target's mean KL change consequently decreases as the weight increases.

The same scalar solution also exists for the comparison with $w_{i,k} = 1$. If $G_{i,k} > 0$, the teacher probabilities are positive and at least one direction is nonconstant. The teacher therefore gives an expected direction strictly below its maximum at that position. Summing across the decision gives $\lim_{\lambda \to \infty} f_{i,k}(\lambda) > G_{i,k}$. Strict monotonicity then yields a finite, unique root at $w_{i,k} = 1$ as well.

For $G_{i,k} > 0$, the finite root gives positive exponential targets. Together with the nonnegative multiplier $\mu_{i,k} = \lambda_{i,k}/w_{i,k}$, these targets satisfy normalization, stationarity, and complementary slackness. Convexity makes these conditions sufficient for the unique optimum.

We first bracket the root and then apply bisection. Solver inputs remain fixed throughout this computation.

A.2.5 Choice of the Response Weight

Normalized mean JSD provides a symmetric signal in $[0, 1]$ and averages over response positions without growing with response length. The map $w_k = 1 + \delta_k$ retains a unit baseline and caps the weight at two. It introduces no additional shape parameter. For the same predictions, a larger weight reduces the required unweighted correction G_k/w_k, hence the selected scale and target KL, while preserving the weighted projection onto the direct teacher gradient. Thus its role is to adjust target distance together with the loss weight.

The derivation does not require the linear map. Any fixed $w_k \geq 1$ admits the finite scalar solution above. More generally, for $G_k > 0$, define

$$H_k = \frac{1}{|\mathcal{I}_k|} \sum_{t \in \mathcal{I}_k} \left[\max_{C \in \mathcal{P}_t} d_t(C) - q_t^\top \mathbf{d}_t \right].$$

The finite solution exists whenever $w_k > G_k/H_k$. Positive teacher probabilities imply $H_k > G_k$, so $w_k \geq 1$ suffices. A positive weight below this threshold cannot meet the required contribution; equality at the threshold requires a limiting target on the boundary. Positivity alone is therefore not a sufficient condition for the finite exponential solution.

For fixed predictions, implicit differentiation gives

$$\frac{d\lambda_k}{dw_k} = -\frac{G_k}{w_k^2 f_k'(\lambda_k)}, \qquad \frac{dF(\{Q_t(\lambda_k)\})}{dw_k} = -\frac{\lambda_k G_k}{w_k^2}.$$

These identities describe sensitivity of the target construction. Table 2 compares the chosen map with $w_k = 1$ at 0.6B, supporting its use in the evaluated configuration. It does not compare alternative nonlinear maps or estimate their effect on accuracy. The α comparison changes the training coefficient, not the map from JSD to w_k. The three evaluated coefficients correspond to separately trained policies and their downstream evaluations. In contrast, λ_k is solved for each response. Varying this scale on recorded predictions changes targets without retraining a policy, so such calculations measure target geometry rather than sensitivity of task accuracy.

A.2.6 Teacher-Gradient Identity

Let $\mathbf{z}_{i,t}$ collect the full-vocabulary student logits at $h_{i,t}$. Write $\nabla_{\mathbf{z}_{i,t}}$ for derivatives with respect to this logit vector. Define the direct teacher gradient $\mathbf{g}_{i,t}^{\mathrm{T}}$ and the weighted target gradient $\mathbf{g}_{i,t}^{\mathrm{A}}$ at the update's starting parameters,

$$\begin{aligned} \mathbf{g}_{i,t}^{\mathrm{T}} &= \nabla_{\mathbf{z}_{i,t}} D_{\mathrm{KL}}(p_{i,t} \| q_{\theta,i,t}) \big|_{\theta=\theta^-}, \\ \mathbf{g}_{i,t}^{\mathrm{A}} &= \nabla_{\mathbf{z}_{i,t}} \left[w_{i,k} D_{\mathrm{KL}}(\widehat{p}_{i,t} \| q_{\theta,i,t}) \right] \big|_{\theta=\theta^-}. \end{aligned} \tag{17}$$

Here k is the decision containing position t. For vectors, $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product and $\| \cdot \|_2$ its norm. Eq. (8) gives

$$\begin{aligned} \langle \mathbf{g}_{i,t}^{\mathrm{A}}, \mathbf{g}_{i,t}^{\mathrm{T}} \rangle &= w_{i,k} (\widehat{p}_{i,t} - q_{i,t})^\top \mathbf{M}_{i,t} (p_{i,t} - q_{i,t}), \\ \| \mathbf{g}_{i,t}^{\mathrm{T}} \|_2^2 &= (p_{i,t} - q_{i,t})^\top \mathbf{d}_{i,t}. \end{aligned}$$

Let $\mathbf{g}_{i,k}^{\mathrm{T}}$ and $\mathbf{g}_{i,k}^{\mathrm{A}}$ concatenate the respective token gradients across decision k. Summing over the decision and applying the equality in Eq. (16) gives

$$(\mathbf{g}_{i,k}^{\mathrm{A}})^\top \mathbf{g}_{i,k}^{\mathrm{T}} = \| \mathbf{g}_{i,k}^{\mathrm{T}} \|_2^2. \tag{18}$$

This identity concerns the distillation gradients with respect to full-vocabulary logits, at fixed sampled histories and $\theta = \theta^-$. The teacher target in this comparison is the distribution on the shared partition.

A.2.7 Comparison with Direct Teacher Matching

Fix one decision with $n = |\mathcal{I}_k|$ and use the same histories, partition, and student predictions for both constructions. Let F denote Eq. (11); direct matching follows Eq. (3).

Proposition (comparison with $q_t + (p_t - q_t)/w_k$). For a fixed weight $w_k \geq 1$, define

$$u_t^{\mathrm{mix}} = \left(1 - \frac{1}{w_k}\right) q_t + \frac{1}{w_k} p_t. \tag{19}$$

This is a probability distribution satisfying the projection requirement with equality. Its weighted logit gradient $\mathbf{g}_t^{\mathrm{mix}}$ obeys

$$\begin{aligned} \frac{w_k}{n} \sum_{t \in \mathcal{I}_k} (u_t^{\mathrm{mix}} - q_t)^\top \mathbf{d}_t &= G_k, \\ \mathbf{g}_t^{\mathrm{mix}} = w_k \mathbf{R}_t (q_t - u_t^{\mathrm{mix}}) &= \mathbf{R}_t (q_t - p_t) = \mathbf{g}_t^{\mathrm{T}}. \end{aligned} \tag{20}$$

Thus $q_t + (p_t - q_t)/w_k$ and direct matching give the same parameter gradient and one identical update when the batch, remaining objective, clipping, and optimizer state agree.

For the minimum-KL targets $\widehat{p}_t$, optimality and convexity give

$$F(\{\widehat{p}_t\}) \le F(\{u_t^{\mathrm{mix}}\}) \le \frac{1}{w_k} F(\{p_t\}). \tag{21}$$

More precisely, since both targets attain the same projection, Eq. (15) yields

$$F(\{u_t^{\mathrm{mix}}\}) - F(\{\widehat{p}_t\}) = \frac{1}{n} \sum_{t \in \mathcal{I}_k} D_{\mathrm{KL}}(u_t^{\mathrm{mix}} \| \widehat{p}_t). \tag{22}$$

Thus the KL improvement is strict whenever the targets differ at any position. Define $\mathbf{e}_k = \mathbf{g}_k^{\mathrm{A}} - \mathbf{g}_k^{\mathrm{T}}$. Eq. (18) gives

$$\begin{aligned} \langle \mathbf{e}_k, \mathbf{g}_k^{\mathrm{T}} \rangle &= 0, \\ \|\mathbf{g}_k^{\mathrm{A}}\|_2^2 &= \|\mathbf{g}_k^{\mathrm{T}}\|_2^2 + \|\mathbf{e}_k\|_2^2. \end{aligned} \tag{23}$$

Substitution, KL convexity, and norm expansion establish these identities. Minimum target KL and minimum gradient norm are distinct properties.

When the update differs. The target $q_t + (p_t - q_t)/w_k$ coincides with the exponential solution exactly when a common $\lambda_k \ge 0$ and constants b_t that may vary across positions satisfy, for every partition element C,

$$\log\left(1 + \frac{p_t(C)/q_t(C) - 1}{w_k}\right) = \lambda_k d_t(C) - b_t. \tag{24}$$

Normalization gives $b_t = \log Z_t(\lambda_k)$. Since $\mathbf{R}_t^\top \mathbf{R}_t$ is positive definite, different targets give different logit gradients; $p_t = q_t$ throughout gives zero gradients.

Meaning of the geometry. For independently adjustable logits, $-\mathbf{g}_k^{\mathrm{A}}$ and $-\mathbf{g}_k^{\mathrm{T}}$ decrease the direct teacher loss at the same first-order rate $\|\mathbf{g}_k^{\mathrm{T}}\|_2^2$. Equation (22) gives the exact reduction in target KL.

For shared parameters, let $\mathbf{J}_k = \partial \mathbf{z}_k / \partial \theta$ be the Jacobian of concatenated logits at the fixed histories and starting parameters. The parameter-gradient inner product is

$$\langle \mathbf{J}_k^\top \mathbf{g}_k^{\mathrm{A}}, \mathbf{J}_k^\top \mathbf{g}_k^{\mathrm{T}} \rangle = (\mathbf{g}_k^{\mathrm{A}})^\top \mathbf{J}_k \mathbf{J}_k^\top \mathbf{g}_k^{\mathrm{T}}. \tag{25}$$

The identity concerns logits before transformation by the model Jacobian and optimizer. The training comparisons evaluate the resulting policies.

A.2.8 Allocation within a Complete Response

For this analysis, keep the same weight w_k when comparing grouping choices. Write $\mathbf{S}_t = \mathrm{diag}(q_t) - q_t q_t^\top$ and $V_t = \mathbf{d}_t^\top \mathbf{S}_t \mathbf{d}_t$, the variance of the direction entries under q_t. A local expansion of the exponential target gives

$$\begin{aligned} Q_t(\lambda) - q_t &= \lambda \mathbf{S}_t \mathbf{d}_t + O(\lambda^2), \\ (Q_t(\lambda) - q_t)^\top \mathbf{d}_t &= \lambda V_t + O(\lambda^2), \\ D_{\mathrm{KL}}(Q_t(\lambda) \| q_t) &= \tfrac{1}{2} \lambda^2 V_t + O(\lambda^3). \end{aligned} \tag{26}$$

For small tilts, larger V_t supplies more projection. The correction is proportional to $q_t(C)[d_t(C) - \sum_D q_t(D) d_t(D)]$ over the shared partition, incorporating student confidence and the teacher direction.

More generally, let $a_t = (Q_t(\lambda) - q_t)^\top \mathbf{d}_t$ denote a position's supplied projection. Along this family, the derivative of its KL cost with respect to a_t is λ whenever the direction is nonconstant. A shared scale equalizes these marginal costs across positions. Requiring each position to supply its own teacher reference divided by w_k imposes additional restrictions. Such targets are feasible for the aggregate requirement, so TrajPD has no larger mean KL, with strict improvement when the separate optima require different scales.

Two-position example. Consider an illustrative binary partition with $\mathbf{M}_1 = \mathbf{M}_2 = \mathbf{I}$, student probabilities $q_1 = (1/2, 1/2)$ and $q_2 = (9/10, 1/10)$, and teacher probabilities $p_t = q_t + (\epsilon, -\epsilon)$ for $0 < \epsilon < 1/10$. Here $\mathbf{I}$ is the identity matrix and ϵ controls the discrepancy. Both positions have direction $(\epsilon, -\epsilon)$ and teacher reference $2\epsilon^2$, but $V_1 = \epsilon^2$ and $V_2 = 9\epsilon^2/25$. As $\epsilon \to 0$, the shared solution has first-coordinate changes

$$\begin{aligned} \widehat{p}_1(1) - q_1(1) &= \frac{25}{17}\frac{\epsilon}{w_k} + O(\epsilon^2), \\ \widehat{p}_2(1) - q_2(1) &= \frac{9}{17}\frac{\epsilon}{w_k} + O(\epsilon^2). \end{aligned} \tag{27}$$

Requiring a separate reference at each position gives ϵ/w_k at both. TrajPD instead assigns more correction to the less certain first prediction, reducing total KL while retaining the total projection.

A response groups predictions forming one action before the next observation. The objective averages conditional KLs on sampled prefixes; it does not optimize a full-trajectory KL or assign future outcomes to earlier decisions.

Finally, $w_k = 1 + \delta_k$ sets the unweighted reference to G_k/w_k. It changes the targets along the exponential family while the weighted projection remains fixed.

A.3 Student Update

A.3.1 Token Averaging and Task Scores

For any token quantity $f(i,t)$ in a batch of $B \geq 2$ trajectories, define its global action-token average

$$\widehat{\mathbb{E}}_{\mathrm{tok}}[f] = \frac{1}{Z_{\mathrm{act}}} \sum_{i=1}^{B} \sum_{t \in \mathcal{T}_i} f(i,t). \tag{28}$$

The leave-one-out advantage A_i, task-score objective $\mathcal{L}_{\mathrm{task}}$, and complete objective $\mathcal{L}_{\mathrm{total}}$ are defined in Section 4.4 and Eq. (7). Substituting Eq. (28) expresses both objectives using the same global average.

Verifier scores include partial credit. Indexed sample occurrences remain distinct even when their contents coincide, and advantages are computed before data-parallel partitioning.

The fixed action-token average includes sampled end-of-sequence tokens and excludes prompts, observations, forced control tokens, and padding. Trajectories use nucleus sampling with top-p 0.95, and sampled-token log probabilities use the original full-vocabulary model distribution. The gradient below is therefore taken with respect to the stated sampled objective.

A.3.2 Parameter Gradient and Update

Let ∇_θ denote differentiation with respect to student parameters. For fixed targets, the target-entropy term has zero derivative. Consequently,

$$\begin{aligned} \nabla_\theta D_{\mathrm{KL}}(\widehat{p}_{i,t} \| q_{\theta,i,t}) &= -\sum_{C \in \mathcal{P}_{i,t}} \widehat{p}_{i,t}(C) \nabla_\theta \log q_{\theta,i,t}(C), \\ \nabla_\theta \mathcal{L}_{\mathrm{task}} &= -\widehat{\mathbb{E}}_{\mathrm{tok}}[A_i \nabla_\theta \log \pi_\theta(y_{i,t} \mid h_{i,t})]. \end{aligned} \tag{29}$$

Define the target-weighted parameter score as $\mathbf{s}_{i,t} = \sum_{C \in \mathcal{P}_{i,t}} \widehat{p}_{i,t}(C) \nabla_\theta \log q_{\theta,i,t}(C)$. Differentiating Eq. (7) gives

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\mathrm{total}} = -\widehat{\mathbb{E}}_{\mathrm{tok}}\big[& A_i \nabla_\theta \log \pi_\theta(y_{i,t} \mid h_{i,t}) \\ & + \alpha w_{i,k_i(t)} \mathbf{s}_{i,t} \big]. \end{aligned} \tag{30}$$

Shared parameters combine these token contributions before gradient clipping and the AdamW update in Algorithm 1.

B Experimental Settings and Training Procedure

The prepared training split contains 28,493 tasks. The initialization data have 2,859 training examples and 128 validation examples. The evaluation splits contain 30 AIME24, 30 AIME25, 198 GPQA-Diamond, and 175 LiveCodeBench v6 questions. Table 3 lists the settings for the main TrajPD experiments. Algorithm 1 states the order used to collect responses, obtain supervision, construct targets, and update the policy.

Table 3: Core settings for the main TrajPD runs.

Configuration	Value
Training temperature	1.0
Global batch size	128
Mini batch size	128
Rollouts per prompt	1
Teacher log-probability top-k	16
Training top-p	0.95
Maximum prompt length	2,560
Maximum response sequence length	24,576
Peak learning rate	3×10^{-6}
Optimizer	AdamW
Learning-rate schedule	Cosine
Training updates	100
Distillation coefficient α	1.0

The agent evaluations use temperature 1.0, top-p 0.6, and eight responses per question. The response sequence limit is the framework's sequence budget. Generation is capped at 16,384 new tokens in total, with 8,192 per turn and 4,096 in the final-answer phase. Tool observations and protocol messages also occupy sequence positions. Teacher top-k specifies the number of returned log probabilities; sampling top-k is disabled.

Algorithm 1 TrajPD training

Require: Student π_θ, frozen teacher π_{ϕ^*}, task distribution $\mathcal{D}$, batch size $B \geq 2$, coefficient $\alpha > 0$

for each training iteration **do**
  Set $\theta^- \leftarrow \theta$ and sample B tasks from $\mathcal{D}$
  Collect one trajectory per task using π_{θ^-} and the training sampling procedure
  Record response boundaries and sampled student positions, excluding tool observations and forced tokens
  Obtain verifier scores r_i and teacher probabilities at the sampled contexts
  Form the shared partitions and compute $p_{i,t}$, $q_{i,t}$, and $\mathbf{M}_{i,t}$
  Compute $A_i \leftarrow r_i - \frac{1}{B-1} \sum_{j \neq i} r_j$ over the complete batch
  for each decision k in each trajectory i **do**
    Compute $\mathbf{d}_{i,t}$, $G_{i,k}$, and $w_{i,k}$
    if $G_{i,k} = 0$ **then**
      Set $\lambda_{i,k} \leftarrow 0$
    else
      Bracket and solve Eq. (6) with equality for $\lambda_{i,k}$
    end if
    Set $\widehat{p}_{i,t} \leftarrow Q_{i,t}(\lambda_{i,k})$ for $t \in \mathcal{I}_{i,k}$
  end for
  Hold histories, partitions, targets, weights, and A_i fixed during differentiation
  Take one AdamW step using Eq. (7), normalized by Z_{act}
end for
return Updated student π_θ

C Additional Experimental Results

C.1 Training Comparison

Figure 5 compares recorded task returns and EOS termination. Task returns use the saved trajectory scores. EOS completion is recomputed from 30,400 responses using the last generated token under the response mask; it differs from the protocol completion used in Section 5.4. KL compares teacher and TrajPD targets on the same TrajPD predictions. Seven-step averages use available points without extrapolation.

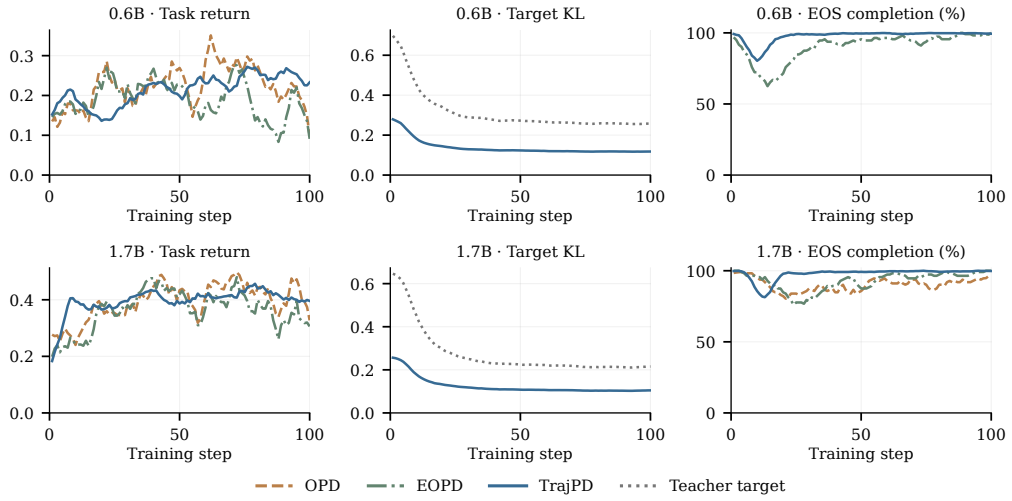

Figure 5: Training dynamics at 0.6B (top) and 1.7B (bottom). Left: task return for OPD, EOPD, and TrajPD, using the recorded trajectory returns. Middle: KL for TrajPD targets and direct teacher targets on TrajPD training predictions. Right: the fraction of responses whose last generated token is EOS, recomputed using the same definition for each available run. OPD 0.6B is omitted from this panel because its token records were not retained. Curves use centered seven-step moving averages.

C.2 Sampling Results Supplementing the Main Table

For c_j correct answers among eight responses, Avg@8 is $100J^{-1}\sum_j c_j/8$. Table 1 retains the original Pass@8, estimated by 1,000 resamples with replacement (seed 42). Figure 6 instead uses the exact expectation

$$R_k = \frac{100}{J}\sum_{j=1}^{J}\left[1-\left(1-\frac{c_j}{8}\right)^k\right], \qquad k \in \{1,2,4,8\}. \tag{31}$$

These counts reuse eight recorded responses. Sampling without replacement gives $100J^{-1}\sum_j[1-\binom{8-c_j}{k}/\binom{8}{k}]$, a different quantity.

C.3 Training Measurements for the Component and Coefficient Comparisons

Figure 7 reports actual training measurements for Table 2. Panels (a)–(d) compare the component settings; panels (e)–(f) show task return and the solved scale for $\alpha = 0.1, 0.5$, and 1. Direct teacher targets give equal teacher and target KL; uniform weights remain one. In Table 2, $\bar{\lambda}$ averages the logged scale over updates 91–100. Within each update, each response scale is weighted by the number of its generated positions, following the training logger.

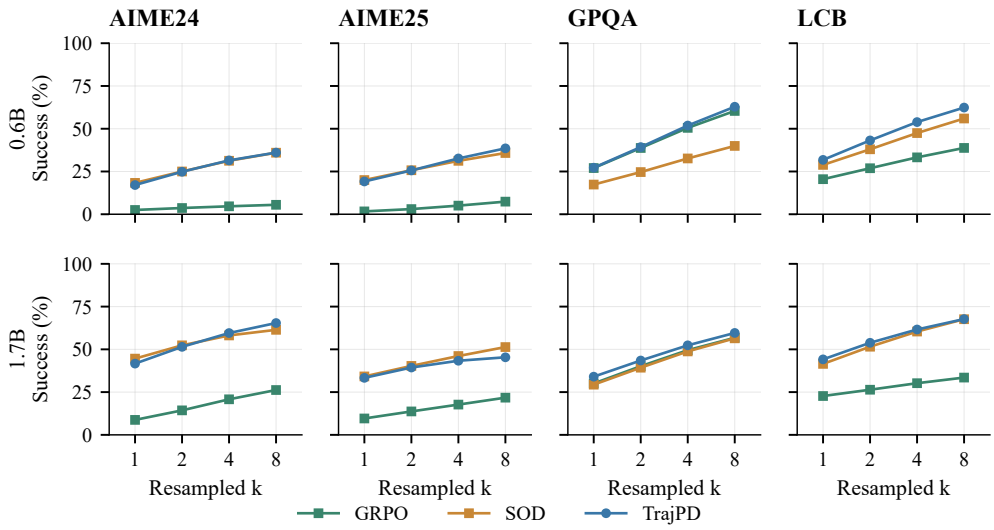

Figure 6: Success across resampling counts for GRPO, SOD, and TrajPD. Markers show the exact success probability at each count; lines connect those values. All methods use sampling with replacement from eight recorded responses per question. Rows correspond to model size; columns correspond to benchmarks.

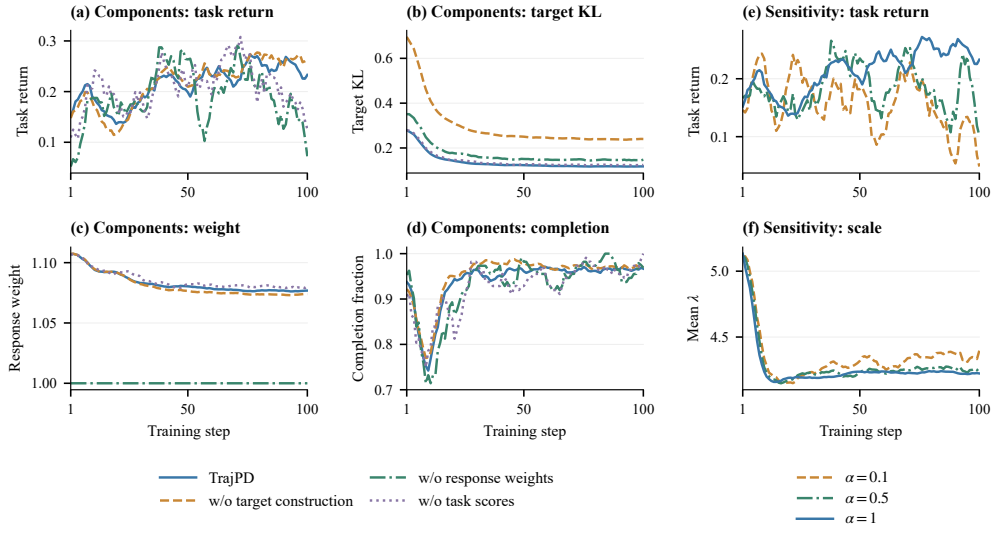

Figure 7: Recorded measurements for the 0.6B component and coefficient experiments. Panels (a)–(d) compare components; panels (e)–(f) compare coefficients. The two groups use separate legends and share the training-step axis.

Table 4: Component variants and coefficient sensitivity at 0.6B. Scores are Avg@8 (%); Mean gives equal weight to the four benchmarks. Bold indicates the best performance and underline indicates second-best.

Variant	AIME24	AIME25	GPQA	LCB	Mean
TrajPD ($\alpha = 1$)	<u>17.08</u>	**19.17**	27.21	**31.79**	**23.81**
w/o target construction	**18.75**	<u>18.33</u>	25.00	26.57	<u>22.16</u>
w/o response weights	11.67	15.00	<u>28.35</u>	26.64	20.41
w/o task scores	9.58	14.58	**28.47**	28.86	20.37
$\alpha = 0.1$	5.00	12.50	26.52	<u>31.43</u>	18.86
$\alpha = 0.5$	14.17	14.17	25.13	27.29	20.19

C.4 Scale Verification on Recorded Predictions

Table 5 varies the solved scale on the same 25 recorded responses (19 at 0.6B and six at 1.7B). Multipliers 0.5, 1, and 2 change the prescribed contribution and target KL; the solved value satisfies the equality. This calculation reuses predictions and does not train new policies.

Table 5: Sensitivity to the solved scale on recorded AIME responses (19 at 0.6B; six at 1.7B). Values are medians. Bold marks the solved scale, which preserves the prescribed contribution.

Size	Multiplier	λ	Contribution ratio	Target KL	Target entropy
0.6B	0.5	2.110	0.546	0.032	0.807
0.6B	1	**4.221**	**1.000**	**0.113**	**0.775**
0.6B	2	8.441	1.490	0.281	0.663
1.7B	0.5	2.179	0.537	0.038	1.189
1.7B	1	**4.358**	**1.000**	**0.134**	**1.111**
1.7B	2	8.715	1.533	0.353	0.916

D AIME Behavior and Case Analysis

D.1 Population and Annotation Procedure

The population comprises all 960 TrajPD responses across AIME24/25 and both sizes. The grader supplies correctness and answer parsing; trajectories supply termination, length, and executed calls. Parsed and missing answers partition incorrect responses; earlier turn cutoffs can overlap either state. Cases are selected for contrasting outcomes on the same question and independently checked. They illustrate errors without estimating their frequencies.

Table 6: Recorded states among incorrect TrajPD AIME responses. The last three columns can overlap with answer states and with one another.

Size	Dataset	Incorrect	Parsed	Missing	Completed	Turn cutoff	No tool
0.6B	AIME24	199	143	56	143	171	145
0.6B	AIME25	194	140	54	143	166	122
1.7B	AIME24	140	108	32	109	118	86
1.7B	AIME25	160	131	29	132	130	92

At 0.6B on AIME24, TrajPD has more correct responses than SOD on five questions, equal counts on 21, and fewer on four. Figure 8 reports the distribution of correct counts for GRPO, SOD, and TrajPD using all 30 questions per benchmark.

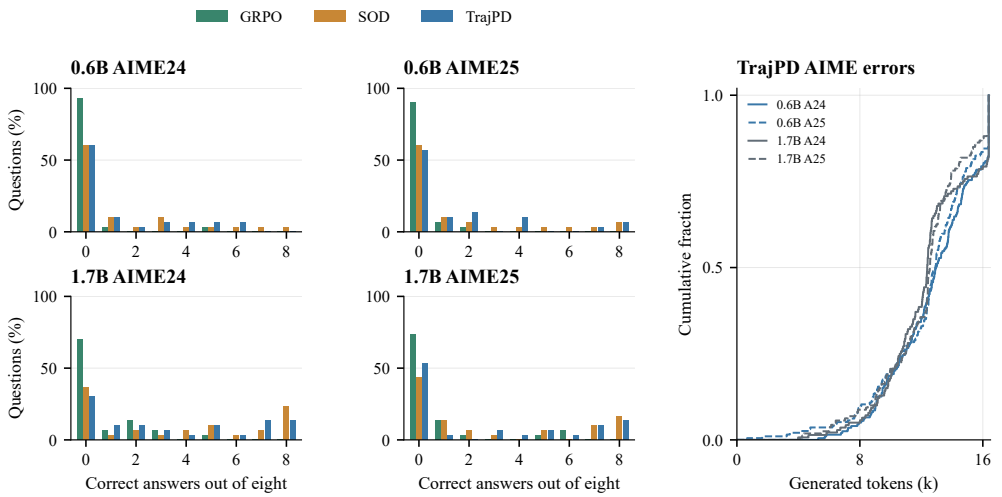

Figure 8: AIME answer counts and response lengths. The four bar panels report the percentage of questions with each correct count out of eight responses, using all 30 questions for each benchmark and size. Colors identify GRPO, SOD, and TrajPD. The right panel shows length distributions among incorrect TrajPD responses for both AIME years and sizes.

D.2 Domain Summary

Table 7 reports the domain summary alongside the population measurements above.

Table 7: Domain-level target geometry and evaluation behavior. Math pools AIME 24 and AIME 25. KL reduction is computed from the mean joint and token-wise KL on the analyzed decisions; the remaining columns use evaluation responses. Changes are TrajPD minus SOD in percentage points. The final column reports SOD → TrajPD correct rates among normally completed responses.

Domain	Size	Decisions	KL reduction	ΔAvg@8	ΔCompletion	Correct|complete
Math	0.6B	22	16.5%	−1.0	+30.2	40.4 → 23.3
Math	1.7B	27	16.6%	−1.9	+9.0	50.0 → 42.8
Science	0.6B	8	20.5%	+9.9	+38.6	30.5 → 28.8
Science	1.7B	12	18.7%	+4.7	+14.0	35.0 → 34.8
Code	0.6B	12	18.2%	+2.9	+26.4	48.9 → 33.2
Code	1.7B	10	17.0%	+2.6	+7.0	48.9 → 43.9

D.3 AIME24: A Question Selected from the Component Comparison

On AIME24 question 0, TrajPD, direct teacher targets, and SOD obtain 3/8, 4/8, and 8/8 correct answers. TrajPD response `aime2024_000000_001` uses $45/8$ and submits 53 without a call. Setting $X = \log_2 x$, $Y = \log_2 y$, $Z = \log_2 z$ gives $X - Y - Z = 1/2$, $Y - X - Z = 1/3$, $Z - X - Y = 1/4$. The solution $(-7/24, -3/8, -5/12)$ yields $|4X + 3Y + 2Z| = 25/8$ and answer 33, also obtained by direct teacher response `aime2024_000000_001`. Table 8 retains every sample.

Among 693 incorrect responses, 522 have parsed answers, 585 have an earlier turn cutoff, and 445 execute no call; these conditions overlap. The checked cases identify an invalid restriction, a mistaken game classification, and an arithmetic error. Evaluation records have no logged training weights or scales. The comparison of response and position requirements concerns target KL; Table 2 assesses the complete construction after training.

Table 8: Every recorded sample for AIME24 question index 0 at 0.6B. A cell gives the submitted answer; ✓ denotes correctness and – denotes no parsed answer. Sample indices identify records within each method, rather than matched random draws.

Method	0	1	2	3	4	5	6	7
TrajPD	33✓	53	33✓	31	37	33✓	13	71
Direct teacher	33✓	33✓	33✓	–	33✓	–	109	37
SOD	33✓	33✓	33✓	33✓	33✓	33✓	33✓	33✓

D.4 Recorded GPQA Pair

Table 9: Two TrajPD 1.7B responses to the same GPQA question. Both normally complete in two assistant turns. The reference answer is D.

Evidence step	Reference-matching response	Contrasting response
Numerical basis	Tool width sum: 7.24e-7 eV	States 1e-8 exceeds 6.585e-7 eV
Option choice	D (1e-4 eV)	B (1e-8 eV)
Submitted answer	D, matching the reference	B

E Training–Evaluation Overlap Audit

We screen 433 evaluation prompts against released Open-AgentRL SFT-3K/30K and the prepared training splits. After normalizing Unicode, case, whitespace, and common fraction commands, we check exact matches and inspect pairs with a character five-gram cosine similarity of at least 0.75.

Table 10: Prompt-level overlap screen. "Prepared" combines the actual teacher and student training splits; "released" combines the two upstream files. Flagged pairs have a character five-gram cosine similarity of at least 0.75. "Confirmed" counts problems manually verified as identical among the flagged pairs. Counts are evaluation problems, not training records.

		Prepared training splits			Released files	
Benchmark	Eval.	Exact	Flagged	Confirmed	Flagged	Confirmed
AIME 24	30	0	0	0	0	0
AIME 25	30	0	1	0	1	0
GPQA	198	0	1	0	1	0
LCB	175	0	0	0	8	8

The flagged AIME25 and GPQA pairs are different problems. Eight LCB prompts match the released 30K file; none occurs in the prepared teacher GRPO or student distillation splits. Table 10 reports this local training-data screen.

F Training Overview

Figure 9 retains the 100 training updates at both sizes. Bold curves use centered seven-update averages. The final panel reports the maximum relative discrepancy from G_k across all updates.

G Complete Evaluation and Behavior Measurements

G.1 Confidence Intervals on All Benchmarks

Figure 10 shows the five methods with complete results, with one panel per size. Intervals use 10,000 question resamples (seed 42), retaining eight responses per question.

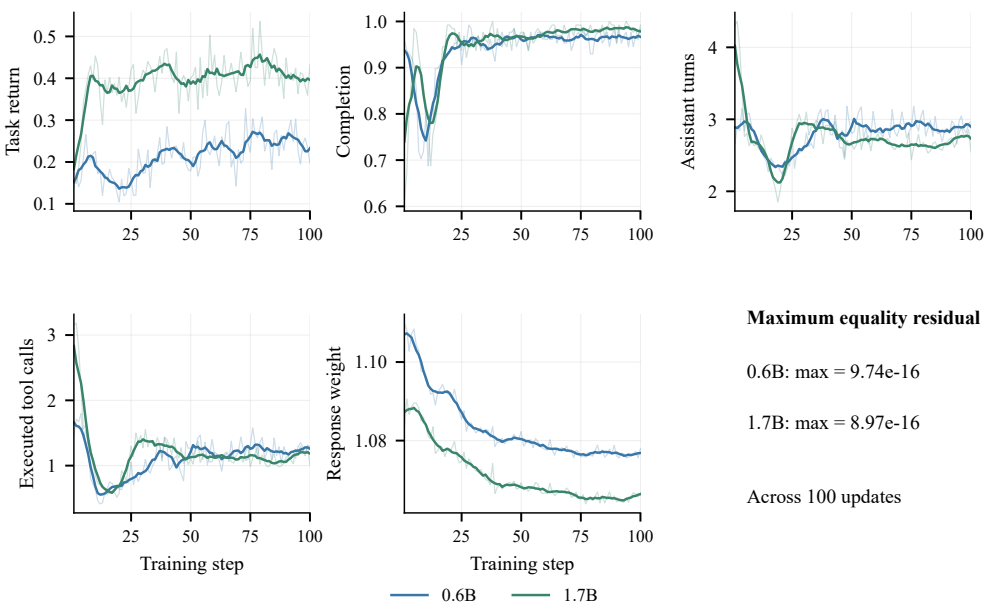

Figure 9: TrajPD training measurements for Qwen3-0.6B and Qwen3-1.7B.

G.2 Uncertainty of Performance Comparisons

Relative to SOD, the GPQA gains at 0.6B and 1.7B have paired 95% intervals of [6.63, 13.13] and [1.64, 7.70] percentage points. The LiveCodeBench intervals are $[-0.50, 6.43]$ and $[-0.86, 6.29]$; the AIME intervals also include zero. The four-benchmark mean gains are 2.67 and 0.89 points, with intervals [0.59, 4.60] and $[-1.77, 3.34]$. These intervals quantify variation across evaluation questions for the recorded policies, not independent training runs.

The comparisons reported in the main performance discussion use the same questions across policies. The macro mean weights benchmarks equally, with questions resampled independently within each benchmark. Intervals describe evaluation uncertainty for recorded policies, not variation across training runs.

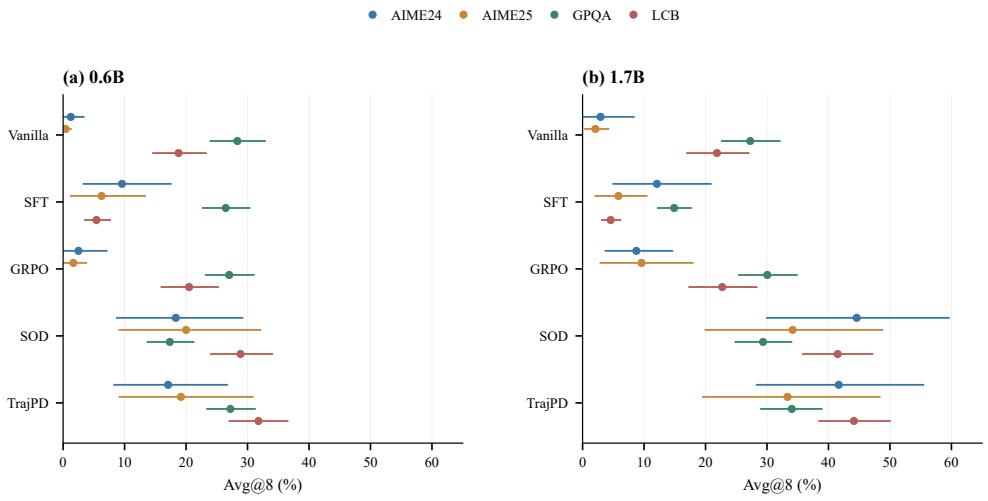

Figure 10: Avg@8 and evaluation uncertainty for Vanilla, SFT, GRPO, SOD, and TrajPD. Panels show 0.6B and 1.7B separately. Points report each method's score; horizontal segments denote 95% intervals, and colors identify benchmarks.

G.3 Lengths and Tool Use

Figure 11 compares TrajPD, SOD, GRPO, and the 4B teacher. Figure 12 compares TrajPD, SOD, GRPO, and the teacher, grouping executed calls into 0, 1, 2, and 3+. Both figures use every response from the methods shown in the retained evaluations. The same teacher evaluation is shown in both size rows as a reference. Tool counts include executed calls only.

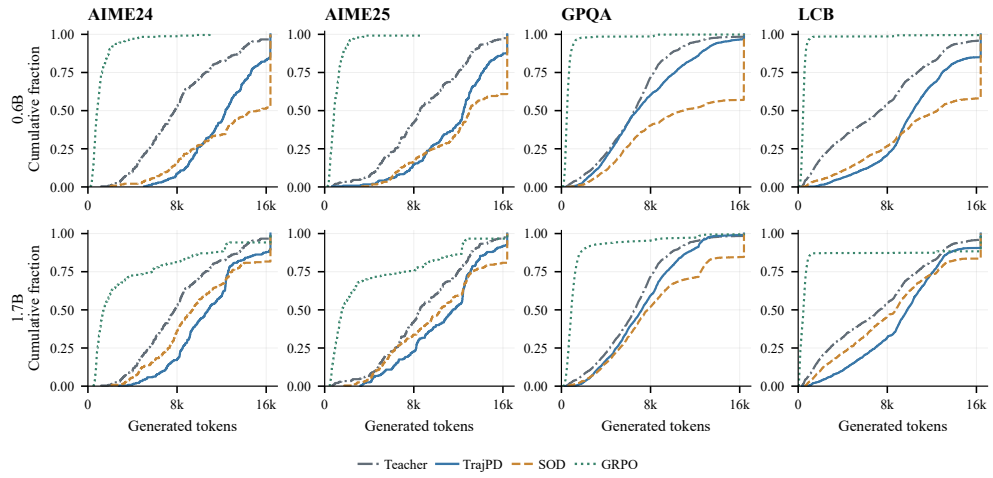

Figure 11: Empirical response-length distributions on all four benchmarks.

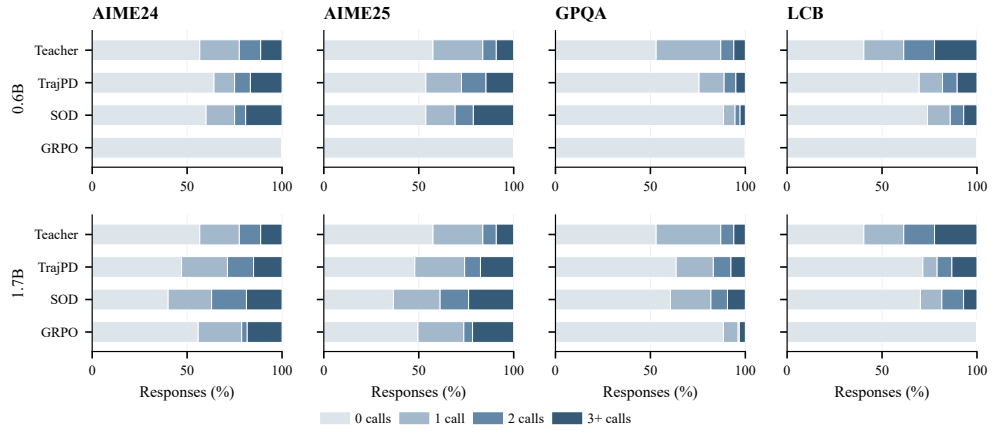

Figure 12: Distribution of executed tool calls. Each bar sums to 100% and divides responses into 0, 1, 2, and 3+ calls. Rows compare the teacher, TrajPD, SOD, and GRPO.

H Evaluation Prompt Templates

The following templates show the evaluation instructions supplied with the benchmark problems. Bracketed fields are replaced by the corresponding problem content.

H.1 AIME 2024 AND AIME 2025

> Analyze and solve the following problem step by step. You may use the Python tool for calculation or verification. When the answer is final, stop calling tools and write it as \boxed{answer}. Do not put units inside the box.
>
> [problem]

Figure 13: Evaluation prompt template for AIME 24 and AIME 25.

H.2 GPQA

> Analyze and solve the following [domain] problem step by step.
>
> [benchmark problem]
>
> The tool could be used for more precise and efficient calculation and could help you to verify your result before you reach the final answer.
>
> **Note: You should first analyze the problem and form a high-level solution strategy, then utilize the tools to help you solve the problem.**
>
> Remember once you make sure the current answer is your final answer, do not call the tools again and directly output the final answer in the following text format, the answer format must be: \boxed{'The final answer goes here.'}.
>
> Here you need to put the final uppercase letter option of this problem into \boxed{}

Figure 14: Evaluation prompt template for GPQA.

H.3 LIVECODEBENCH

> You will be given a question (problem specification) and will generate a correct Python program that matches the specification and passes all tests.
>
> [problem specification]
>
> Here are some input and output examples of the expected code:
>
> Input:[public input list]
>
> Output: [public output list]
>
> **Note: You should first analyze the problem and form a high-level solution strategy, then utilize the tools to help you solve the problem.**
>
> Read the inputs from stdin solve the problem and write the answer to stdout (do not directly test on the sample inputs). Enclose your code within delimiters as follows. Ensure that when the python program runs, it reads the inputs, runs the algorithm and writes output to STDOUT.
>
> ```python
> # YOUR CODE HERE
> ```
>
> Before sumbit your code, you can utilize tools to check the correctness of your code, once you make sure the current code is correct, do not call the tools again and submit your code within ```python

Figure 15: Evaluation prompt template for LCB.

I CASE STUDY AND ADDITIONAL EXAMPLES

We present paired correct and incorrect responses from Qwen3-1.7B for one question in each benchmark. Correct examples are selected to illustrate a verifiable solution. For the same question, we select an incorrect, normally completed response nearest the median generated length among eligible incorrect responses, excluding infrastructure failures. These illustrative pairs do not estimate the frequency of a behavior. Tables separate assistant turns from tool observations. Trace entries are abbreviated summaries of the recorded content, with numerical observations reproduced from the record. Analysis paragraphs interpret each solution using its reference answer.

I.1 AIME 2024

Alice and Bob alternately remove one or four tokens from a stack, with Alice moving first and the last move winning. The task counts initial stack sizes from 1 through 2024 for which Bob can guarantee a win.

Correct response

Stage	Student trajectory and observation
Assistant turns 1–2	Formulate winning positions using legal removals of one or four tokens. Submit a recurrence-based count for stacks up to 2024.
Tool observation	809.
Assistant turn 3	Submit $\boxed{809}$, matching the reference.

Analysis of the correct response. The student evaluates the winning and losing positions with a tool and receives 809, then submits $\boxed{809}$. The interaction contains three assistant turns, one tool call, and 8,725 generated tokens. To verify the result, a losing position is one from which every legal move reaches a winning position. Starting from the empty stack, this gives positive losing positions congruent to 0 or 2 modulo 5. Their counts are 404 and 405, respectively, summing to the reference answer 809.

Incorrect response

Stage	Student trajectory and observation
Recorded reasoning	Infer a pattern of losing positions that includes 8.
Tool observation	No executed tool call in this trajectory.
Final submission	$\boxed{1012}$; reference answer 809.

Analysis of the incorrect response. The paired response submits $\boxed{1012}$ after three assistant turns and 12,667 generated tokens, with no executed tool call. Its proposed pattern includes 8 among the losing positions. At 8, however, removing one token reaches the losing position 7. The discrepancy is specific to the inferred position pattern in this response. The recorded reference is 809.

I.2 AIME 2025

The task counts integer pairs (x, y) with $-100 \leq x, y \leq 100$ satisfying $12x^2 - xy - 6y^2 = 0$.

Correct response

Stage	Student trajectory and observation
Assistant turn 1	Enumerate integer pairs in the specified square and count those satisfying $12x^2 - xy - 6y^2 = 0$.
Tool observation	117.
Assistant turn 2	Submit $\boxed{117}$, matching the reference.

Analysis of the correct response. The student enumerates the integer pairs within the stated bounds. The tool returns 117, and the student submits $\boxed{117}$. The trajectory contains two assistant turns, one tool call, and 6,419 generated tokens. An independent algebraic check factors the equation as

$$(3x + 2y)(4x - 3y) = 0.$$

The first branch has $(x, y) = (2t, -3t)$ for integers $-33 \le t \le 33$, and the second has $(x, y) = (3t, 4t)$ for integers $-25 \le t \le 25$. Their union contains $67 + 51 - 1 = 117$ pairs, counting $(0, 0)$ once.

Incorrect response

Stage	Student trajectory and observation
Assistant turn 1	Count nonzero values of y divisible by 12, assigning two pairs to each, and add the origin.
Tool observation	33.
Assistant turn 2	Submit $\boxed{33}$; reference answer 117.

Analysis of the incorrect response. The paired response tests values of y divisible by 12, adds two solutions for each nonzero such value, and counts the origin once. The tool returns 33, which is submitted as $\boxed{33}$. This trajectory contains two assistant turns, one tool call, and 9,184 generated tokens. The branch conditions should be counted separately before taking their union. This pair illustrates why the expression supplied to a tool matters as well as the numerical feedback.

I.3 GPQA

The question gives two energy-level lifetimes of 10^{-9} and 10^{-8} seconds and asks which proposed energy separation allows the levels to be distinguished. The options are A, 10^{-11} eV; B, 10^{-8} eV; C, 10^{-9} eV; and D, 10^{-4} eV.

Correct response

Stage	Student trajectory and observation
Assistant turn 1	Calculate both energy widths and their sum using the given lifetimes.
Tool observation	Widths 6.5821×10^{-7} and 6.5821×10^{-8} eV; sum 7.2403×10^{-7} eV.
Assistant turn 2	Compare candidate separations and submit $\boxed{D}$, matching the reference.

Analysis of the correct response. The correct response uses the reduced Planck constant $\hbar = 1.0545718 \times 10^{-34}$ joule seconds and the conversion $1\,\mathrm{eV} = 1.602176634 \times 10^{-19}$ joules. Dividing $\hbar$ by each lifetime and converting to eV yields the tool observations

$$\begin{aligned} \text{first energy width} &\approx 6.582119459370421 \times 10^{-7}\ \mathrm{eV}, \\ \text{second energy width} &\approx 6.582119459370420 \times 10^{-8}\ \mathrm{eV}, \\ \text{sum} &\approx 7.240331405307463 \times 10^{-7}\ \mathrm{eV}. \end{aligned}$$

The student compares the options with these values and submits D, matching the reference. There are two assistant turns, one tool call, and 8,409 generated tokens.

Incorrect response

Stage	Student trajectory and observation
Recorded reasoning	Write the comparison $10^{-8} > 6.585 \times 10^{-7}$ between energy values in eV.
Tool observation	No executed tool call in this trajectory.
Final submission	B; reference answer D.

Analysis of the incorrect response. The paired response submits B after two assistant turns and $10{,}004$ generated tokens, with no tool call. It writes the comparison $10^{-8}\,\mathrm{eV} > 6.585 \times 10^{-7}\,\mathrm{eV}$, reversing the numerical ordering. The correct response instead bases its choice on the explicitly returned widths. This is a comparison of two recorded solutions to the same question, not a causal estimate of tool-use effects.

I.4 LIVECODEBENCH

The Properties Graph task treats each integer array as a vertex. Two vertices share an edge when the number of distinct common values is at least the given threshold. The output is the number of connected components. The prepared public example uses arrays $[1, 2]$, $[1, 1]$, $[3, 4]$, $[4, 5]$, $[5, 6]$, and $[7, 7]$, with threshold 1 and reference output 3.

Correct response

Stage	Student trajectory and observation
Assistant turn 1	Construct the intersection graph and count its connected components. Submit the candidate solution to the tool.
Tool observation	0, returned by the empty-input branch.
Assistant turn 2	Submit a solution reading the collection and threshold from separate input lines.
Separate evaluation	Pass the retained public test. An additional 206 constructed verification cases agree with a reference implementation.

Analysis of the correct response. The submitted solution reads the array collection and threshold from separate input lines, converts each array to a set, builds edges by set intersection, and counts components by graph traversal. It passes the retained public test. A separate verification of the submitted solution on 206 constructed cases also agrees with a reference implementation. The recorded interaction has two assistant turns, one tool call, and $3{,}240$ generated tokens. The tool observation is 0 from an empty-input branch; the public-test outcome is the separate evaluation of the final submitted solution.

Incorrect response

Stage	Student trajectory and observation
Assistant turn 1	Submit the graph traversal solution with comma-based parsing of the complete input.
Tool observation	No executed tool call in this trajectory.
Separate evaluation	The submitted solution outputs 0 on the retained public example; expected output 3.

Analysis of the incorrect response. The paired response also constructs edges and traverses components, but parses the complete input by splitting it at commas. For the prepared example, the first resulting fragment is an incomplete array expression, so its exception branch outputs 0 instead of 3. The response has one assistant turn, no tool call, and $4{,}593$ generated tokens. The recorded correctness score applies to the submitted program, including its input handling.