BRANCHING ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) provides teacher supervision on student-generated trajectories, but a single trajectory realizes only one continuation at each decision, limiting supervision coverage under a finite sampling budget. We propose **BranchOPD (Branching On-Policy Distillation)**, which selects informative decision points using student uncertainty and teacher–student disagreement, resamples student-policy continuations from shared prefixes, and applies token-level teacher supervision to both main trajectories and branches. BranchingOPD operates at token-level decisions in mathematics and within-turn decisions in agent tasks. Shared-prefix reuse reduces student-generated token costs by approximately 19% on ALFWorld and 31% on WebShop relative to seven independent trajectories. We also introduce protocol-boundary repair (PBR), which uses targeted supervision under protocol constraints to mitigate repetition and wasted generation. BranchOPD outperforms Vanilla OPD and Multi-Rollout OPD ($n = 7$) across all evaluated benchmarks, with gains over Vanilla OPD of 18.55 percentage points in mean mathematics accuracy over 16 sampled answers (avg@16) and 13.8 points in mean ALFWorld ID/OOD success. Mechanistic analyses and selection ablations show that branching at uncertain decisions expands subsequent behavior coverage and outperforms random selection and turn-initial branching.

1 INTRODUCTION

Distillation is a standard way to transfer the reasoning and interaction abilities of a large language model to a smaller one. Training on fixed teacher outputs (Hinton et al., 2015; Kim & Rush, 2016) exposes the student only to prefixes the teacher would write, whereas at inference the student must continue from its own. On-policy distillation (OPD) closes this gap: the student generates trajectories and the teacher scores every generated token (Agarwal et al., 2023; Ross et al., 2010; Bengio et al., 2015; Gu et al., 2023; Ko et al., 2024).

Each trajectory, however, follows only one of the continuations available at every decision. Even with the teacher's full next-token distribution at a prefix, the student receives no supervision on the multi-step histories that the unchosen tokens would have led to. This matters in mathematical reasoning, where an early step constrains the rest of the derivation, and in agent tasks, where an action changes all later observations (Yao et al., 2022b; Schick et al., 2023; Wang et al., 2022b; Wei et al., 2022; Qin et al., 2023). Sampling more independent trajectories widens coverage, but most of the extra tokens regenerate identical or near-identical prefixes. As Figure 1 shows, resampling from a single prefix can reach quite different paths, sometimes even when the first resampled token matches the original.

Recent OPD methods decide where to spend supervision on trajectories that have already been sampled: they reweight tokens, reasoning steps, or turns (Xu et al., 2026; Zhong et al., 2026; Tan et al., 2026; Sheng et al., 2026), order trajectories by difficulty (Wang et al., 2026b), or reuse teacher-scored samples (Zheng et al., 2026). We ask instead which *additional* trajectories to sample. Tree search and branching reinforcement learning already share prefixes to explore locally (Li et al., 2025; Dong et al., 2025; Zong et al., 2026; Ji et al., 2025; Yao et al., 2023; Zhou et al., 2023a; Hao et al., 2023; Khomich et al., 2026; Yu et al., 2026c; He et al., 2026); we bring this idea to distillation, where every branch token receives dense teacher supervision and no task reward is needed.

We propose BranchOPD. For each input, the student first generates a main trajectory. We then pick a few fork points using student uncertainty and teacher–student disagreement, and sample new

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.

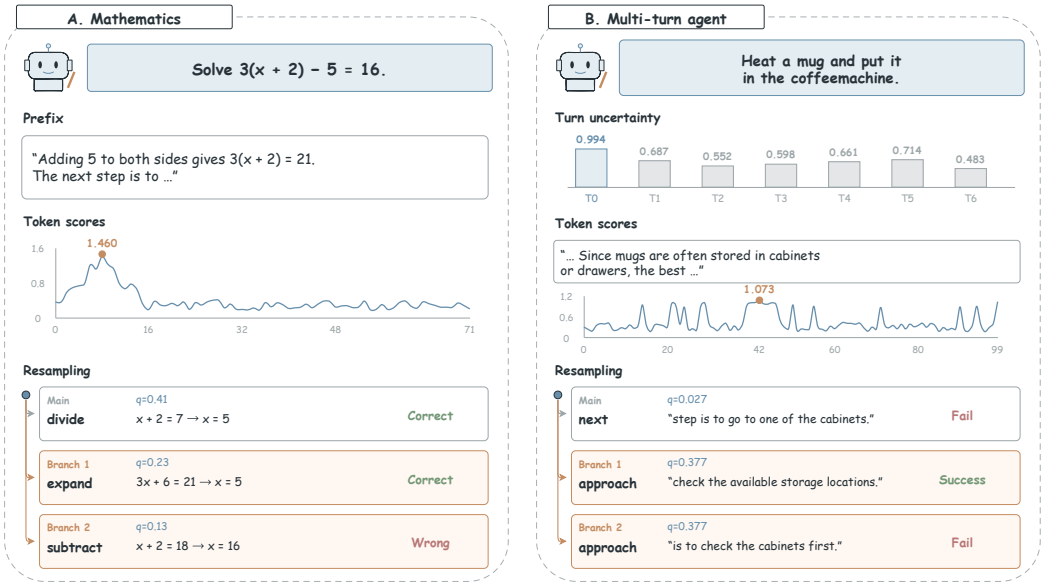


Figure 1: Branching in mathematics (illustrative) and multi-turn interaction (a logged ALFWorld example). Branches that start with the same token can still diverge later. Position scores illustrate the selection step; they are neither probabilities nor the final rank-normalized scores.

continuations from the frozen student policy at each fork. In mathematics tasks, we select token positions directly; in agent tasks we first select turns and then positions within them. The teacher scores all generated tokens on the main trajectory and the branches, and a shared prefix enters the loss only once. Because branches come from the student rather than from the teacher or a refined trajectory (Jiang et al., 2026; Zhou et al., 2026), they stay on the student's own distribution.

A commonly encountered failure mode in OPD is improper termination: the student may continue generating after producing the final answer, often emitting redundant content or stray formatting tokens (Yang et al., 2026b; Luo et al., 2026). We handle this problem by *protocol-boundary repair* (PBR), an auxiliary loss on detected protocol events, and apply PBR to the baselines so that comparisons isolate the effect of branching.

Our contributions are:

- **Branching on-policy distillation.** A shared-prefix sampling scheme that selects informative decisions and resamples student continuations from them for token-level teacher supervision, in both mathematical reasoning and multi-turn agent tasks.
- **Protocol-boundary repair.** An auxiliary loss that reinforces beneficial and suppresses harmful protocol events in student outputs, reducing repetition and wasted generation.
- **Experiments and analysis.** BranchOPD outperforms Vanilla OPD and Multi-Rollout OPD on all evaluated benchmarks, including when every method uses PBR, while generating about 19% and 31% fewer student tokens than seven independent trajectories on ALFWorld and WebShop. Ablations and diagnostics show that the choice of fork points matters.

2 Related Work

On-policy distillation. GKD, MiniLLM, and DistiLLM train students on their own samples and study which divergence to minimize (Agarwal et al., 2023; Gu et al., 2023; Ko et al., 2024). Reasoning supervision can also come from solution selection, process feedback, or explanations (Lightman et al., 2023; Zelikman et al., 2022; Hsieh et al., 2023). More recent OPD methods reweight supervision across tokens (Xu et al., 2026; Sheng et al., 2026), reasoning steps (Zhong et al., 2026), or interaction turns (Tan et al., 2026), or schedule trajectories by difficulty (Wang et al., 2026b).

EOPD adds a forward-KL term at tokens where the teacher has high entropy (Jin et al., 2026), and other work reformulates the target or constrains the update (Jang et al., 2026; Ko et al., 2026). Closest to ours, SPOT forces teacher-proposed tokens at selected positions, scores the resulting student continuations with an outcome verifier, and distills a calibrated local target (Qu et al., 2026). In BranchOPD, branches are sampled from the student itself and supervised token by token, without task rewards.

Quality of distillation supervision. Probability mismatch and degenerate agreement motivate studies of what makes OPD supervision useful (Yu et al., 2026d; Xin et al., 2026; Armandpour et al., 2026). Trajectory refinement and expert corrections improve the examples shown to the student (Jiang et al., 2026; Lauffer et al., 2025), and teacher continuations from student anchors extend supervision beyond sampled rollouts (Zhou et al., 2026); our continuations come from the student. Our objective is token-local and complements work that separates token-level updates from sequence-level credit assignment (Liu et al., 2026). PBR follows unlikelihood training (Welleck et al., 2019) in supervising specific output events.

Tree search and reinforcement learning. Local and tree-structured exploration reuses prefixes to sample alternative decisions (Li et al., 2025; Dong et al., 2025; Zong et al., 2026; Ji et al., 2025). Recent methods branch at high-entropy tokens with sibling-return baselines (He et al., 2026), adapt rollout budgets (Khomich et al., 2026; Yu et al., 2026c), or roll back the environment and reflect (Yu et al., 2026b). These methods compare branches through task rewards; BranchOPD uses shared-prefix exploration to collect more tokens for teacher supervision. Appendix E.6 covers further learning and systems work.

3 Preliminaries

3.1 Student trajectories

Let $x \sim \mathcal{D}$ be an input, $\mathcal{V}$ the vocabulary, π_θ the student policy, and π_T a frozen teacher. We write $a_t \in \mathcal{V}$ for a generated token and s_t for its prefix. In mathematical reasoning, a trajectory is a single response $\tau = (x, a_1, \ldots, a_{|\tau|})$ that ends at a stop token or the length limit, with $s_t = (x, a_{<t})$ and $\pi_\theta(\tau \mid x) = \prod_t \pi_\theta(a_t \mid s_t)$. In multi-turn agent tasks, following Tan et al. (2026), the student interacts with an environment $\mathcal{E}$ for up to M turns:

$$\tau = (x, y_1, o_1, \ldots, y_M, o_M),$$
$$\pi_\theta(\tau \mid x) = \prod_{t \in \mathcal{T}(\tau)} \pi_\theta(a_t \mid s_t) \prod_{m=1}^{M} P_{\mathcal{E}}(o_m \mid x, y_{\le m}, o_{<m}). \tag{1}$$

Each response y_m is a token sequence that may contain reasoning and an environment action, and o_m is the resulting observation. $\mathcal{T}(\tau)$ indexes the student-generated tokens, whose prefixes s_t include earlier responses and observations; observations are never supervised. Mathematics is the special case $M = 1$ without observations.

For $t \in \mathcal{T}(\tau)$, let $\tau_{\ge t}$ denote the continuation from s_t, including later environment transitions, and $\pi_\theta(\tau_{\ge t} \mid s_t)$ its distribution. To resume from s_t in an agent task, the environment must be restored to its state at s_t, by snapshot or action replay, so that the resumed continuation has the same distribution as an uninterrupted rollout.

3.2 On-policy distillation

OPD trains the student on its own trajectories under teacher supervision (Agarwal et al., 2023; Gu et al., 2023; Ko et al., 2024). Trajectories are sampled from $\pi_{\theta_{\text{old}}}$, a frozen copy of the student; we sample at temperature one without truncation, so $\pi_{\theta_{\text{old}}}$ is exactly the sampling distribution. OPD minimizes the reverse KL divergence on student-visited prefixes:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, \tau \sim \pi_{\theta_{\text{old}}}(\cdot|x)} \Big[\frac{1}{|\mathcal{T}(\tau)|} \sum_{t \in \mathcal{T}(\tau)} \mathrm{KL}\left(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_T(\cdot \mid s_t)\right) \Big]. \tag{2}$$

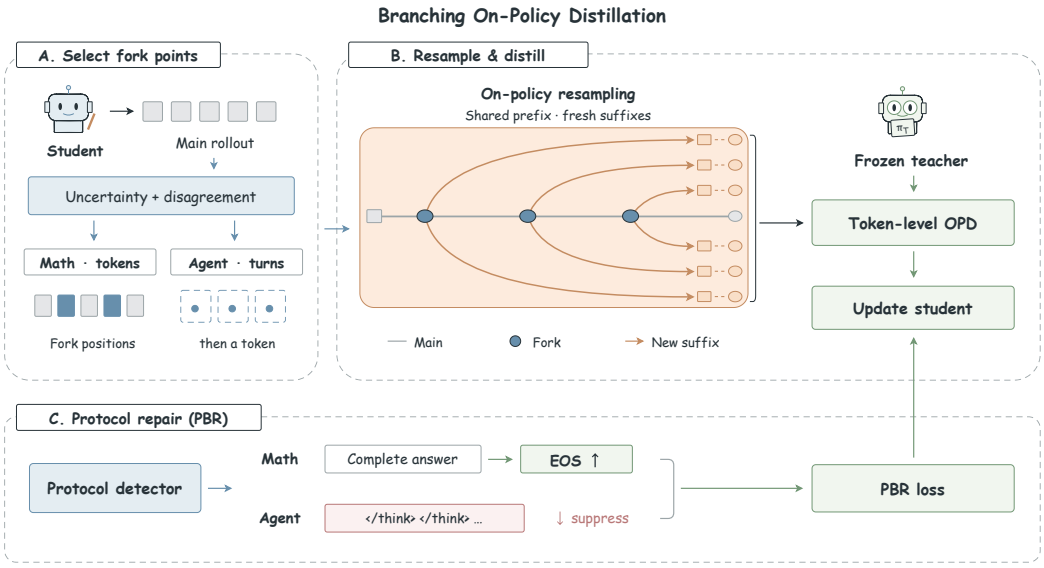


Figure 2: Overview of BranchOPD. Fork points are selected on the main trajectory, suffixes are resampled from the student, and the teacher scores every generated token; shared prefixes are counted once. The bottom examples show concrete PBR constraints used in Section 5.5.

In practice, the teacher scores only the sampled tokens, which gives the token-level distillation signal

$$\Delta \log p_t = \log \pi_T(a_t \mid s_t) - \log \pi_\theta(a_t \mid s_t). \tag{3}$$

If the teacher's support covers the student's, $\mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)}[-\Delta \log p_t]$ equals the reverse KL at s_t. $\Delta \log p_t$ is positive when the teacher assigns the sampled token higher probability than the student does. With the importance ratio $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$ and $\pi_\theta \ll \pi_{\theta_{\text{old}}}$, the gradient at each prefix is

$$\nabla_\theta \, \mathrm{KL}\left(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_T(\cdot \mid s_t)\right) = -\mathbb{E}_{a_t \sim \pi_{\theta_{\text{old}}}(\cdot|s_t)}\left[r_t(\theta)\, \Delta \log p_t \, \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]. \tag{4}$$

OPD is thus a policy-gradient update with the detached token advantage $A_t^{\text{OPD}} = \mathrm{sg}[\Delta \log p_t]$, usually optimized with the PPO clipped surrogate (Schulman et al., 2017; Tan et al., 2026). With n trajectories $\{\tau_i\}_{i=1}^n$ per input and $N = \sum_i |\mathcal{T}(\tau_i)|$,

$$\mathcal{L}_{\text{clip}}(\theta) = -\frac{1}{N} \sum_{i=1}^{n} \sum_{t \in \mathcal{T}(\tau_i)} \min\left\{r_{i,t}(\theta) A_{i,t}^{\text{OPD}},\ \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon) A_{i,t}^{\text{OPD}}\right\}. \tag{5}$$

Vanilla OPD uses $n = 1$; Multi-Rollout OPD uses $n > 1$ independent trajectories. As in the local distillation update of Liu et al. (2026), gradients do not flow through the sampled prefixes (derivations in Appendix A).

4 Method

As Figure 2 shows, BranchOPD selects fork points on a student trajectory, resamples continuations from them with the student policy, and trains on all generated tokens with the OPD loss and PBR.

4.1 Selecting fork points

For each input x, we sample a main trajectory $\tau_0 \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ and select $B_{\text{eff}} \le B$ fork positions $t_1, \ldots, t_{B_{\text{eff}}} \in \mathcal{T}(\tau_0)$ with prefixes $s_b = s_{t_b}$. From each prefix we independently sample k new suffixes from $\pi_{\theta_{\text{old}}}(\cdot \mid s_b)$, so each input yields at most $1 + Bk$ trajectories. In agent tasks the environment is first restored to its state at s_b; in mathematics, generation simply continues from s_b.

We keep every sampled suffix, including those whose first token repeats the main trajectory's, since they can still diverge later.

Candidate positions are scored with two signals. The uncertainty score U_t is the student's entropy, computed from its top-20 token probabilities. The disagreement score D_t sums $-\Delta \log p_t$ over a local window of 128 tokens, so it is large where the student prefers tokens that the teacher does not. Both scores are rank-normalized within the trajectory (denoted R) and mixed:

$$v_t = \alpha R(U_t) + (1-\alpha)R(D_t), \qquad v_{\text{turn}}(m) = \text{mean}\left(\text{TopFrac}_\eta\{v_t : t \in y_m\}\right), \tag{6}$$

where TopFrac_η keeps the top η fraction of scores within turn m. In mathematics, we select up to B forks among the highest-scoring eligible positions. In agent tasks, we select up to B turns with the highest v_{turn}, subject to a minimum gap between selected turns, and place each fork at a position drawn uniformly from the three highest-scoring eligible positions of its turn. By default, $\alpha = 0.5$, $\eta = 0.2$, $B = 3$, and $k = 2$; the scores are not used in the loss. Appendix D lists the eligibility rules and selection settings.

4.2 Conditional resampling and coverage

Let $\mathcal{F}_0$ collect the main trajectory and the selected forks, and let $\tau_{b,j}$ be the j-th suffix sampled from s_b. We fix $\pi_{\theta_{\text{old}}}$ and θ before sampling the branches. For an integrable suffix statistic f_θ, define

$$J_\theta(s_b) = \mathbb{E}_{\tau_{\geq t_b} \sim \pi_{\theta_{\text{old}}}(\cdot|s_b)} f_\theta(\tau_{\geq t_b}), \qquad \widehat{J}_{\text{RS}}(s_b) = \frac{1}{k}\sum_{j=1}^{k} f_\theta(\tau_{b,j}). \tag{7}$$

Proposition 1 (Conditional unbiasedness)**.** *If restored contexts have the correct continuation distribution and each branch satisfies $\tau_{b,j} \mid \mathcal{F}_0 \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_b)$, then $\mathbb{E}[\widehat{J}_{\text{RS}}(s_b) \mid \mathcal{F}_0] = J_\theta(s_b)$.*

The result follows from linearity of conditional expectation (Appendix A). If the branches are also conditionally independent with finite variance, $\text{Var}(\widehat{J}_{\text{RS}} \mid \mathcal{F}_0) = \text{Var}(f_\theta \mid \mathcal{F}_0)/k$. Fork selection may depend on the main trajectory, but it is contained in $\mathcal{F}_0$ and therefore does not bias this estimate. Branching also widens coverage at the fork. Let a_0 be the main-trajectory token at s_b and $a_{b,j}$ the first token of $\tau_{b,j}$. Then

$$\begin{aligned} \Pr(a \text{ visited} \mid \mathcal{F}_0) &= 1 - [1 - \pi_{\theta_{\text{old}}}(a \mid s_b)]^k, \\ \Pr(\exists j : a_{b,j} \neq a_0 \mid \mathcal{F}_0) &= 1 - \pi_{\theta_{\text{old}}}(a_0 \mid s_b)^k. \end{aligned} \tag{8}$$

Both probabilities grow with k but with diminishing increments, which motivates spreading the budget over several forks.

4.3 Training objective

The training batch contains the main trajectory and the $B_{\text{eff}}k$ branches, indexed by i. The mask $M_{i,t}$ keeps all student-generated tokens of the main trajectory and, on each branch, only the tokens from the fork onward; observations and padding are masked out. We compute $\Delta \log p_{i,t}$ (Equation 3) with the current parameters, apply a log-probability floor, and detach the result to form the advantage. With signal threshold δ and PPO clipping ϵ (Schulman et al., 2017), Equation 5 becomes

$$A_{i,t} = \text{clip}(\text{sg}[\Delta \log p_{i,t}], -\delta, \delta), \quad r_{i,t}(\theta) = \frac{\pi_\theta(a_{i,t} \mid s_{i,t})}{\pi_{\theta_{\text{old}}}(a_{i,t} \mid s_{i,t})}, \tag{9}$$

$$g_{i,t} = -\min\{r_{i,t}A_{i,t}, \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon)A_{i,t}\}, \tag{10}$$

$$L_{\text{tree}} = \frac{\sum_{i,t} M_{i,t} g_{i,t}}{\sum_{i,t} M_{i,t}}. \tag{11}$$

Because the loss averages over valid tokens, longer suffixes contribute more terms, and each shared prefix is counted once.

4.4 PROTOCOL-BOUNDARY REPAIR

Some errors are defined by the task protocol rather than by the teacher: a mathematical response should end after its final answer, and an agent response should not begin with a stray closing-thought tag. OPD does not reliably remove them, because a teacher conditioned on a repetitive student prefix may continue the repetition itself (Appendix B.5). PBR supervises such events directly. Fixed detection rules scan each generated trajectory and emit constraints (s, C, b): at prefix s, the next token should ($b = 1$) or should not ($b = 0$) lie in a nonempty token set $C \subsetneq \mathcal{V}$; detection is not differentiated. We define

$$p_\theta(C \mid s) = \sum_{v \in C} \pi_\theta(v \mid s),$$

$$\ell_{\text{proto}}(s, C, b) = -b \log p_\theta(C \mid s) - (1 - b) \log[1 - p_\theta(C \mid s)], \tag{12}$$

$$L_{\text{train}} = L_{\text{tree}} + \Psi\left(\lambda \, \text{mean}_{(s,C,b) \in \mathcal{C}_{\text{batch}}} \ell_{\text{proto}}\right), \tag{13}$$

where $\mathcal{C}_{\text{batch}}$ is the set of constraints detected in the batch and λ is the PBR weight; the auxiliary term is zero when $\mathcal{C}_{\text{batch}}$ is empty. Ψ is the identity, or $\Psi(u) = \min(u, c_{\text{PBR}})$ when a loss cap is used. Positive constraints raise the probability of beneficial events and negative constraints lower that of harmful ones, as in unlikelihood training (Welleck et al., 2019). PBR applies equally to main and branch trajectories. Appendix F gives pseudocode.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate on mathematical reasoning and three agent benchmarks. For mathematics, we distill Qwen3-8B into Qwen3-4B-Base on DAPO-Math-17K (Yang et al., 2025; Yu et al., 2025). For the agent tasks, the student is Qwen3-4B; the teacher is Qwen3-30B-A3B on ALFWorld and a GRPO-trained checkpoint of the same model on WebShop and Search-QA. To test other student sizes and model families, we also distill Qwen3-4B into Qwen3-1.7B, and Qwen2.5-7B-RL into Qwen2.5-3B on ALFWorld. Benchmarks, prompts, and hyperparameters are described in Appendices E, H, and D.

Evaluation. For mathematics, we report avg@16, the accuracy averaged over 16 sampled answers per problem (not pass@16), on MATH-500, AIME 2024, and AIME 2025; Avg is the unweighted mean of the three unrounded scores. On ALFWorld (Shridhar et al., 2020), we report success rate (SR) on 140 seen (ID) and 134 unseen (OOD) tasks and their mean. On WebShop (Yao et al., 2022a), we report SR on 128 held-out tasks, and on Search-QA, exact-match SR on the same mixed validation set for all models. Agents are evaluated with one trajectory per task; no branching is used at evaluation time.

Baselines. Our main baselines are Vanilla OPD, Multi-Rollout OPD with seven independent trajectories per input (matching the $1 + Bk = 7$ trajectories of BranchOPD), and Entropy-Weighted OPD, which reweights the token losses of a single trajectory. Table 1 also includes offline KD, GRPO, EOPD (Jin et al., 2026), and SPOT (Qu et al., 2026) (Appendix D.2); Table 2 includes KD, GRPO, and the curriculum method TCOD (Wang et al., 2026b). Baselines trained with PBR are marked (+PBR).

5.2 MAIN RESULTS

BranchOPD gives the best trained-student results in both tables, improving over Vanilla OPD by 18.55 points in mathematics Avg and 13.83 points in mean ALFWorld success (Tables 1 and 2). The gain does not come from PBR alone: when every method uses PBR, BranchOPD reaches 44.37 mathematics Avg, 5.43 and 5.28 points above Vanilla OPD and Multi-Rollout OPD ($n = 7$), and also above Entropy-Weighted OPD.

Mathematical reasoning. BranchOPD outperforms KD, GRPO, EOPD, and SPOT by 11.27, 8.81, 6.40, and 5.94 Avg points (Table 1). Gains are larger on AIME than on MATH-500: over

Table 1: Mathematics avg@16 (%). Avg is the macro-average across the three evaluations. Baseline variants using protocol-boundary repair are marked (+PBR).

Method	MATH-500	AIME24	AIME25	Avg
Qwen3-8B teacher	89.13	58.54	46.25	64.64
Qwen3-4B-Base student	48.91	8.75	7.50	21.72
KD	74.73	12.50	12.08	33.10
GRPO	79.18	14.17	13.33	35.56
Vanilla OPD	64.33	7.08	6.04	25.82
Multi-Rollout OPD ($n = 7$)	66.64	10.21	6.67	27.84
Entropy-Weighted OPD	67.94	12.29	8.96	29.73
EOPD	79.73	17.50	16.67	37.97
SPOT	79.88	17.50	17.92	38.43
Vanilla OPD (+PBR)	79.13	19.58	18.13	38.94
Multi-Rollout OPD ($n = 7$, +PBR)	78.94	19.79	18.54	39.09
Entropy-Weighted OPD (+PBR)	80.24	21.88	20.83	40.98
BranchOPD	**81.24**	**26.04**	**25.83**	**44.37**

Table 2: Performance on ALFWorld, Search-QA, and WebShop. We report success rate (SR, %); ALFWorld reports ID SR and OOD SR. Avg. SR is the mean success rate over the three datasets, averaging ALFWorld ID and OOD first. †: the WebShop and Search-QA teacher checkpoints are GRPO-trained.

Method	ALFWorld		WebShop	Search-QA	Avg. SR
	ID SR	OOD SR	SR	SR	
Qwen3-30B-A3B	53.57	50.00	31.25†	51.12†	44.72
Qwen3-4B student	35.71	35.82	2.34	28.71	22.27
KD	38.57	38.06	18.75	43.50	33.52
GRPO	51.43	50.75	31.25	49.00	43.78
Vanilla OPD	42.14	43.28	25.78	48.24	38.91
Multi-Rollout OPD ($n = 7$)	43.57	42.54	26.56	48.26	39.29
Entropy-Weighted OPD	46.43	46.27	27.34	48.56	40.75
TCOD	52.14	50.75	31.25	48.70	43.80
Vanilla OPD (+PBR)	49.29	49.25	30.47	48.58	42.77
Multi-Rollout OPD ($n = 7$, +PBR)	50.00	47.76	31.25	48.60	42.91
Entropy-Weighted OPD (+PBR)	52.86	51.49	32.03	48.90	44.37
BranchOPD	**57.86**	**55.22**	**33.59**	**49.05**	**46.39**

SPOT, BranchOPD improves by 1.36 points on MATH-500 but by 8.54 and 7.91 points on AIME24 and AIME25. GRPO and SPOT use answer verification, whereas BranchOPD uses only teacher supervision.

Agent tasks. Averaged over the three benchmarks, BranchOPD reaches 46.39 SR, outperforming KD, GRPO, and TCOD by 12.87, 2.62, and 2.60 points, respectively (Table 2). On ALFWorld, it raises mean success from 49.27% for Vanilla OPD (+PBR) to 56.54%, ahead of the other PBR baselines and 4.75 points above its teacher (51.79%), without any task reward. On WebShop, it solves 43 of 128 tasks instead of 39 and also exceeds the teacher. On Search-QA, the gain over Vanilla OPD (+PBR) is 0.47 points. Tables 4 and 5 report results for a smaller student and another model family.

5.3 Ablations: Where and How Much to Branch

Where to branch. The caption of Table 3 defines each selection rule. Mixing entropy and disagreement works best. Compared with random forks, it improves ALFWorld ID/OOD SR by 7.14/8.96 points and raises mathematics Avg from 40.11 to 44.37. Either signal alone recovers most,

Table 3: Ablations of fork selection and branch budget, all with PBR (ALFWorld SR and mathematics Avg, %). *Fork selection* rows share the eligible positions, budget, and resampling, and differ only in how forks are chosen. *Random*: uniform over eligible positions, without scoring (agent tasks keep the minimum turn gap). *Entropy only* / *Disagreement only*: v_t in Equation 6 with $\alpha = 1$ / $\alpha = 0$. *Turn-initial* (agent tasks only): turns chosen by the same v_{turn}, but each fork is placed at the turn's first generated token. *Branch budget* rows vary the number of forks B and continuations per fork k ($1 + Bk$ trajectories per input).

Setting	ALFWorld		Math Avg
	ID SR	OOD SR	
Fork selection ($B = 3, k = 2$)			
Random	50.71	46.27	40.11
Entropy only ($\alpha = 1$)	55.71	52.99	43.24
Disagreement only ($\alpha = 0$)	55.00	52.24	42.97
Turn-initial	55.71	53.73	–
Entropy + disagreement ($\alpha = 0.5$)	**57.86**	**55.22**	**44.37**
Branch budget (entropy + disagreement)			
$B = 3, k = 1$	54.29	50.75	42.18
$B = 3, k = 3$	57.14	54.48	43.91
$B = 2, k = 2$	57.14	54.48	43.94
$B = 3, k = 2$ **(default)**	**57.86**	**55.22**	**44.37**

but not all, of this gain. Turn-initial branching trails the default by 2.14/1.49 points on ALFWorld, so where the fork falls within a turn also matters. Table 6 reports further selection and filtering controls.

How much to branch. With $B = 3$, increasing the continuations per fork from $k = 1$ to $k = 2$ improves all three metrics, while $k = 3$ brings no further gain. With $k = 2$, going from two to three forks adds 0.43 mathematics points and 0.73 ALFWorld points, so it pays to cover several decisions with a few continuations each (Table 7).

5.4 Behavior Coverage and Sampling Efficiency

Behavior coverage is the range of alternative behaviors the student explores from the same decision state. We measure both the probability mass left to alternatives at a fork, $1-\pi_{\theta_{\text{old}}}(a_0 \mid s)$ with a_0 the main-trajectory token, and the behavioral differences that branches actually produce. At the selected forks (step 100), this mass is about 0.63 on ALFWorld and 0.69 on WebShop, and branches realize much of it: 74.2% and 47.7% take a different action sequence from the main trajectory, and 55.5% and 37.5% contain an action absent from it (Table 14). Branching thus turns mass that a single rollout leaves unrealized into additional histories with teacher feedback. Branches can also recover from failures: when the main trajectory fails, at least one branch succeeds in 5.0% (WebShop) to 42.7% (mathematics) of cases (Table 15). Appendix C tracks these statistics over training.

While achieving broader behavior coverage, BranchOPD requires fewer student-generated tokens by reusing the prefix shared with the main trajectory. On ALFWorld, one trajectory requires 2,813 student tokens on average; seven independent trajectories would therefore require 19,691 tokens, whereas our seven-trajectory branching setting uses only 15,924 tokens, saving 19.1%. On WebShop, the corresponding cost is 4,824 tokens versus 7,014 for seven independent trajectories, saving 31.2% (Figure 3; Table 11). These counts exclude teacher scoring, backpropagation, and environment replay.

5.5 Protocol-Boundary Repair

PBR explicitly controls protocol events that should either be encouraged or suppressed. For example, it encourages termination after a complete mathematical answer by setting (C=EOS) and (b=1), while suppressing extra leading closing-thought tags in agent responses with (b=0). On mathematics, PBR alone lifts Vanilla OPD from 25.82 to 38.94 Avg (Table 1) and leaves more room below the

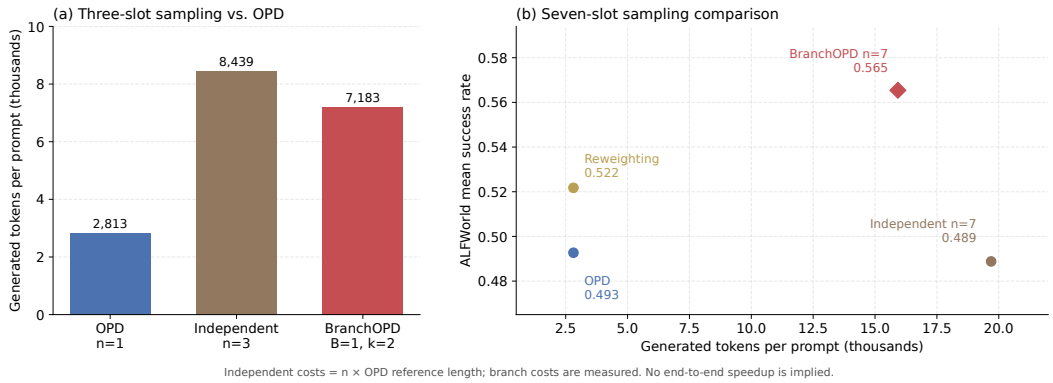


Figure 3: ALFWorld generation cost and success with PBR, for three (left) and seven (right) trajectory slots. OPD, Independent, and Reweighting denote Vanilla OPD, Multi-Rollout OPD, and Entropy-Weighted OPD. Multi-Rollout costs are constructed from Vanilla OPD runs; branch costs are measured.

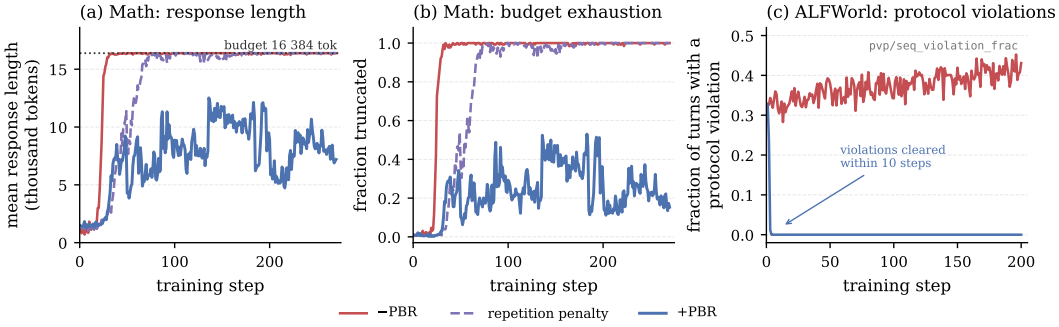


Figure 4: Training dynamics with PBR: mathematics response length and truncation rate (fraction of responses stopped at the length limit), and ALFWorld turn-level protocol violation (fraction of turns with a violation). Repetition penalties delay budget exhaustion but leave generation near the cap; PBR reduces both truncation and violations.

length limit (Figure 4). On ALFWorld, it improves ID/OOD SR by 7.14/5.97 points and removes the extra leading tags entirely. A teacher that does not repeat in its own rollouts can still reinforce repetition when conditioned on a repetitive student prefix, so the repair must act on student prefixes (Appendix B.5).

Why targeted repair? We also compare PBR with generic remedies that broadly discourage repetition or truncate problematic samples, which are consistently less effective than PBR (Table 13). Repetition penalties achieve only 28.70–30.40 Avg, filtering overlong samples reaches 33.20, and early stopping reaches 22.05, compared with 38.94 for PBR. Moreover, repetition penalties mainly slow the increase in response length rather than preventing it from eventually approaching the generation limit (Figure 4).

6 Conclusion

We presented BranchOPD, which resamples student continuations at selected decision points to broaden on-policy distillation, together with protocol-boundary repair. It outperforms Vanilla OPD and Multi-Rollout OPD on every evaluated benchmark while generating fewer student tokens than independent rollouts. What a student learns from OPD depends not only on the teacher signal but also on which of its own alternative histories it gets to see.

AI USE STATEMENT

We used large language models to assist with literature retrieval and synthesis, paper drafting and translation, refinement of theoretical formulations and proofs, and interpretation of results. The authors are responsible for the accuracy, originality, and final content of the submission, including AI-assisted text, claims, and artifacts.

ETHICS STATEMENT

Our research uses public benchmarks and controlled task environments, and does not involve human-subject experiments or collection of sensitive personal data.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2023. URL https://arxiv.org/abs/2306.13649.

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

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks, 2015. URL https://arxiv.org/abs/1506.03099.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying

Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization, 2025. URL https://arxiv.org/abs/2507.19849.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel,

Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: On-policy distillation of large language models, 2023. URL https://arxiv.org/abs/2306.08543.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model, 2023. URL https://arxiv.org/abs/2305.14992.

Bowei He, Yankai Chen, Xiaokun Zhang, and Xue Liu. Branching policy optimization: Sandbox-native language agent reinforcement learning, 2026. URL https://arxiv.org/abs/2607.14171.

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, 2021. URL https://arxiv.org/abs/2103.03874.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, pp. 663–685, 1952. doi: 10.1080/01621459.1952.10483446. URL https://doi.org/10.1080/01621459.1952.10483446.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023. URL https://arxiv.org/abs/2305.02301.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation, 2026. URL https://arxiv.org/abs/2601.07155.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning, 2025. URL https://arxiv.org/abs/2509.21240.

Li Jiang, Haoran Xu, Yichuan Ding, and Amy Zhang. Trajectory-refined distillation, 2026. URL https://arxiv.org/abs/2606.08432.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

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, 2026. URL https://arxiv.org/abs/2603.07079.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL https://arxiv.org/abs/1705.03551.

Nikita Khomich, Leopold Hermansson, and Ido Hakimi. Epig-tree: Compute-optimal branching for gradient-efficient reinforcement learning, 2026. URL https://arxiv.org/abs/2609.20004.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation, 2016. URL https://arxiv.org/abs/1606.07947.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models, 2024. URL https://arxiv.org/abs/2402.03898.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation, 2026. URL https://arxiv.org/abs/2603.11137.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, pp. 452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Niklas Lauffer, Xiang Deng, Srivatsa Kundurthy, Brad Kenstler, and Jeff Da. Imitation learning for multi-turn lm agents via on-policy expert corrections, 2025. URL https://arxiv.org/abs/2512.14895.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

Qingbin Li, Rongkun Xue, Jie Wang, Ming Zhou, Zhi Li, Xiaofeng Ji, Yongqi Wang, Miao Liu, Zheming Yang, Minghui Qiu, and Jing Yang. Cure: Critical-token-guided re-concatenation for entropy-collapse prevention, 2025. URL https://arxiv.org/abs/2508.11016.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Shiqi Liu, Zeyu He, Letian Tao, Guojian Zhan, Jiaxin Gao, Feihong Zhang, Jingliang Duan, Wei Xiong, Kehua Sheng, Bo Zhang, Yang Guan, and Shengbo Eben Li. Beyond token-local imitation: Reward-compatible temporal credit assignment for on-policy distillation, 2026. URL https://arxiv.org/abs/2609.16937.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2605.15155.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. Demystifying opd: Length inflation and stabilization strategies for large language models. *arXiv preprint arXiv:2604.08527*, 2026.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train any ai agents with reinforcement learning, 2025. URL https://arxiv.org/abs/2508.03680.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models, 2026. URL https://arxiv.org/abs/2602.04942.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, Tianjie Zhang, Wei Xu, Jie Tang, and Yuxiao Dong. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning, 2024. URL https://arxiv.org/abs/2411.02337.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023. URL https://arxiv.org/abs/2307.16789.

Zikun Qu, Min Zhang, Mingze Kong, Zhiwei Shang, Zhengyu Chen, Yikun Ban, Shuang Qiu, and Zhongxiang Dai. SPOT: Sparse probing and outcome calibration for on-policy distillation, 2026. URL https://arxiv.org/abs/2608.04419.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2010. URL https://arxiv.org/abs/1011.0686.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

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

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

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, 2024. URL https://arxiv.org/abs/2409.19256.

Huanxin Sheng, Zhiling Ye, Haonan Wang, Jian Wang, Jinjie Gu, and Jian Kang. 1% of tokens can be enough: On gradient estimation in on-policy distillation, 2026. URL https://arxiv.org/abs/2609.24432.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning, 2020. URL https://arxiv.org/abs/2010.03768.

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. URL https://arxiv.org/abs/2606.27814v6.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multi-hop questions via single-hop question composition, 2021. URL https://arxiv.org/abs/2108.00573.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents, 2026a. URL https://arxiv.org/abs/2604.10674.

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. Tcod: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents, 2026b. URL https://arxiv.org/abs/2604.24005v1.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader?, 2022a. URL https://arxiv.org/abs/2203.07540.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2022b. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL https://arxiv.org/abs/2201.11903.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training, 2019. URL https://arxiv.org/abs/1908.04319.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. Agentgym: Evolving large language model-based agents across diverse environments, 2024. URL https://arxiv.org/abs/2406.04151.

Haoran Xin, Anhao Zhao, Ying Sun, Jin Li, Xiaoyu Shen, and Hui Xiong. Escaping the kl agreement trap in on-policy distillation, 2026. URL https://arxiv.org/abs/2606.09471.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation, 2026. URL https://arxiv.org/abs/2604.14084.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation, 2026a. URL https://arxiv.org/abs/2602.12125.

Yuxiao Yang, Tianrun Yu, Shangzhe Li, Kaixiang Zhao, Xuchao Zhang, Chetan Bansal, Huaxiu Yao, Taylor W Killian, and Weitong Zhang. When eos tokens disagree: Understanding length inflation in on-policy distillation. *arXiv preprint arXiv:2609.20511*, 2026b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2022a. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2022b. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL https://arxiv.org/abs/2305.10601.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models, 2026. URL https://arxiv.org/abs/2602.12275.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Yan Yu, Zhengxi Lu, Yizhou Liu, Yichen Pan, Aozhe Wang, Qipeng Chen, Hua Yang, Wenqi Zhang, Weiming Lu, Qianglong Chen, and Yongliang Shen. Retireopd: Self-retiring on-policy distillation for agentic reinforcement learning, 2026a. URL https://arxiv.org/abs/2609.20784.

Yi Yu, Liuyi Yao, Yaliang Li, Enshu Wang, and Libing Wu. Rollback the world, keep the reflection: Rollback-induced reflection for long-horizon llm agents, 2026b. URL https://arxiv.org/abs/2609.18304.

Youngjun Yu, Sanghwan Jang, and Hwanjo Yu. Difficulty-adaptive tree-structured policy optimization for expanding reasoning coverage in rlvr, 2026c. URL https://arxiv.org/abs/2609.08650.

Zichao Yu, Chengzhi Yu, Shengze Xu, Yujin Han, Bingqing Jiang, Xu Wang, and Difan Zou. Mismatch matters: On-policy distillation beyond token agreement, 2026d. URL https://arxiv.org/abs/2608.09836.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

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, 2026. URL https://arxiv.org/abs/2601.18734.

Keye Zheng, Hanyu Li, Zhan Cheng, and Yuan Gao. Cloopd: Closing the learner loop in on-policy distillation, 2026. URL https://arxiv.org/abs/2609.24141.

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. URL https://arxiv.org/abs/2605.07725.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models, 2023a. URL https://arxiv.org/abs/2310.04406.

Jingang Zhou, Yuyi Zhou, Haiyang Guo, Xukai Wang, Shuai Feng, Sirui Gao, Jian Xu, Qingpei Guo, and Xu-Yao Zhang. Teacher should think ahead: Adaptive continuations for reliable on-policy distillation, 2026. URL https://arxiv.org/abs/2609.22254.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2023b. URL https://arxiv.org/abs/2307.13854.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl, 2024. URL https://arxiv.org/abs/2402.19446.

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, 2026. URL https://arxiv.org/abs/2601.04767.

A THEORETICAL DETAILS

Throughout, the rollout policy $\pi_{\theta_{\text{old}}}$ is fixed within each sampling group, s denotes a prefix s_t, and

$$\Delta \log p(s, a) = \log \pi_T(a \mid s) - \log \pi_\theta(a \mid s) \tag{14}$$

is the distillation signal of Equation 3 at token a. We assume $\pi_T(a \mid s) > 0$ whenever $\pi_\theta(a \mid s) > 0$, and $\pi_\theta \ll \pi_{\theta_{\text{old}}}$ at every supervised prefix.

A.1 OPD GRADIENT AND TOKEN SURROGATE

At a fixed prefix s, the reverse KL is $\mathrm{KL}(\pi_\theta(\cdot \mid s) \| \pi_T(\cdot \mid s)) = \sum_a \pi_\theta(a \mid s)[-\Delta \log p(s, a)]$. Using $\nabla_\theta \pi_\theta = \pi_\theta \nabla_\theta \log \pi_\theta$ and $\nabla_\theta[-\Delta \log p(s, a)] = \nabla_\theta \log \pi_\theta(a \mid s)$,

$$\nabla_\theta \,\mathrm{KL}(\pi_\theta(\cdot \mid s) \| \pi_T(\cdot \mid s)) = \sum_a \pi_\theta(a \mid s)\big(1 - \Delta \log p(s, a)\big) \nabla_\theta \log \pi_\theta(a \mid s) \tag{15}$$

$$= -\sum_a \pi_\theta(a \mid s)\, \Delta \log p(s, a)\, \nabla_\theta \log \pi_\theta(a \mid s), \tag{16}$$

since $\sum_a \pi_\theta \nabla_\theta \log \pi_\theta = \nabla_\theta \sum_a \pi_\theta = 0$. Under $\pi_\theta \ll \pi_{\theta_{\text{old}}}$, rewriting the sum as an expectation over $a \sim \pi_{\theta_{\text{old}}}(\cdot \mid s)$ with weight $r(\theta) = \pi_\theta(a \mid s)/\pi_{\theta_{\text{old}}}(a \mid s)$ gives Equation 4. With $A^{\text{OPD}} = \mathrm{sg}[\Delta \log p(s, a)]$, the unclipped surrogate $-r(\theta)A^{\text{OPD}}$ has gradient $-A^{\text{OPD}} r(\theta) \nabla_\theta \log \pi_\theta(a \mid s)$, whose expectation under $\pi_{\theta_{\text{old}}}$ equals the KL gradient when the detached signal is evaluated at the differentiation point. PPO and signal clipping then define the training surrogates in Equations 5 and 11.

Summing over $t \in \mathcal{T}(\tau)$ and averaging over $\tau \sim \pi_{\theta_{\text{old}}}$ yields the gradient of Equation 2 with the prefix distribution held fixed; the update does not differentiate through the rollout distribution (Liu et al., 2026). Equation 2 normalizes within each trajectory, whereas the token-mean surrogate normalizes over the batch and thus weights trajectories by length; the two coincide for a single trajectory.

A.2 CONDITIONAL RESAMPLING

Let $\mathcal{F}_0$ contain the main trajectory and the selected fork prefixes. At a fork prefix $s_b = s_{t_b}$, write a fresh continuation as $\tau_{\geq t_b} = (a, \tau_{>t_b})$, where a is its first token and $\tau_{>t_b}$ includes subsequent tokens and environment transitions. For an integrable statistic f_θ fixed before drawing the continuations,

$$\pi_{\theta_{\text{old}}}(\tau_{\geq t_b} \mid s_b) = \pi_{\theta_{\text{old}}}(a \mid s_b)\, \pi_{\theta_{\text{old}}}(\tau_{>t_b} \mid s_b, a), \tag{17}$$

$$\mu_\theta(s_b, a) = \mathbb{E}_{\tau_{>t_b} \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_b, a)} f_\theta(a, \tau_{>t_b}), \tag{18}$$

$$J_\theta(s_b) = \sum_a \pi_{\theta_{\text{old}}}(a \mid s_b)\, \mu_\theta(s_b, a). \tag{19}$$

Restoring the fork state and drawing each continuation from $\pi_{\theta_{\text{old}}}(\cdot \mid s_b)$ gives

$$\mathbb{E}[\widehat{J}_{\text{RS}}(s_b) \mid \mathcal{F}_0] = \frac{1}{k} \sum_{j=1}^{k} \mathbb{E}[f_\theta(\tau_{b,j}) \mid \mathcal{F}_0] = J_\theta(s_b). \tag{20}$$

This proves Proposition 1 using only the conditional marginal law of each fresh branch. For conditionally independent branches with finite variance,

$$\mathrm{Var}(\widehat{J}_{\text{RS}}(s_b) \mid \mathcal{F}_0) = \frac{\mathrm{Var}(f_\theta(\tau_{\geq t_b}) \mid \mathcal{F}_0)}{k}. \tag{21}$$

Fork selection depends only on the main trajectory and is therefore contained in $\mathcal{F}_0$; the expectation above is taken over the fresh branch draws alone.

A.3 Coverage and diminishing returns

A token a is absent from k independent branch starts at s_b with probability $(1 - \pi_{\theta_{\text{old}}}(a \mid s_b))^k$. Therefore, for the number N_k of distinct first tokens among those branches,

$$\mathbb{E}[N_k \mid \mathcal{F}_0] = \sum_{a \in \mathcal{V}} \left[1 - (1 - \pi_{\theta_{\text{old}}}(a \mid s_b))^k\right], \tag{22}$$

$$\mathbb{E}[N_{k+1} - N_k \mid \mathcal{F}_0] = \sum_{a \in \mathcal{V}} \pi_{\theta_{\text{old}}}(a \mid s_b)(1 - \pi_{\theta_{\text{old}}}(a \mid s_b))^k. \tag{23}$$

The marginal increase is nonnegative and decreases with k. For the realized main token a_0, at least one branch starts differently with probability $1 - \pi_{\theta_{\text{old}}}(a_0 \mid s_b)^k$. Branches that share the first token can still diverge later, so we measure both first-token alternatives and later behavioral differences (Appendix C).

A.4 Token aggregation and protocol repair

Proposition 1 concerns the average of a fixed suffix statistic, whereas the training loss averages over valid tokens in the main trajectory and all branches. A suffix with L generated tokens contributes L loss terms, and shared prefix tokens enter once. Fork selection thus determines which prefixes are supervised, continuation length determines how much each contributes, and the mask $M_{i,t}$ sets the normalization.

For PBR, a positive event contributes $-\log p_\theta(C \mid s)$ and a negative event $-\log(1 - p_\theta(C \mid s))$; their gradients increase and decrease the event probability, respectively. Detection is held fixed during differentiation. With the cap $\Psi(u) = \min(u, c_{\text{PBR}})$, the auxiliary gradient is unchanged below the cap and zero above it; a batch without constraints contributes zero.

A.5 Weighted forced-token comparator

The forced-token controls in Table 8 use a natural token plus k alternatives. At a fixed prefix s, write $p_{\text{old}}(a) = \pi_{\theta_{\text{old}}}(a \mid s)$, sample $A_0 \sim p_{\text{old}}$, and let

$$S(A_0) = \{A_0\} \cup \text{TopK}_{a \neq A_0}\, p_{\text{old}}(a). \tag{24}$$

With deterministic tie-breaking and at least $k+1$ tokens in the support, order tokens as $a_1, a_2, \ldots$ by decreasing p_{old} and write $Q_{k+1} = \sum_{i=1}^{k+1} p_{\text{old}}(a_i)$. Their inclusion probabilities are

$$\rho(a_i) = \begin{cases} 1, & i \le k, \\ Q_{k+1}, & i = k+1, \\ p_{\text{old}}(a_i), & i > k+1. \end{cases} \tag{25}$$

The first k tokens are always included; token $k+1$ enters when A_0 is among the first $k+1$ tokens; all others enter through the natural draw. After forcing token a, the continuation $\tau_{>t}^{(a)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid s, a)$ has conditional mean $\mu_\theta(s, a)$. Horvitz–Thompson weights (Horvitz & Thompson, 1952) then give

$$\widehat{J}_{\text{HT}}(s) = \sum_{a \in S(A_0)} \frac{p_{\text{old}}(a)}{\rho(a)} f_\theta(a, \tau_{>t}^{(a)}), \tag{26}$$

$$\mathbb{E}[\widehat{J}_{\text{HT}}(s) \mid s] = \sum_a \frac{p_{\text{old}}(a)}{\rho(a)} \rho(a) \mu_\theta(s, a) = J_\theta(s). \tag{27}$$

Every token with positive probability has positive inclusion probability. The scaled weights $w_{\text{scaled}}(a) = (k+1)\, p_{\text{old}}(a)/\rho(a)$ satisfy $\widehat{J}_{\text{HT}} = (k+1)^{-1} \sum_{a \in S} w_{\text{scaled}}(a) f_\theta(a, \tau_{>t}^{(a)})$ and $\mathbb{E} \sum_{a \in S} w_{\text{scaled}}(a) = k+1$.

Table 8 uses scaled HT weights in a pooled token loss, with the main-slot weight averaged across its forks. RB instead self-normalizes the weights within each realized group; because its denominator is random, it is a different estimator. The HT identity above holds when the natural draw and the continuations follow the fixed-prefix sampling distribution.

B Additional Results and Controlled Ablations

This section reports results for other students, sampling rules, generation budgets, and protocol repair, using the metrics of Section 5.

B.1 Student size and model family

Table 4: Qwen3-4B to Qwen3-1.7B distillation. Teachers are task-specific GRPO checkpoints. Values are task SR or mathematics mean avg@16 (%); (+PBR) identifies repaired baselines.

Method	ALFWorld		WebShop	Search-QA	Math Avg
	ID SR	OOD SR	SR	SR	
Qwen3-4B GRPO teacher	46.43	41.04	27.34	47.20	61.60
Qwen3-1.7B student	7.86	6.72	3.91	26.40	11.08
Vanilla OPD (+PBR)	44.29	42.54	26.56	45.10	30.18
Multi-Rollout OPD ($n = 7$, +PBR)	45.00	40.30	26.56	45.40	30.05
Entropy-Weighted OPD (+PBR)	47.14	45.52	27.34	45.90	31.60
BranchOPD	**51.43**	**48.51**	**29.69**	**46.80**	**34.86**

Table 5: Qwen2.5-7B-RL to Qwen2.5-3B on ALFWorld. SR (%) uses 140 ID and 134 OOD tasks; trained baselines use PBR.

Method	ALFWorld	
	ID SR	OOD SR
Qwen2.5-7B-RL teacher	64.29	55.22
Qwen2.5-3B zero-shot	7.14	2.99
Vanilla OPD (+PBR)	47.14	41.79
Multi-Rollout OPD ($n = 7$, +PBR)	47.86	40.30
Entropy-Weighted OPD (+PBR)	50.71	44.03
BranchOPD	**55.71**	**50.75**

With Qwen3-1.7B as the student, BranchOPD improves over Vanilla OPD (+PBR) by 6.56 points on ALFWorld and 4.68 points on mathematics, and exceeds the teacher on ALFWorld and WebShop without task rewards (Table 4). With Qwen2.5-3B, the ALFWorld gain is 8.76 points (Table 5).

B.2 Selection and branching budgets

Table 6: Selection and eligibility ablations with $B = 3, k = 2$ and PBR. ALFWorld SR and mathematics mean avg@16 are percentages. α weights entropy in the mixed selector; – denotes an unevaluated setting.

Setting	ALFWorld		Math Avg
	ID SR	OOD SR	
Random sites	50.71	46.27	40.11
Student entropy only	55.71	52.99	43.24
Disagreement only	55.00	52.24	42.97
Entropy + disagreement ($\alpha = 0.3$)	57.14	54.48	43.88
Entropy + disagreement ($\alpha = 0.5$)	**57.86**	**55.22**	**44.37**
Entropy + disagreement ($\alpha = 0.7$)	56.43	53.73	43.65
No mathematics-aware eligibility	–	–	42.03
Turn-initial	55.71	53.73	–
Failed mains only	55.00	50.75	36.82

Table 7: Branch budgets with entropy + disagreement selection and PBR. Performance is in percent; slots equal $1 + Bk$. Generation is relative to Vanilla OPD on ALFWorld; † marks interpolated or extrapolated costs.

Setting	ALFWorld		Math Avg	Slots	Generation
	ID SR	OOD SR			
B=3,k=1	54.29	50.75	42.18	4	3.33× †
B=3,k=2	**57.86**	**55.22**	**44.37**	7	**5.66**×
B=3,k=3	57.14	54.48	43.91	10	7.99× †
B=1,k=2	54.29	51.49	42.55	3	2.55×
B=2,k=2	57.14	54.48	43.94	5	4.11× †

Entropy + disagreement is the best of the tested selection rules (Table 6). Disabling the mathematics-aware eligibility filter lowers mathematics Avg to 42.03, and branching only from failed main trajectories lowers it to 36.82, so branches from successful trajectories are useful as well.

The two budgets play different roles (Table 7). Going from $k = 1$ to $k = 2$ helps, and $k = 3$ adds nothing. At $k = 2$, increasing B from one to three raises mathematics Avg from 42.55 to 44.37. The five-trajectory setting ($B = 2, k = 2$) keeps most of the gain at a lower generation cost.

B.3 Sampling and perturbation controls

Table 8: Expansion rules with PBR. Performance is in percent. HT uses scaled inclusion weights; RB normalizes weights within each sampled group (Appendix A).

Expansion	Weights	ALFWorld		Math Avg	WebShop	Search-QA
		ID SR	OOD SR		SR	SR
forced-topk	Uniform	52.86	48.51	40.90	31.25	48.76
forced-topk	RB	57.14	51.49	41.77	31.25	**49.22**
forced-topk	HT (scaled)	55.71	50.75	42.11	32.03	48.96
resample	Uniform	**57.86**	**55.22**	**44.37**	**33.59**	49.05

Table 9: ALFWorld continuation diagnostics. The first five metrics use step 100; low-likelihood rates summarize 64-token windows. Rates are percentages and new actions is a count.

Metric	Forced + HT	Resample
Main-token probability	37.19%	36.67%
Alternative mass	62.81%	63.33%
Action disagreement	75.8%	74.2%
New-action coverage (NSR)	58.6%	55.5%
Mean new actions	2.59	2.44
Low-likelihood suffixes	19.2%	5.8%

Among the expansion rules, resampling gives the highest ALFWorld, mathematics, and WebShop scores, and RB the highest Search-QA SR (Table 8). All rules change actions at similar rates, but resampling produces far fewer low-likelihood suffixes, 5.8% versus 19.2% (Table 9). A suffix counts as low-likelihood when its mean student log probability over 64 tokens falls below the fifth percentile of on-policy windows of the same length. Useful coverage can therefore be obtained from continuations that the student is likely to generate.

Table 10: Noise-token branching on ALFWorld with PBR. ID/OOD SR and their mean are percentages. Each branch starts with the forced token `!`.

Setting	ALFWorld		ALF Avg
	ID SR	OOD SR	
Vanilla OPD (+PBR)	49.29	49.25	49.27
Two-stage + noise token	50.71	50.00	50.36
Turn-initial + noise	50.00	50.00	50.00

Forcing a noise token (`!`) at the start of each branch yields small ALFWorld gains of 1.09 and 0.73 points over Vanilla OPD (+PBR), depending on where the forks are placed (Table 10). Perturbed continuations still add some supervision, but the much larger gains of informative branching show that fork selection matters.

B.4 Generation and environment costs

Table 11: Generation costs and performance with PBR. Scores are mathematics Avg or task SR (%); ALFWorld averages ID/OOD. Costs are relative to Vanilla OPD, and savings use the 7× independent-rollout reference.

Task	Vanilla OPD	Multi-Rollout OPD ($n = 7$)	BranchOPD	Generation	Saving vs. 7×	Cost basis
Mathematics	38.94	39.09	44.37	5.68×	18.9%	Position estimate
ALFWorld	49.27	48.88	56.54	5.66×	19.1%	Measured tokens
WebShop	30.47	31.25	33.59	4.81×	31.2%	Measured tokens
Search-QA	48.58	48.60	49.05	4.24×	39.4%	Position estimate

Table 12: Three-slot ALFWorld sampling with PBR. SR is the ID/OOD mean (%); generation and environment steps are relative to Vanilla OPD.

Setting	Slots	Generation	Env. steps	ALF Avg
Vanilla OPD (+PBR)	1	1.00×	1.00×	49.30
Resample ($B = 1, k = 2$)	3	2.55×	3.02×	52.90
Forced, turn-initial + RB ($B = 1, k = 2$)	3	2.49×	3.07×	54.48

Relative to seven independent rollouts, BranchOPD saves 19.1% and 31.2% of student generation on ALFWorld and WebShop while improving SR; estimates from fork positions give savings of 18.9% on mathematics and 39.4% on Search-QA (Table 11). With two added continuations, ALFWorld uses 2.55× the generation and 3.02× the environment steps of Vanilla OPD, because state replay adds environment cost on top of suffix generation (Table 12).

B.5 PROTOCOL REPAIR AND REPETITION

Table 13: Repetition treatments for Vanilla OPD on mathematics. Values are mean avg@16 (%).

Recipe	Math Avg
No PBR	25.82
Repetition penalty	28.70
n-gram / line penalty in advantage	30.40
Overlong-sample filtering	33.20
Early stopping during training	22.05
Full PBR	**38.94**

Without PBR, mathematics responses reach the length cap around step 30. Repetition penalties delay this but do not prevent persistent budget exhaustion (Figure 4). PBR reaches 38.94 mathematics Avg, compared with 33.20 for filtering and 28.70–30.40 for penalties; after step 50, its median response length is about 8,600 tokens and its median truncation rate is 0.28.

A teacher whose own rollouts are clean can still reinforce repetition when queried on a repetitive student prefix; PBR targets these prefix-conditioned errors. On ALFWorld, teacher responses contain on average 1.64/1.50 excess leading `</think>` markers on ID/OOD. Vanilla OPD raises this to 1.94/1.91, and PBR reduces it to zero. A single response can contain several such markers.

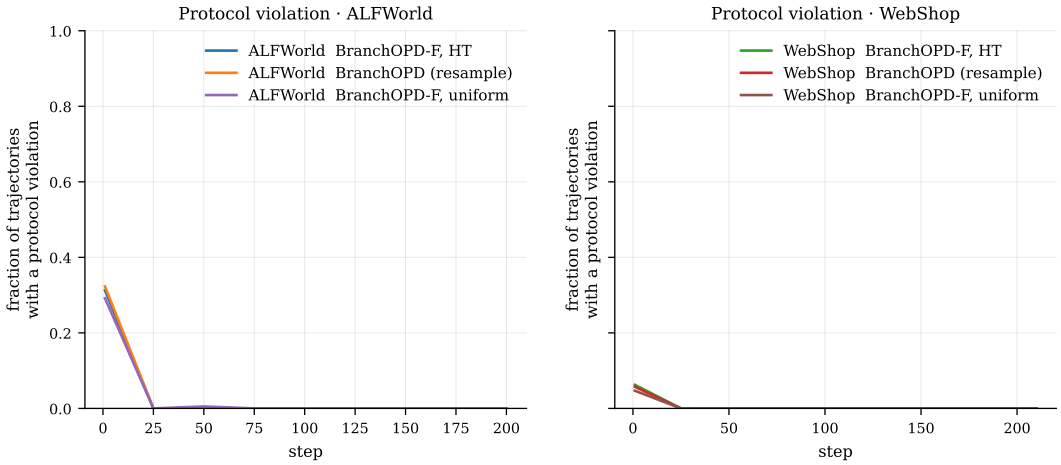


Figure 5: Trajectory-level protocol violation (PVP): the fraction of the 192 main and branch trajectories that contain a violation. All methods shown use PBR.

Trajectory-level protocol violation drops to zero by step 25 and stays near zero (Figure 5). Together with the behavior coverage in Appendix C, this indicates that protocol repair does not suppress exploration.

C BEHAVIOR COVERAGE AND RECOVERY

These diagnostics follow the chain from selecting uncertain decisions, to generating alternative behavior, to recovering successful outcomes.

C.1 PROBABILITY AVAILABLE AT SELECTED DECISIONS

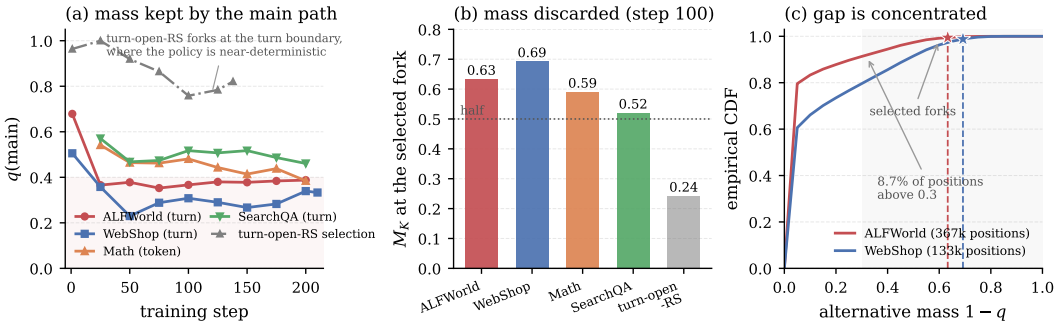


Figure 6: Probability mass available for alternative decisions. (a) "Mass kept by the main path" is the probability of the sampled main token, $\pi_{\theta_{\text{old}}}(a_0 \mid s)$. (b) "Mass discarded" and M_K denote the alternative-token mass $1 - \pi_{\theta_{\text{old}}}(a_0 \mid s)$, i.e., the mass left unrealized by the main draw. (c) All supervised positions compared with selected forks. ALFWorld and WebShop use training dumps; mathematics and Search-QA use reconstructed summaries.

At selected forks, the alternative-token mass $1 - \pi_{\theta_{\text{old}}}(a_0 \mid s)$ is about 0.63 on ALFWorld and 0.69 on WebShop (Figure 6). Ordinary supervised positions are far more concentrated: 80.4% and 59.1% of them have alternative mass below 0.05 (Figure 7). The selected forks are thus decisions with several plausible continuations.

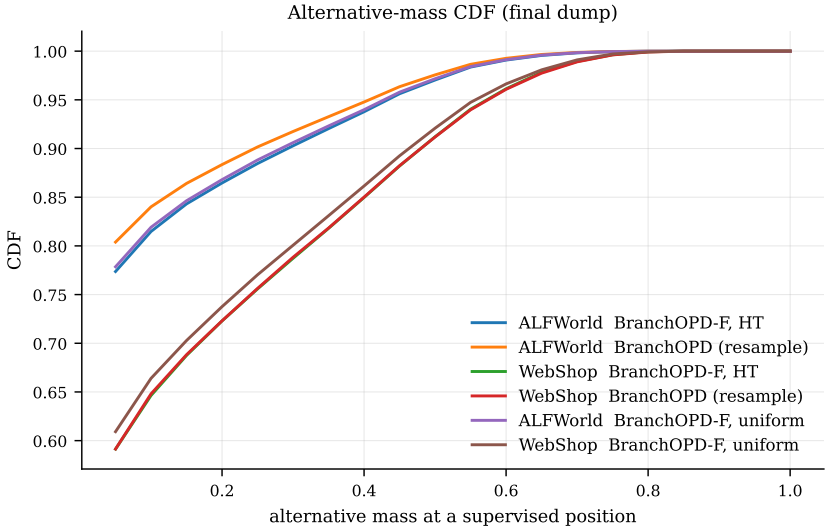


Figure 7: Empirical CDF of alternative mass over supervised positions (ALFWorld step 200, WebShop step 210), accumulated over 20 equal-width bins. "Uniform" denotes uniform weights on forced branches.

C.2 ALTERNATIVE ACTIONS ACROSS TRAINING

Table 14: Behavior coverage from 64 main trajectories and 128 branches per checkpoint. Action and observation disagreement and NSR are percentages; new actions is a count per branch.

Task / step	Action (%)	Observation (%)	NSR (%)	New actions
ALFWorld @100	74.2	74.2	55.5	2.44
ALFWorld @200	74.2	73.4	57.8	3.02
WebShop @100	47.7	38.3	37.5	0.68
WebShop @200	52.3	40.6	32.0	0.60

Action disagreement (EBR) is the fraction of branches whose action sequence differs from that of the main trajectory, and observation disagreement is the analogous fraction for observations. New-action coverage (NSR) is the fraction of branches that contain an action absent from the main trajectory, and mean new actions counts such actions per branch. At step 100, EBR is 74.2% on ALFWorld and 47.7% on WebShop, and NSR is 55.5% and 37.5% (Table 14).

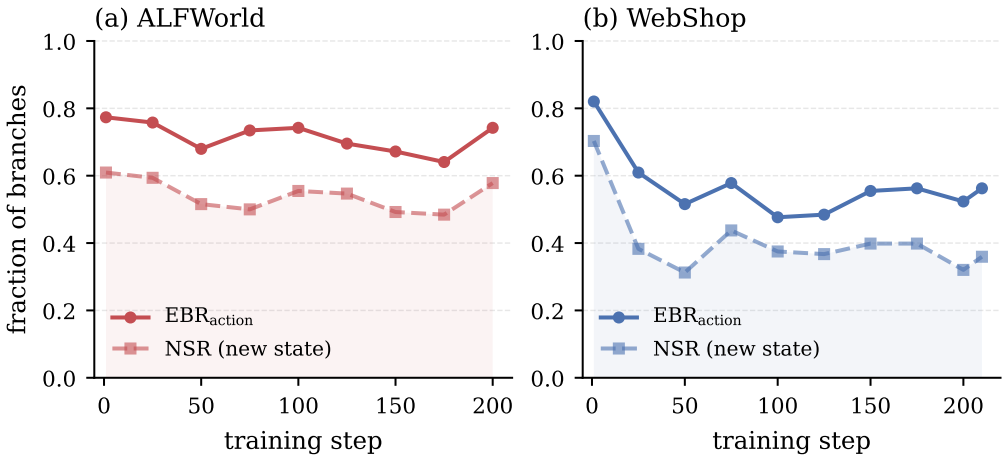


Figure 8: Behavior coverage of resampled branches over training. "New state" in the plot denotes new-action coverage (NSR). High disagreement and NSR mean that branches reach behavior not present in the main trajectory.

This coverage persists throughout training (Figure 8): on WebShop it stabilizes after an initial drop, and on ALFWorld it stays high.

C.3 Recovery opportunities

We define the rescue rate $R_{\uparrow} = \Pr(\text{some branch succeeds} \mid \text{main fails})$ and the harm rate $R_{\downarrow} = \Pr(\text{all branches fail} \mid \text{main succeeds})$. In the latter case, the successful main trajectory is still used for training.

Table 15: Recovery outcomes (%). $R_{\uparrow}$: at least one successful branch given a failed main; $R_{\downarrow}$: all branches fail given a successful main. The main trajectory is retained.

Task / setting	Main success (%)	$R_{\uparrow}$ (%)	$R_{\downarrow}$ (%)
Math B=3,k=2 @200	41.4	42.7	0.0
ALFWorld resample @100	48.4	21.2	19.4
ALFWorld resample @200	45.3	14.3	6.9
WebShop resample @200	37.5	5.0	4.2
Search-QA B=3, pooled dumps	51.9	15.8	15.2

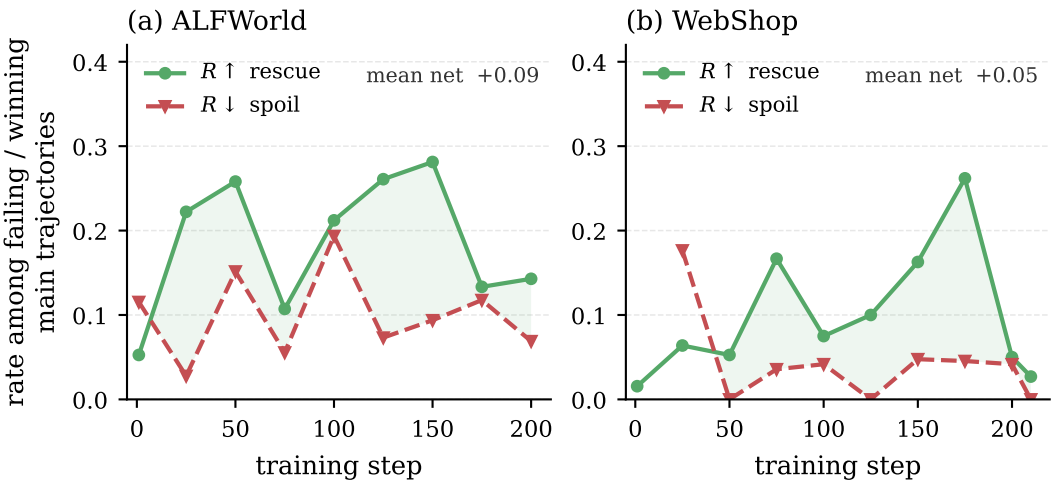


Figure 9: Rescue and harm rates during training, each conditioned on the outcome of the main trajectory. "Mean net" is $R_{\uparrow} - R_{\downarrow}$ averaged over training steps. "Spoil" means that all branches fail although the main trajectory succeeds; the main trajectory itself stays in the training batch.

The rescue rate ranges from 5.0% on WebShop to 42.7% on mathematics (Table 15), and successful alternatives keep appearing throughout training (Figure 9). Such recovered continuations provide corrective supervision and may contribute to the task gains.

C.4 WHERE BEHAVIOR BEGINS TO DIVERGE

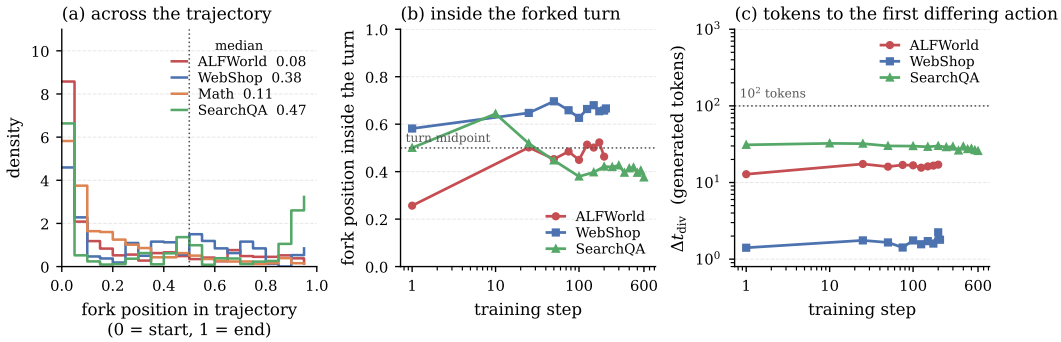


Figure 10: Fork location and distance to the first changed action. (a) Fork position relative to the full trajectory and (b) relative to the selected turn, where 0 and 1 mark the start and end. (c) Number of generated student tokens from the fork to the first changed action.

Fork locations depend on the task and often fall inside a turn (Figure 10). The first changed action appears roughly 12–18 generated tokens after the fork on ALFWorld, 1.5–2 on WebShop, and 30 on Search-QA, so a token-level fork can change later actions while reusing the preceding context.

D TRAINING AND EVALUATION SETTINGS

D.1 SHARED TRAINING CONFIGURATION

The default configuration uses $B = 3$ forks and $k = 2$ continuations per fork. All continuations are kept, including unsuccessful ones, and resampled branches receive no importance weights. Prefix deduplication masks the tokens strictly before the fork and keeps the fork token; environment observations are excluded from the distillation loss. Fork scores mix entropy and teacher–student disagreement with $\alpha = 0.5$, using a 128-token disagreement window and the top-20 student log probabilities. In agent tasks, each turn is scored by the mean of its top 20% eligible-token scores; high-scoring turns are selected with a minimum gap of one turn, and the fork is drawn uniformly from the three highest-scoring eligible tokens of each selected turn. Turn-opening eligibility covers the first 16 tokens of a turn, with no initial-token skip, and is not restricted to turns that follow a tool call. The minimum entropy for eligibility is 0 for agent tasks and 0.5 for mathematics.

Training uses the sampled-token (k_1) estimator of the reverse KL, i.e., the detached advantage $\mathrm{sg}[\Delta \log p_t]$, with PPO clipping ratio $\epsilon = 0.2$ and token-mean reduction. The teacher is queried for its top-64 log probabilities, and the loss uses the sampled-token log ratio. Recorded loss values are clamped to $[-10, 10]$, and log probabilities are floored at -10. Task rewards, KL rewards, and a separate actor KL penalty are disabled. Rollouts and branches use temperature 1, top-$p = 1$, and no top-k truncation. The learning rate is 10^{-6}. We train with FSDP, dynamic batching, offloading, gradient checkpointing, and asynchronous vLLM rollout that records log probabilities. For agent tasks, PBR uses an unlikelihood weight $\lambda = 0.01$ and a cap of $c_{\mathrm{PBR}} = 0.02$ on the weighted loss, and $\lambda = 0$ for the $-$PBR variants; the forbidden boundary token for Qwen models is 151668.

Table 16 lists the per-task settings. Prompt, response, per-turn, and PPO limits are in tokens; NR marks an unreported setting and N/A a setting that does not apply. Mathematics training runs for two epochs over 17,398 examples, i.e., $\lfloor 17{,}398/128 \rfloor = 135$ steps per epoch and 270 steps in total, with checkpointing and evaluation every 50 steps. For mathematics, rollout and teacher inference each run on a separate eight-GPU machine, with tensor-parallel degrees 1 and 2 and GPU memory fractions 0.45 and 0.65.

D.2 ADDITIONAL MATHEMATICS BASELINES

Offline knowledge distillation (KD). The teacher samples one response per training problem to form a fixed dataset. The student is trained on these sequences with an equally weighted (0.5 each)

Table 16: Training settings.

Setting	ALFWorld	WebShop	Search-QA	Mathematics
Training examples	6,400	6,782	NR	17,398
Train/PPO batch size	64/64	64/64	128/128	128/128
Prompt limit	2,048	4,096	2,048	2,048
Response limit	24,576	16,384	8,192	16,384
Per-turn tokens	512	512	512	N/A
Environment step limit	50	15	4	N/A
Environment pool	16	8	NR	N/A
Training duration	2 epochs	4 epochs	600 steps	2 epochs
Rollout/teacher TP	2/8	2/8	2/4	1/2
Rollout/teacher GPUs	8/8	8/8	4/4	8/8
Memory fractions	0.45/0.75	0.45/0.75	0.40/0.70	0.45/0.65
PPO token limit	32,768	24,576	16,384	49,152

sum of token cross-entropy and the forward KL from teacher to student, computed over the teacher's top-16 tokens.

Outcome-based GRPO. The student generates eight responses per problem and receives a binary answer-correctness reward; group-relative advantages computed from these rewards drive the GRPO updates. No teacher is used. KD and GRPO are trained without PBR in Table 1.

Entropy-aware OPD (EOPD). EOPD (Jin et al., 2026) adds a forward-KL term to reverse-KL OPD at tokens where the teacher entropy exceeds a threshold h_0:

$$\ell_{\mathrm{EOPD}}(s) = \ell_{\mathrm{OPD}}(s) + \beta\, \mathbf{1}\{H(\pi_T(\cdot \mid s)) > h_0\}\, \mathrm{KL}(\widetilde{\pi}_T \| \pi_\theta), \tag{28}$$

where ℓ_{OPD} is the reverse-KL surrogate, $\widetilde{\pi}_T$ is the teacher distribution renormalized over its top-k tokens, and β weights the added term. EOPD changes the local target but samples no additional suffixes. It also differs from Entropy-Weighted OPD, which only reweights the token losses of a single trajectory.

Sparse probing and outcome calibration (SPOT). SPOT (Qu et al., 2026) selects positions by normalized teacher entropy, teacher top-k probability mass, and teacher–student mismatch. At each selected position it forces teacher-proposed candidate tokens, lets the student continue, and scores the continuations with an outcome verifier. For positions with a positively valued candidate, it constructs the target

$$p_{\mathrm{cal}}(v \mid s) = \frac{\widetilde{\pi}_T(v \mid s) \exp(\gamma \widehat{V}(s, v))}{\sum_{u \in \mathcal{K}(s)} \widetilde{\pi}_T(u \mid s) \exp(\gamma \widehat{V}(s, u))}, \tag{29}$$

where $\mathcal{K}(s)$ is the teacher candidate set, $\widehat{V}$ the verifier-based continuation value, and γ the calibration strength. A cross-entropy loss toward this target is added to OPD. BranchOPD instead resamples from the student and applies dense teacher supervision along the new suffixes. All four additional baselines are trained without PBR in Table 1.

D.3 EOS ALIGNMENT FOR THE BASE STUDENT AND INSTRUCTION TEACHER

The official Qwen3-4B-Base configuration uses EOS token ID 151643 (`<|endoftext|>`), whereas Qwen3-8B uses ID 151645 (`<|im_end|>`) as its primary EOS and accepts both 151645 and 151643 in its generation configuration. In our runs, the student adopts the teacher's EOS configuration and stops on either token (`eos_token_id = [151645, 151643]`). This overrides the base student's single-token default, so teacher and student share the same end-of-response convention for both teacher probabilities and terminal supervision.

D.4 EVALUATION PROTOCOL

For mathematics, we sample 16 answers per problem with temperature 0.6, top-$p = 0.95$, and top-$k = 20$, with prompt and response limits of 2,048 and 16,384 tokens. For agent tasks, we sample

one trajectory per task at temperature 0.4. The model's optional thinking mode is disabled in the generation configuration; protocol tags written in the prompt are handled separately. WebShop uses the small environment with human interaction disabled and seed 42, and holds out 128 of its 6,910 tasks. The WebShop and Search-QA teachers are taken after 125 and 500 GRPO steps, respectively. Search returns three results per call with a 30-second timeout.

ALFWorld SR is computed separately on the 140 ID and 134 OOD tasks, and the ALFWorld aggregate is their unweighted mean. WebShop SR is the fraction of the 128 tasks solved, and Search-QA SR is exact-match accuracy on the same mixed validation set for every model. The agent Avg. SR is the unweighted mean of the ALFWorld aggregate, WebShop SR, and Search-QA SR; mathematics Avg is the unweighted mean of the three benchmark accuracies.

E Benchmarks and Additional Background

E.1 Mathematical reasoning

MATH contains competition problems in algebra, geometry, number theory, and related subjects (Hendrycks et al., 2021). We evaluate on MATH-500, the 500-problem test subset used in *Let's Verify Step by Step* (Lightman et al., 2023), and on AIME 2024 and 2025. Each AIME year has 30 problems from two 15-problem examinations, with integer answers from 000 to 999. Each problem requires a free-form solution and a final answer. avg@16 averages correctness over 16 independent solutions per problem and then over problems. Training problems come from DAPO (Yu et al., 2025).

E.2 ALFWorld

ALFWorld (Shridhar et al., 2020) is a text-based household environment that grounds language actions in embodied tasks. The agent reads textual observations and issues commands to navigate, inspect, pick up, place, and transform objects, and succeeds if it completes the goal within the step limit. We report success on 140 in-distribution (ID) and 134 out-of-distribution (OOD) tasks, and their unweighted mean. During branch training, the selected interaction state is restored, or the preceding actions are replayed, before generation continues from a token-level fork; observations serve as context and are not supervised.

E.3 WebShop

WebShop (Yao et al., 2022a) is a simulated online-shopping benchmark. Given a purchase instruction, the agent searches a catalog, browses product pages, selects options, and buys an item. We report success rate (SR) on 128 held-out tasks.

E.4 Search-QA

Retrieval-augmented question answering answers information-seeking questions from retrieved evidence (Kwiatkowski et al., 2019). In our Search-QA setting, a trajectory interleaves search queries, retrieved results, and a final answer. SR is the fraction of questions whose normalized final answer exactly matches an accepted answer. Each search returns up to three results, and at most four interaction steps are allowed. The teacher, the untrained student, and all trained models are evaluated on the same mixed validation set.

E.5 Benchmarks discussed as background

GSM8K (Cobbe et al., 2021) contains grade-school math word problems, and Minerva (Lewkowycz et al., 2022) studies quantitative reasoning with language models. ScienceWorld (Wang et al., 2022a) tests science-oriented agent reasoning, WebArena (Zhou et al., 2023b) evaluates agents in reproducible web environments, and AgentGym (Xi et al., 2024) covers a range of agent tasks. HotpotQA (Yang et al., 2018), TriviaQA (Joshi et al., 2017), and MuSiQue (Trivedi et al., 2021) are question-answering benchmarks, some requiring multi-hop evidence.

E.6 Additional learning and systems context

Context distillation, self-distillation, and privileged supervision are other ways to transfer information into model parameters (Ye et al., 2026; Zhao et al., 2026; Penaloza et al., 2026). Recent work also studies reward extrapolation, skill transfer, hybrid supervision, and teacher retirement (Yang et al., 2026a; Wang et al., 2026a; Lu et al., 2026; Yu et al., 2026a). PPO and group-relative optimization are the underlying optimizers (Schulman et al., 2017; Shao et al., 2024; DeepSeek-AI et al., 2025), and agent training systems such as GiGPO, ArCHer, Agent Lightning, WebRL, and Search-R1 optimize task rewards (Feng et al., 2025; Zhou et al., 2024; Luo et al., 2025; Jin et al., 2025; Qi et al., 2024), unlike the protocol-event loss used here. Our experiments use the Qwen3 and Qwen2.5 families, and Llama 3 is a related open model family (Yang et al., 2025; Grattafiori et al., 2024; Qwen et al., 2024). HybridFlow (Sheng et al., 2024), FlashAttention (Dao et al., 2022), and PagedAttention (Kwon et al., 2023) improve execution and memory efficiency, complementing our reduction in generated student tokens. Continual on-policy distillation studies supervision and training distributions that change over time (Zheng et al., 2026); our analysis instead fixes the rollout policy within each sampling group.

F TRAINING ALGORITHM

Algorithm 1 Branching on-policy distillation

Require: Student π_θ, teacher π_T, task source, fork budget B, continuation budget k

for each training iteration **do**
  Freeze the rollout policy $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$ for this sampling group
  Sample a task and generate one main trajectory under $\pi_{\theta_{\text{old}}}$
  Compute eligible-token entropy and teacher–student disagreement scores
  if mathematical reasoning **then**
    Select up to B token-level fork positions
  else
    Select turns, then select eligible fork tokens within those turns
  end if
  Initialize training tokens with the main trajectory's generated tokens
  for each selected fork prefix s_b **do**
    for $j = 1, \ldots, k$ **do**
      Restore the fork state or replay the prefix actions for this branch, if needed
      Independently sample a continuation from $\pi_{\theta_{\text{old}}}(\cdot \mid s_b)$
      Keep its generated tokens from the fork onward, whatever its outcome
    end for
  end for
  Query teacher log probabilities for retained sampled tokens
  Exclude observations and duplicate branch-prefix tokens from the loss
  Form stopped-gradient sampled-token advantages and the clipped token-mean surrogate
  Construct applicable protocol-boundary events and add the PBR loss
  Update θ using the combined objective
end for

If no eligible fork is found, the iteration trains on the main trajectory alone. All sampled branches contribute regardless of task success. At evaluation, agents generate one trajectory per task, and mathematics uses 16 independent responses per problem.

G LIMITATIONS

Despite achieving strong performance across diverse reasoning and agent benchmarks, BranchOPD has several limitations. First, our method relies solely on teacher supervision and does not incorporate task rewards; combining OPD with reward-based learning may provide complementary signals when teacher guidance is imperfect (Tan et al., 2026; He et al., 2026). Second, we do not explicitly control the training curriculum. In multi-turn tasks, trajectories with early unrecoverable mistakes may provide limited useful supervision, suggesting that sampling hard but recoverable states could further improve training efficiency (Wang et al., 2026b). Finally, branching in agent environments requires replaying actions or restoring intermediate states, introducing additional environment-side overhead and limiting applicability when actions are irreversible or the external state changes over time.

H PROMPT TEMPLATES

The figures below give the prompts used in training and evaluation. Mathematics uses a single-turn template with a boxed final answer. ALFWorld and WebShop separate the initial task instruction from later observations and require structured action outputs. Search-QA additionally specifies the format of retrieval results and the feedback for invalid actions. These templates define the output boundaries that PBR checks.

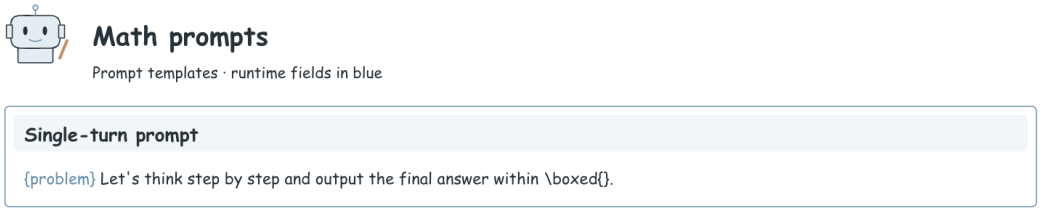


Figure 11: Mathematical reasoning prompt templates.

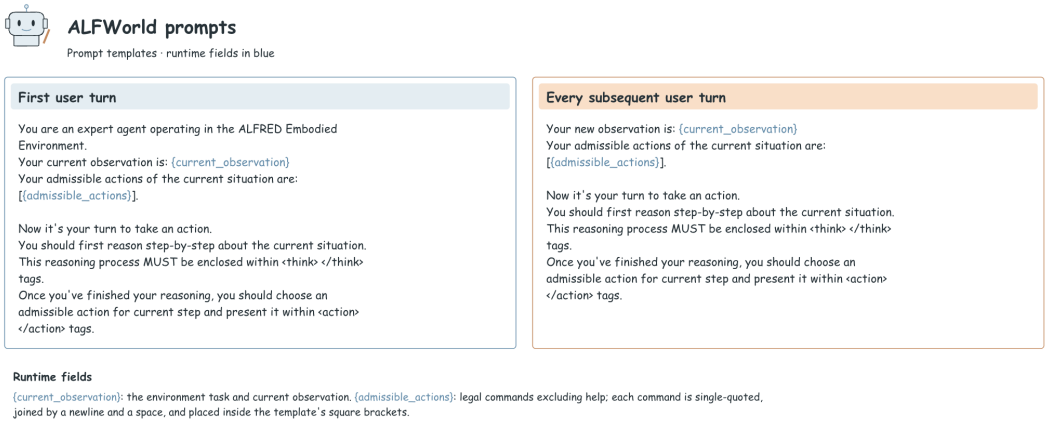


Figure 12: ALFWorld prompt templates.

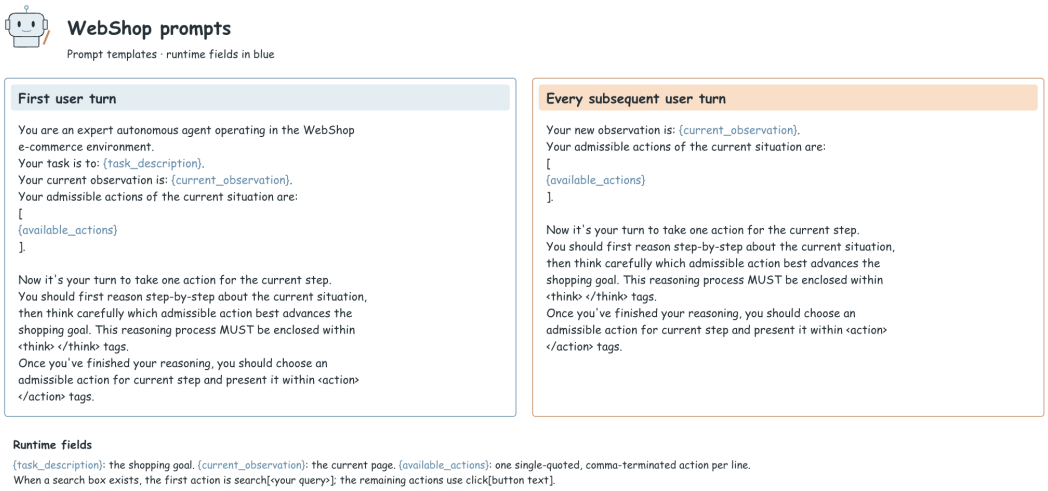


Figure 13: WebShop prompt templates.

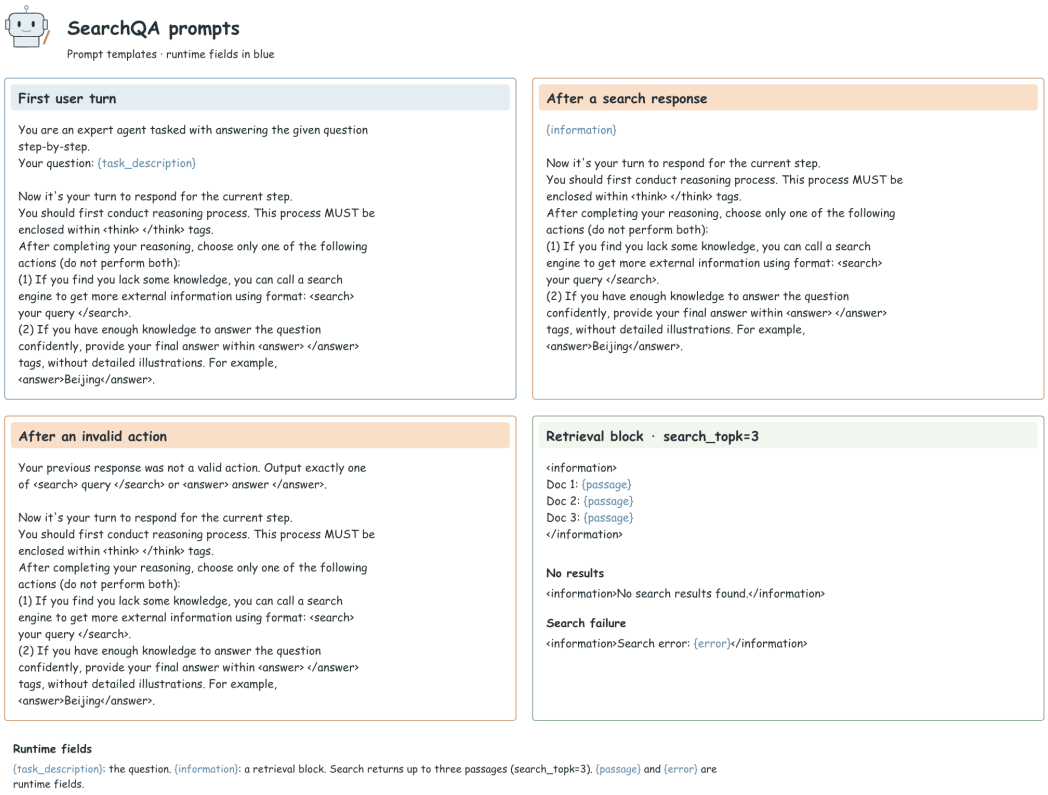


Figure 14: Search-QA prompt templates.