SAME CANDIDATES, STRONGER REASONING: RETHINKING WHAT POST-TRAINING LEARNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training transforms pretrained language models into notably stronger reasoners. Yet recent evidence suggests that the changes driving these gains are concentrated at a small number of token-level decisions. This raises a fundamental puzzle: at these consequential decisions, does post-training rely on candidates beyond the pretrained model's high-ranked set, or primarily reorganize preferences among them? Across 31 model–task–checkpoint settings, 92.9% of post-training Top-1 changes remain within the pretrained Top-16 on average, while their relative preferences are substantially reorganized. More strikingly, this inheritance persists even on successful post-training-induced prefixes that are not visited by the pretrained policy. Accumulated local preference changes can therefore compose into genuinely new global competence. Motivated by this finding, we introduce ProQ, a post-training paradigm that shifts the learning target from full-distribution next-token adaptation to value-guided selection over the model's own proposals. Within a single LLM and shared vocabulary, standard-token logits retain normal reasoning and candidate proposal, regularized toward the pretrained model; reserved-token logits are repurposed to produce candidate-conditioned value scores. These scores provide a natural RL interface for local selection. Trained only on mathematical reasoning, ProQ achieves strong gains on math benchmarks and transfers to out-of-domain coding and knowledge tasks; the benefits persist across different model sizes and families.

1 INTRODUCTION

Post-training has become a central stage for transforming pretrained language models into strong reasoning models, yielding substantial improvements in instruction following, task generalization, and challenging reasoning benchmarks (Ouyang et al., 2022; Chung et al., 2024; Guo et al., 2025). Modern post-training pipelines typically begin with supervised fine-tuning and further refine behavior through preference-based or reinforcement-learning objectives (Rafailov et al., 2023; Shao et al., 2024; Yu et al., 2025; Yang et al., 2025). These advances have motivated growing interest in understanding what post-training actually changes in pretrained models.

To understand these changes, prior work has studied them at both the sequence and token levels. At the sequence level, studies probe reasoning boundaries through large-budget sampling and trajectory-level success, and characterize how post-training reshapes the structure, search behavior, and diversity of generated reasoning trajectories (Chen et al., 2025; Wen et al., 2026; Liu et al., 2025; Guo et al., 2025; Nguyen et al., 2025; Karouzos et al., 2026). More recent work localizes consequential changes to individual token positions. Meng et al. (2026) show that reinforcement learning with verifiable rewards (RLVR) relies on *sparse but critical* policy changes concentrated at a small fraction of decisions, which account for a disproportionate share of reasoning gains. Related studies connect these effects to token uncertainty and high-entropy decision points (Wang et al., 2025; Cheng et al., 2026), while others investigate token-level credit assignment and selective policy optimization (He et al., 2026; Akgül et al., 2026). Together, these analyses clarify *where* post-training changes the policy and *which* local decisions matter.

The sparsity of these effects raises a deeper question: what does post-training actually learn at the few decisions that matter most? At each reasoning state, we define the *Base candidate set* as the candidates ranked highly by the pretrained policy. One possibility is that post-training increasingly

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.

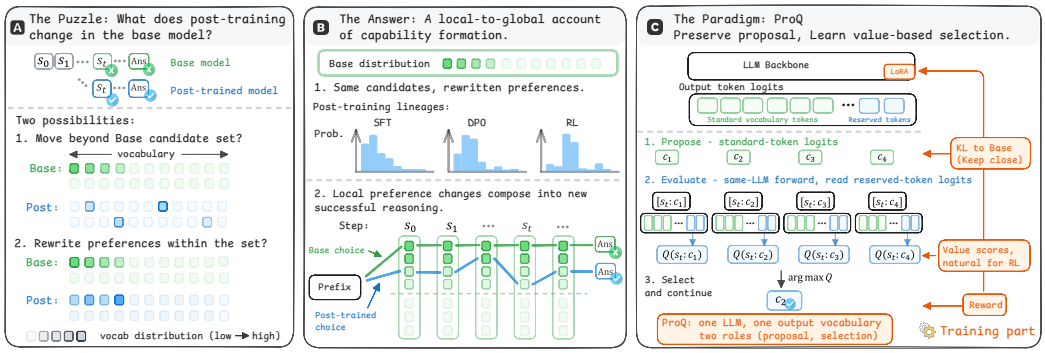


Figure 1: **Same candidates, stronger reasoning.** (A) Does post-training improve reasoning by moving beyond the Base candidate set, or by rewriting preferences within it? (B) We find that post-training largely preserves candidate availability while progressively reorganizing local preferences; accumulated local changes then compose into globally different successful trajectories. (C) This motivates PROQ: rather than adapting the full next-token policy, ProQ separates proposal from evaluation. Standard-token logits preserve candidate proposal under Base regularization, while reserved-token logits produce candidate-conditioned scores for value-guided selection.

draws its preferred decisions from lower-ranked regions of the pretrained policy; another is that the relevant alternatives already lie within the Base candidate set, with training primarily rewriting their relative preferences. These possibilities point to different sources of reasoning improvement. We therefore ask: *How does post-training acquire stronger global reasoning: by expanding beyond the Base candidate set, or by rewriting preferences within it?*

Our analyses reveal a striking local-to-global asymmetry in how post-training transforms a reasoning policy. Across complete post-training lineages spanning supervised fine-tuning, preference optimization, and reinforcement learning, the actions preferred after training remain largely within the Base candidate set. Their relative preferences, however, change far more substantially. More importantly, candidate inheritance is not confined to states shared with the Base policy: after the post-trained policy branches onto a different reasoning path, the subsequent decisions along that path still remain largely within the Base candidate set. This suggests a local-to-global account of capability formation:

> Post-training primarily rewrites preferences over inherited candidates, and autoregressive generation composes these local preference changes into globally different reasoning trajectories.

This finding suggests a different post-training principle: when pretraining already supplies useful local alternatives, the missing component may be *evaluation rather than proposal*. We develop this principle into *Proposal-preserving learning (ProQ)*, which functionally separates candidate proposal from candidate evaluation without separating the model. Within the same LLM and shared vocabulary, standard-token logits retain normal reasoning and candidate proposal, while reserved-token logits produce candidate-conditioned scores used for value-guided selection. Because supervision comes from eventual task outcomes, these scores provide an RL-compatible signal for relative candidate selection. Concretely, ProQ learns by ranking successful candidates above unsuccessful ones and explicitly separating the successful candidate from competing failures. In parallel, a standard-token KL anchor keeps proposal behavior close to the pretrained policy, concentrating adaptation on candidate selection rather than broad next-token reshaping.

Our contributions are summarized as follows:

- **Same candidates, rewritten preferences.** Across models, tasks, and complete post-training lineages, we show that preferred decisions remain largely within the Base candidate set while their relative preferences are substantially reorganized.
- **Local changes, global capability.** On Base-fail/Post-success trajectories, we show that accumulated preference changes over inherited candidates compose into globally different successful reasoning, separating local candidate availability from global competence.

- **Proposal-preserving learning.** We introduce ProQ, which separates proposal from candidate evaluation within the same LLM, learns reward-grounded candidate scoring for value-guided selection, and keeps standard-token proposal behavior close to Base.
- **Broad gains and transfer.** Across three model families, ProQ consistently improves reasoning performance and transfers from mathematical training to out-of-domain coding and knowledge tasks.

2 Related Work

Token-level analysis. Recent work has shown that reasoning post-training can depend on a surprisingly small number of consequential local decisions. High-entropy positions have been identified as reasoning forks that receive important policy updates, motivating selective optimization at uncertain tokens (Wang et al., 2025; Cheng et al., 2026). Meng et al. (2026) further show that RLVR can rely on *sparse but critical* policy changes concentrated at a small fraction of decisions, while related work studies token-level credit through reward-conditioned information or localized failure signals (He et al., 2026; Ko et al., 2026). Together, these studies establish that large reasoning gains arise from highly localized policy changes, but do not determine what is being learned at those changes.

Candidate sets and capability formation. A complementary line of work suggests that useful reasoning decisions can already lie among a model's highly ranked candidates. Select to Think identifies cross-model *local sufficiency*, where a stronger model's preferred token often appears in a weaker model's Top-K candidates (Ye et al., 2026). ReasonMaxxer similarly observes that many beneficial RL corrections promote candidates favored by the pretrained model (Akgül et al., 2026), while RL with Promising Tokens restricts optimization to candidates selected from the Base model's semantic prior (Pang et al., 2026). These works establish candidate locality as an empirical property or an optimization prior, but leave unresolved what this locality implies about capability formation.

Position. Our work asks a more fundamental question: *where does new reasoning capability reside when many of the required local decisions are already present before post-training?* We show that such capability can emerge at the level of state-dependent organization and autoregressive composition, rather than requiring correspondingly novel local decisions.

3 How Does Post-Training Reorganize Pretrained Candidates?

Prior work shows that post-training can yield large reasoning gains through surprisingly sparse local changes. Yet sparsity alone does not reveal what is learned at these consequential decisions: does post-training increasingly rely on candidates that the pretrained model ranks weakly, or mainly reorganize those it already ranks highly? We answer this by tracing candidate identity and preference allocation across complete post-training lineages and reasoning trajectories. The main text uses complementary representative lineages, while Appendix C provides broader cross-model and cross-domain validation.

3.1 Where do post-training's preferred candidates come from?

For brevity, *Base* denotes the pretrained checkpoint preceding post-training. We compare the Base and post-trained policies. When their Top-1 choices disagree, we record where the post-trained choice was ranked by Base, and then extend the analysis to the post-trained Top-4/Top-8 candidate set.

Figure 2 visualizes this disagreement-conditioned analysis on DeepSeekMath-7B, comparing its Instruct and RL checkpoints against the same Base model. Each point corresponds to a token position where post-training changes the Base top-1 prediction. Its ring indicates where the new winner ranked under the Base model (i.e., *Base rank*), and color indicates the Base model's local uncertainty at that decision.

Post-training largely operates within the Base candidate set. Despite their different training objectives, both Instruct and RL largely favor actions already contained in the shallow Base candidate set: over 91% of changed decisions fall within Base Top-8 and over 96% within Top-16. This inheritance extends beyond the selected token. Across models and tasks, a substantial fraction of the post-trained Top-4/Top-8 candidate set is already recovered from relatively shallow Base support

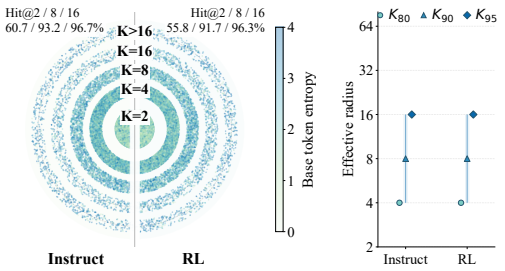


Figure 2: *Post-training decisions remain within the Base candidate set.* Each point marks a Base–Post top-1 disagreement, with its ring indicating the Base rank of the post-trained choice and color showing Base uncertainty. Despite their different post-training objectives, both Instruct and RL remain strongly concentrated in shallow Base ranks: over 96% of changed decisions fall within Base Top-16.

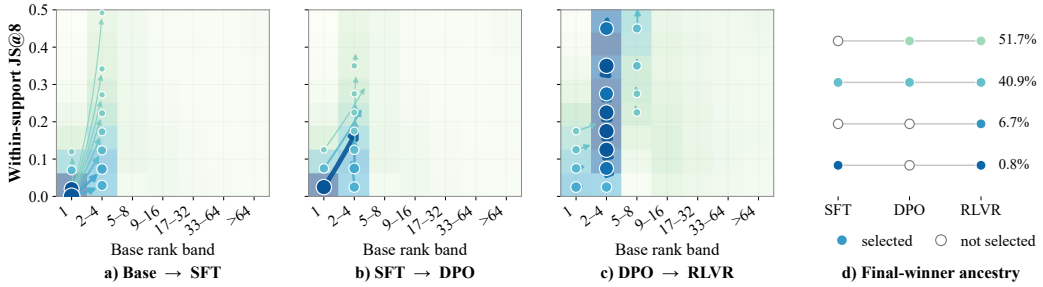


Figure 3: **Post-training primarily strengthens preference rewriting within an inherited candidate space.** Shown for the OLMo-2-7B Base→SFT→DPO→RLVR lineage on OpenThoughts-Math. Across stages, preferred tokens remain concentrated in shallow Base-rank bands, while within-support preference deviation from Base grows substantially. The final panel shows that most eventual RLVR winners are promoted before RLVR, indicating progressive promotion across stages rather than abrupt late-stage introduction.

(Figure A4). This inheritance exhibits a graded effective radius, spanning shallow Top-8/Top-16 regimes through Top-32–64 and deeper long-tail cases.

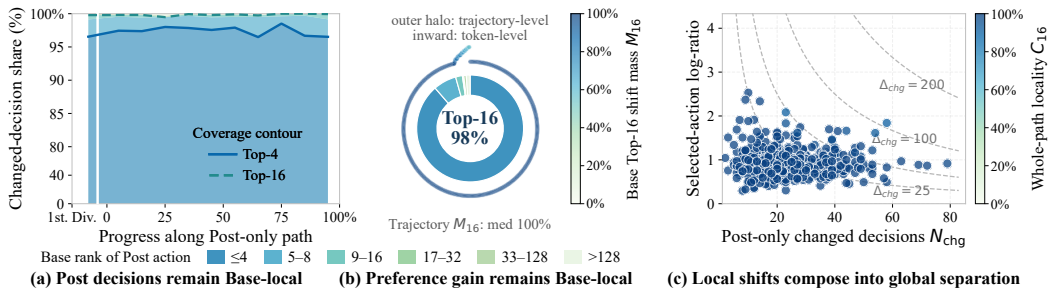
The same pattern holds across Qwen2.5 and OLMo-2, spanning supervised, preference-based, and RL stages as well as mathematical and general-domain tasks (Appendix C.1). Together, these show that post-training largely reorganizes a candidate structure already exposed by the pretrained model.

3.2 What changes within the inherited candidate set?

We next ask what successive post-training stages change when preferred candidates remain largely inherited. Figure 3 traces the OLMo-2-7B Base→SFT→DPO→RLVR lineage on OpenThoughts-Math. The horizontal axis records the Base rank of each stage's preferred token, while the vertical axis measures Base-relative preference reorganization within the fixed Base Top-8 support using Jensen–Shannon divergence. Background shading shows the density of disagreement states, circles summarize states with similar coordinates, and arrows trace their movement across successive stages.

Post-training progressively reorganizes preferences within the Base candidate set. As shown in Figure 3, the two axes evolve very differently across training. Preferred candidates remain concentrated in shallow Base-rank regions, while their probability allocation moves progressively farther from Base. Within-Top-8 JS divergence roughly doubles from SFT to RLVR, while Base Top-16 coverage remains above 95% throughout. A similar separation appears on TruthfulQA (Lin et al., 2022) (Figure A7), indicating that successive stages primarily reorganize preferences among inherited candidates rather than increasingly draw decisions from deeper Base ranks.

Final decisions emerge through progressive promotion. The final panel tracks when the eventual RLVR choice first becomes Top-1 across stages. Only 6.7% first become Top-1 at RLVR, whereas 40.9% are already selected from SFT onward and 51.7% are at DPO. Together with their shallow Base ranks, this shows that many final decisions are progressively promoted across training rather than introduced abruptly at the final stage. Corresponding rank trajectories are reported in Appendix C.2–C.3.



(a) Post decisions remain Base-local **(b) Preference gain remains Base-local** **(c) Local shifts compose into global separation**

Figure 4: **Candidate inheritance persists beyond the Base trajectory.** For Gain examples where Base fails and Post succeeds, we replay Base on the Post-generated trajectory. Panel (a) shows that changed Post decisions remain concentrated in shallow Base ranks even along these new prefixes, and Panel (b) shows that 98% of the positive Post–Base preference shift falls within Base Top-16. Panel (c) shows how moderate Base-local preference changes accumulate across reasoning steps into large trajectory-level policy separation.

3.3 Can globally new reasoning emerge from inherited local decisions?

Previous analyses establish local candidate inheritance along aligned reasoning states. To move beyond shared-prefix analysis, we turn to *Gain* examples, where the Base model fails but the post-trained model succeeds. After the first divergence, the post-trained policy visits prefixes that are not encountered along the Base trajectory, allowing us to test whether candidate inheritance persists in states induced by the post-trained policy itself.

Candidate inheritance persists beyond the Base trajectory. Figure 4 reports the analysis for Qwen2.5-Math-1.5B on OpenThoughts-Math. Panel (a) tracks where changed Post decisions rank under Base throughout the post-induced trajectory. Base Top-16 coverage remains near saturation, showing that the successful path continues to rely largely on the Base candidate set at those new prefixes. Panel (b) reaches the same conclusion from preference change: 98% of the positive Post–Base shift is concentrated on Base Top-16 candidates. This links inherited local support to globally different reasoning behavior.

Accumulated local changes compose into global success. Panel (c) asks how these local changes accumulate along each Gain trajectory. The horizontal axis gives the number of post-divergence decisions that differ from Base, while the vertical axis gives the average preference increase on the Post-selected actions. Most trajectories combine many changed decisions with only moderate per-decision shifts, while retaining high Top-16 locality. This pattern shows that large trajectory-level policy differences are often built through the accumulation of many moderate preference changes over Base-local actions, rather than a few extreme local revisions. The same qualitative pattern holds across additional model families, scales, and benchmarks; see Appendix C.4.

4 ProQ: Proposal-Preserving Learning

Conventional post-training directly adapts the next-token policy to favor desirable decisions. Our analysis suggests a different target: when useful candidates are already present in the pretrained policy, the key problem is not to generate new ones, but to identify which should be preferred. ProQ therefore preserves standard-token proposal behavior while learning candidate-conditioned value for selection among the model's own proposals.

4.1 Learning to select rather than relearning to propose

At a consequential reasoning state s_t, let $\mathcal{C}_K(s_t) = \{c_1, \ldots, c_K\}$ denote the K candidates proposed by ProQ's current standard-token policy. Rather than further reshaping the next-token distribution to promote specific candidates, ProQ assigns each proposal a candidate-conditioned value score and selects

$$Q^\star(s_t, c_j) = \mathbb{E}[R \mid s_t, c_j], \qquad c^\star = \arg\max_{c_j \in \mathcal{C}_K(s_t)} Q_\phi(s_t, c_j), \tag{1}$$

where $Q^\star(s_t, c_j)$ denotes the downstream value of taking c_j at state s_t, and R is the eventual task outcome. As deployment depends only on $\arg\max_c Q_\phi(s, c)$, ProQ trains the score to preserve the success-induced ordering among candidates. This provides a natural RL interface: trajectory-level rewards directly supervise the relative value ordering of local candidate decisions.

One LLM, one vocabulary, two roles. Many contemporary LLM vocabularies include reserved or otherwise unused token IDs (Grattafiori et al., 2024; Land & Bartolo, 2024; Team et al., 2024; Yang et al., 2025). Following prior work on repurposing such outputs for auxiliary prediction (Manvi et al., 2026; Ye et al., 2026), we divide the shared vocabulary conceptually into standard tokens $\mathcal{V}_{\text{std}}$ and a small reserved subset $\mathcal{V}_{\text{res}} = \{r_1, \ldots, r_B\}$. Standard-token logits retain their usual role in autoregressive reasoning and candidate proposal, while the reserved-token logits are repurposed for candidate evaluation.

For each proposal c_j, we append the candidate to the current prefix and perform a one-token branch forward on $[s_t; c_j]$ through the same LLM. The model then produces logits over the full shared vocabulary; ProQ reads only the positions indexed by $\mathcal{V}_{\text{res}}$ and interprets them over ordered value bins:

$$p^{\text{res}}_{\phi,b}(s_t, c_j) = \frac{\exp(z^{\text{res}}_{\phi,b}(s_t, c_j)/T_{\text{bin}})}{\sum_{b'=1}^{B} \exp(z^{\text{res}}_{\phi,b'}(s_t, c_j)/T_{\text{bin}})}, \qquad Q_\phi(s_t, c_j) = \sum_{b=1}^{B} v_b \, p^{\text{res}}_{\phi,b}(s_t, c_j), \tag{2}$$

where $v_b = (b-1)/(B-1)$ spans $[0, 1]$. The reserved-token distribution therefore forms a discrete success-value representation.

Crucially, proposal and evaluation are functionally separated without separating the model: the same vocabulary output produces standard-token logits for generation and reserved-token logits for candidate scoring. ProQ can therefore learn a new reward-grounded decision signal while preserving the language model's ordinary reasoning pathway.

4.2 ProQ learning

ProQ learns candidate-value ordering from downstream outcomes. For each training state s and candidate c_j, we force c_j as the next token and let a fixed Base continuation policy complete the reasoning trajectory. Repeating this intervention M times yields

$$R(s, c_j) = \frac{1}{M} \sum_{m=1}^{M} r^{(m)}(s, c_j),$$

where $r^{(m)}(s, c_j) \in \{0, 1\}$ indicates final-answer correctness. Thus, trajectory-level reward provides candidate-specific supervision for learning relative value among local alternatives.

We train the selector with two complementary objectives:

$$\mathcal{L}_{\text{select}} = \mathcal{L}_{\text{pair}} + \lambda_{\text{top1}} \mathcal{L}_{\text{top1}}. \tag{3}$$

The pairwise term ranks successful candidates above unsuccessful ones within the same state, while the Top-1 term explicitly separates the strongest successful candidate from its strongest unsuccessful competitor:

$$\ell_{\text{pair}}(s) = \operatorname*{mean}_{i \in \mathcal{P}(s),\, j \in \mathcal{N}(s)} \log\left(1 + \exp\frac{q_j - q_i}{T_{\text{pair}}}\right), \qquad \ell_{\text{top1}}(s) = \frac{[m - (q^\star_+ - q^\star_-)]_+}{T_{\text{top1}}}. \tag{4}$$

Here, $q_i = Q_\phi(s, c_i)$; $\mathcal{P}(s)$ and $\mathcal{N}(s)$ denote successful and unsuccessful candidates; and $q^\star_+ = \max_{i \in \mathcal{P}(s)} q_i$ and $q^\star_- = \max_{j \in \mathcal{N}(s)} q_j$ are their highest predicted values. T_{pair} and T_{top1} control the corresponding loss scales, while m specifies the desired Top-1 margin.

Because the same adapted LLM also produces candidate proposals, value learning can perturb its standard-token behavior. We therefore anchor the standard-token distribution to the Base policy, excluding reserved tokens from the constraint:

$$\mathcal{L}_{\text{propose}} = D_{\text{KL}}\left(\pi^{\text{std}}_B(\cdot \mid s) \,\|\, \pi^{\text{std}}_\phi(\cdot \mid s)\right). \tag{5}$$

The final objective directly mirrors the proposal–evaluation factorization of ProQ:

$$\mathcal{L}_{\text{ProQ}} = \mathcal{L}_{\text{select}} + \lambda_{\text{prop}} \mathcal{L}_{\text{propose}}. \tag{6}$$

ProQ therefore learns candidate-conditioned value ordering while keeping the standard-token proposal behavior close to the pretrained policy.

4.3 INFERENCE

ProQ invokes value-based selection only at high-uncertainty reasoning decisions. During generation, we compute the entropy H_t of the standard-token distribution and use a fixed, label-free threshold τ to determine whether selection is needed. When $H_t < \tau$, generation follows the model's standard decoding rule $\mathrm{Decode}(\cdot)$. At a triggered state, ProQ takes the current Top-K proposals, performs parallel one-token branch forwards through the same LLM, and maps the resulting reserved-token logits to candidate-conditioned scores $Q_\phi(s, c)$. The highest-scoring candidate is selected, and autoregressive generation continues from that decision:

$$c_t = \begin{cases} \arg\max\limits_{c \in \mathcal{C}_K^\phi(s_t)} Q_\phi(s_t, c), & H_t \geq \tau, \\ \mathrm{Decode}\big(\pi_\phi^{\mathrm{std}}(\cdot \mid s_t)\big), & H_t < \tau. \end{cases} \tag{7}$$

Algorithm 1 summarizes the complete procedure. ProQ therefore requires neither a separate value model nor candidate-level trajectory generation; additional computation arises only from the unselected one-token branches at the sparse triggered positions.

5 EXPERIMENTS

We evaluate PROQ through three research questions:

- **RQ1 (Overall performance):** How does PROQ perform across models and reasoning domains?
- **RQ2 (Selection efficacy):** Does ProQ learn reliable candidate-value rankings for local selection?
- **RQ3 (Design and efficiency)**: Which components drive PROQ's gains, and at what cost?

5.1 SETUP

We outline key configurations below, with complete implementation details provided in Appendix B.

Models, Training, and Evaluation. We evaluate three models spanning Qwen- and Llama-based architectures and multiple scales: DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025), Llama-3.2-3B (Grattafiori et al., 2024), and Qwen3-8B (Yang et al., 2025). All trainable methods are post-trained only on mathematical reasoning problems from OpenThoughts-Math (Guha et al., 2026), using the same outcome-based correctness reward. We evaluate mathematical reasoning on MATH-500 (Lightman et al., 2024), OlympiadBench (He et al., 2024), and AIME25 (Zhang & Math-AI, 2025); and out-of-domain transfer on coding with LiveCodeBench (Jain et al., 2025), and on general reasoning and knowledge with MMLU-Pro (Wang et al., 2024) and GPQA Diamond (Rein et al., 2023). Within each model family, all methods use the same zero-shot prompts and decoding protocol.

Baselines. We compare PROQ with both full-policy and candidate-restricted post-training. *GRPO* (Shao et al., 2024) and *DAPO* (Yu et al., 2025) optimize the full next-token policy, while *RLPT* (Pang et al., 2026) restricts RL to promising model-proposed candidates and provides the closest comparison to ProQ's proposal-preserving formulation. All trainable methods start from the same checkpoint and use the same training problems and reward; training-cost comparisons account for their respective rollout and optimization budgets.

ProQ Configuration. We use DeepSeek-R1-Distill-Qwen-1.5B as the representative configuration and report model-specific settings in Appendix B. ProQ uses $K = 8$ candidate proposals and $B = 16$ reserved tokens, with rank-16 LoRA (scaling 32, dropout 0.05); in total, it updates 18.49M parameters (1.04% of Base), including 24.6K reserved-token outputs. Training uses AdamW with peak learning rate 2×10^{-5}, batch size 48, and $\lambda_{\mathrm{prop}} = 30$. At inference, ProQ invokes the selection only when token entropy exceeds the 97.5th percentile of a label-free calibration distribution, then selects the highest-scoring candidate from the current Top-8 set.

5.2 RQ1: OVERALL PERFORMANCE

Mathematical reasoning. Across all three backbones, ProQ consistently improves mathematical reasoning and is best or tied on eight of the nine model–benchmark settings in Table 1. The gains are most pronounced on the more challenging OlympiadBench and AIME25 benchmarks, where ProQ substantially exceeds trajectory-matched policy-optimization baselines. This suggests that explicitly

Table 1: **ProQ yields broad reasoning gains that transfer beyond the training domain.** Across three backbones and six benchmarks, ProQ improves mathematical reasoning while achieving the strongest performance on all coding and broad-knowledge evaluations, despite being trained exclusively on mathematical reasoning. Results are averaged over three decoding seeds.

	Mathematical Reasoning			Out-of-Domain Generalization		
Post-training	**MATH**	**OlympiadBench**	**AIME25**	**LiveCodeBench**	**MMLU-Pro**	**GPQA-D**
DeepSeek-R1, 1.5B						
Base	65.4	33.7	15.6	14.3	23.0	10.1
GRPO	70.0	38.5	18.9	18.2	27.4	14.9
DAPO	65.5	35.2	20.0	14.2	26.0	14.1
RLPT	68.9	36.3	17.8	17.4	25.8	10.6
ProQ	**77.2**$_{\pm 2.0}$	**45.4**$_{\pm 0.6}$	**23.3**$_{\pm 3.3}$	**18.8**$_{\pm 0.5}$	**28.8**$_{\pm 1.1}$	**19.8**$_{\pm 1.6}$
Llama-3.2, 3B						
Base	44.5	24.4	2.2	13.1	42.1	23.7
GRPO	44.1	25.0	**3.3**	13.0	41.4	23.7
DAPO	44.5	26.1	1.1	11.3	42.5	23.2
RLPT	45.2	26.4	2.2	13.3	42.8	24.7
ProQ	**47.8**$_{\pm 1.8}$	**27.7**$_{\pm 2.9}$	**3.3**$_{\pm 0.0}$	**16.2**$_{\pm 1.6}$	**45.1**$_{\pm 1.6}$	**27.2**$_{\pm 1.4}$
Qwen3, 8B						
Base	81.3	68.6	63.3	44.6	70.6	51.8
GRPO	81.9	69.2	63.3	43.3	75.3	55.0
DAPO	81.6	68.6	60.0	45.0	73.2	51.5
RLPT	**82.3**	68.9	60.0	42.7	73.4	49.0
ProQ	82.1$_{\pm 1.3}$	**72.6**$_{\pm 0.6}$	**73.3**$_{\pm 3.3}$	**46.3**$_{\pm 0.7}$	**75.9**$_{\pm 1.4}$	**57.1**$_{\pm 1.0}$

Table 2: **ProQ preserves proposal performance while converting learned local value into trajectory-level gains.** Shown for DeepSeek-R1, 1.5B. Rescue/Keep accuracy evaluates local value ordering for beneficial and harmful alternatives, while trajectory Rescue/Degradation measures the corresponding incorrect→correct and correct→incorrect outcome transitions.

	End-to-End Accuracy			Local Value Diagnostics		Trajectory Transitions	
Benchmark	**Base Acc.**	**Adapted Acc.**	**ProQ Acc.**	**Rescue Acc.**	**Keep Acc.**	**Rescue (%)**	**Degradation (%)**
MATH	65.4	65.8	77.2	57.9	41.9	14.5	3.1
AIME25	15.6	14.4	23.3	45.4	57.1	10	1.1
MMLU-Pro	23.0	23.6	28.8	72.5	53.3	9.5	4.3
GPQA-D	10.1	12.6	19.8	57.7	42.1	12.9	5.7

learning candidate value provides a stronger mechanism for improving difficult reasoning decisions than directly reshaping the full next-token policy.

Out-of-domain transfer. ProQ is trained exclusively on mathematical reasoning yet achieves the best performance in all nine coding and broad-knowledge evaluations. This transfer holds across DeepSeek-R1, Llama-3.2, and Qwen3, despite large differences in architecture, scale, and Base capability. The result indicates that ProQ learns a transferable local decision criterion rather than a task-specific preference over mathematical reasoning tokens.

5.3 RQ2: Selection efficacy

RQ2.1: Is there actionable headroom within the model's own candidate set? Before evaluating learned value-based selection, we first ask how much improvement is available within the Base candidate set. Across models and benchmarks, Top-8 oracle selection yields 16.7–34.9 points of headroom, averaging 24.3 points. Rank-depth analyses further show that much of this opportunity already lies among shallow Base candidates. Complete statistics are reported in Table A3, with rank-wise decompositions in Figures A15 and A16.

RQ2.2: Does ProQ learn useful local value signals while preserving proposal behavior? Table 2 separates proposal adaptation, local value quality, and their trajectory-level effect. *Adapted Acc.* denotes end-to-end accuracy when the ProQ-trained model decodes only from its standard-token policy, with value-based selection disabled. Across benchmarks, Adapted remains close to Base, while enabling ProQ consistently improves end-to-end accuracy. Locally, *Rescue Acc.* measures whether ProQ ranks a successful alternative above the current proposal, whereas *Keep Acc.* measures whether it preserves a successful Base proposal. At the trajectory level, rescues consistently exceed degradations, showing that these local interventions produce a positive net shift in reasoning outcomes. The same proposal–value separation persists on Qwen3-8B; see Table A4.

RQ2.3: Do selective local interventions compose into consistent global gains? We next examine *how* learned local value influences generation. We vary the intervention trigger using standard-token entropy or KL disagreement with a stronger 32B reference model, and sweep percentile thresholds from 50 to 99. Across both trigger families and benchmarks, Figure A17 reveals a consistent non-monotonic regime: broad intervention degrades reasoning, selective triggering yields the strongest performance. This selective regime also reduces generation-length hit caps, indicating that local value is most effective as a sparse control signal at consequential decisions rather than as a replacement for the standard proposal policy throughout generation. Hit-cap statistics and end-to-end results are reported in Table A5; full trigger sweeps are provided in Appendix D.2.

5.4 RQ3: Design and efficiency

RQ3.1: Which objective design supports effective selection? We ablate how ProQ learning should be structured. All-pairs success–failure ranking substantially improves over Base-anchored supervision, while a Top-1 margin further improves the controlled result by sharpening discrimination between the best successful and unsuccessful candidates. Proposal KL sharply limits standard-token drift without benefiting from increasingly strong regularization, motivating the final combination of pairwise ranking, Top-1 alignment, and proposal preservation. Details are reported in Appendix D.3.

RQ3.2: How is proposal preservation balanced against value learning? Because proposal and evaluation share the same adapted LLM, value learning must improve candidate discrimination while keeping standard-token proposal behavior close to Base. Figure A18 shows that across selector-triggered and non-triggered states, the adapted standard-token policy remains strongly aligned with Base in candidate membership and local ordering. Even at triggered states, Base–ProQ Top-8 overlap remains 98.8%, despite larger distributional drift. This contrast shows that ProQ primarily adapts probability allocation while preserving the underlying proposal structure, with detailed distributional analyses provided in Appendix D.4.

RQ3.3: What are ProQ's training and inference costs? ProQ concentrates its training cost in a one-time, reusable counterfactual-data construction stage. For the 1.5B configuration, we collect 61,408 reward-labeled candidate-conditioned continuations from 3,838 Base trajectories and reuse this asset throughout optimization. Training then requires 300 updates over 14,400 state exposures while adapting only 1.04% of the Base model. At inference, ProQ operates as ordinary decoding except at sparse trigger positions, where the current Top-8 proposals are evaluated through parallel one-token branch forwards of the same LLM before selecting the next token. Detailed training-cost and end-to-end inference measurements are reported in Appendix D.5 and Table A6, respectively.

6 Conclusion

We studied what reasoning post-training learns when large reasoning gains arise from a small number of consequential token-level changes. Across post-training stages, the preferred decisions remain largely within the pretrained model's high-ranked candidate set, while their relative preferences are progressively reorganized. This inheritance persists even after the post-trained policy diverges onto newly successful trajectories, separating local candidate availability from executable global competence. Motivated by this view, ProQ shifts post-training from direct next-token policy adaptation toward value-guided selection among the model's own proposals. Its gains on mathematical reasoning and transfer to coding and knowledge tasks establish candidate-level scoring as a transferable post-training target. More broadly, capability acquisition need not require new local choices, but can emerge from reorganizing and composing those already proposed by the model.

AI USE STATEMENT

We used generative AI tools as interactive research and writing assistants. These tools were used to provide feedback on methodology and experimental design, assist with code implementation, and support the interpretation of experimental results. We also used them for literature search and summarization, suggesting experimental parameters, drafting and editing code, and polishing the manuscript. The scientific hypotheses and conceptual framework of the work were developed by the authors. All AI-assisted outputs were reviewed and verified by the authors, who take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce both the empirical analysis and PROQ experiments. Appendix B documents the exact model checkpoints and post-training lineages, training and evaluation datasets, benchmark protocols, PROQ implementation, baseline configurations and trajectory-matched comparison protocol. Appendix C gives the complete construction and metric definitions for the candidate-inheritance, preference-rewriting, rank-evolution, and Gain-trajectory analyses, together with extended results across models and tasks. The supplementary materials include the implementation and experiment configurations used to generate the reported results, including training and evaluation.

REFERENCES

Ömer Faruk Akgül, Rajgopal Kannan, Willie Neiswanger, and Viktor Prasanna. Rethinking rl for llm reasoning: It's sparse policy selection, not capability learning. *arXiv preprint arXiv:2605.06241*, 2026.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2025.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 36, pp. 30377–30385, 2026.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of machine learning research*, 25(70):1–53, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 108059–108130, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Yuhang He, Haodong Wu, Siyi Liu, Hongyu Ge, Hange Zhou, Keyi Wu, Zhuo Zheng, Qihong Lin, Zixin Zhong, and Yongqi Zhang. Rethinking token-level credit assignment in rlvr: A polarity-entropy analysis. *arXiv e-prints*, pp. arXiv–2604, 2026.

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Constantinos Karouzos, Xingwei Tan, and Nikolaos Aletras. Where does output diversity collapse in post-training? *arXiv preprint arXiv:2604.16027*, 2026.

Jaeyong Ko, Pilsung Kang, and Yukyung Lee. Cliff tokens: Identifying single-token failure triggers in llm mathematical reasoning. *arXiv preprint arXiv:2606.25524*, 2026.

Sander Land and Max Bartolo. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11631–11646, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *Advances in Neural Information Processing Systems*, 38:17998–18031, 2025.

Rohin Manvi, Joey Hong, Tim Seyde, Maxime Labonne, Mathias Lechner, and Sergey Levine. Zero-overhead introspection for adaptive test-time compute. In *The Fourteenth International Conference on Learning Representations*, 2026.

Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Guoyin Wang, Bolin Ding, Jingren Zhou, et al. Sparse but critical: A token-level analysis of distributional shifts in rlvr fine-tuning of llms. In *International Conference on Learning Representations*, volume 2026, pp. 124171–124210, 2026.

Phuc Minh Nguyen, Chinh D La, Duy MH Nguyen, Nitesh V Chawla, Binh T Nguyen, and Khoa D Doan. The reasoning boundary paradox: How reinforcement learning constrains language models. *arXiv preprint arXiv:2510.02230*, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Jing-Cheng Pang, Liang Lu, Xian Tang, Kun Jiang, Sijie Wu, Kai Zhang, and Xubin Li. Reinforcement learning with promising tokens for large language models. *arXiv preprint arXiv:2602.03195*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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

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

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2025.

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

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. In *International Conference on Learning Representations*, pp. 49450–49483, 2026.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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

Wenxuan Ye, Yangyang Zhang, Xueli An, Georg Carle, and Yunpu Ma. Select to think: Unlocking SLM potential with local sufficiency. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=9AIG4VatqI.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2025.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

This appendix provides detailed positioning, experimental settings, and extended evidence for the analyses and ProQ results. We first position our work relative to prior studies and describe the complete evaluation setup. We then present extended analyses of candidate inheritance and local-to-global capability formation, followed by additional ProQ results, objective ablations, proposal-preservation diagnostics, and computational-cost analyses.

A Positioning Relative to Related Work

Several recent studies show that reasoning post-training can depend on a small number of consequential local changes. Token-level analyses identify sparse, high-uncertainty decisions as particularly important (Meng et al., 2026; Akgül et al., 2026), while complementary methods exploit candidate locality by restricting optimization to promising actions (Pang et al., 2026) or selecting among candidates already proposed by another model (Ye et al., 2026). Together, these works establish that substantial reasoning gains need not require unrestricted modification of the full vocabulary at every step.

These observations, however, leave a more fundamental question unresolved: *where does new reasoning capability reside when many of the required local decisions are already present before post-training?* Our work directly studies this capability-level distinction. By tracing complete post-training lineages and following the post-trained policy onto newly successful prefixes, we show that globally different reasoning behavior can emerge while local candidate supply remains largely inherited. This identifies state-dependent reorganization and autoregressive composition, rather than local candidate novelty alone, as a mechanism through which new reasoning competence can be formed. This view motivates ProQ. Instead of treating the full next-token policy as the sole object of adaptation, ProQ preserves standard-token proposal behavior and learns reward-grounded candidate-conditioned scores over the model's own proposals within the same LLM. These scores support value-guided selection over a largely inherited local candidate set. Thus, ProQ operationalizes the broader hypothesis that stronger global reasoning can be learned through value-based control over a largely inherited local decision basis.

B Evaluation Settings

B.1 Model families and post-training lineages

We analyze candidate inheritance and preference rewriting across multiple model families, scales, and post-training regimes. Our study includes DeepSeekMath (Shao et al., 2024), Qwen2.5-Math/QwQ (Yang et al., 2024), OLMo-2 (OLMo et al., 2024), and Qwen3 (Yang et al., 2025). Where explicit intermediate checkpoints are available, we trace complete stage-wise lineages spanning supervised fine-tuning, preference optimization, and reinforcement learning. Table A1 lists the exact Base and post-trained checkpoints used in each analysis.

B.2 ProQ training and evaluation

We evaluate ProQ on three backbones spanning Qwen- and Llama-based architectures and multiple scales: DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025), Llama-3.2-3B (Grattafiori et al., 2024), and Qwen3-8B (Yang et al., 2025). All trainable methods use the same mathematical problem pool and outcome-based correctness reward.

We construct the training pool from a 4,096-problem sample of OpenThoughts-Math. After filtering two over-length prompts and reserving 128 examples each for validation and test, 3,838 problems remain for training.

We evaluate mathematical reasoning on MATH-500 (Lightman et al., 2024), OlympiadBench (He et al., 2024), and AIME25 (Zhang & Math-AI, 2025). MATH-500 contains 500 competition-level problems spanning diverse mathematical topics. For OlympiadBench, we use the text-only scorable subset compatible with our evaluation pipeline. AIME25 contains all 30 problems from the 2025 AIME I and II examinations.

Algorithm 1 PROQ inference. Green denotes standard-token proposal operations and blue denotes reserved-token value operations.

Require: Input x, ProQ-adapted LLM f_ϕ, the standard decoding rule $\mathrm{Decode}(\cdot)$, trigger threshold τ, candidate size K

$s_1 \leftarrow x$
for t **do**
  Run f_ϕ on s_t and obtain standard-token logits $z_\phi^{\mathrm{std}}(s_t)$
  $\pi_\phi^{\mathrm{std}}(\cdot \mid s_t) \leftarrow \mathrm{Softmax}(z_\phi^{\mathrm{std}}(s_t))$
  $H_t \leftarrow H\left(\pi_\phi^{\mathrm{std}}(\cdot \mid s_t)\right)$
  if $H_t < \tau$ **then**
    $c_t \leftarrow \mathrm{Decode}\left(\pi_\phi^{\mathrm{std}}(\cdot \mid s_t)\right)$,
  else
    $\mathcal{C}_K^\phi(s_t) \leftarrow \mathrm{TopK}\left(\pi_\phi^{\mathrm{std}}(\cdot \mid s_t)\right)$
    for all $c_j \in \mathcal{C}_K^\phi(s_t)$ **in parallel do**
      Run one-token branch $f_\phi([s_t; c_j])$
      Read reserved-token logits $z_\phi^{\mathrm{res}}(s_t, c_j)$
      $Q_\phi(s_t, c_j) \leftarrow \mathrm{ValueReadout}\left(z_\phi^{\mathrm{res}}(s_t, c_j)\right)$
    end for
    $c_t \leftarrow \arg\max_{c_j} Q_\phi(s_t, c_j)$
  end if
  $s_{t+1} \leftarrow [s_t; c_t]$
  if c_t is EOS **then**
    break
  end if
end for
return generated trajectory

For code generation, we use the LiveCodeBench (Jain et al., 2025) fine-grained v6 slice containing 175 newly introduced problems, evaluated by execution-based pass@1. For knowledge-intensive reasoning, we use a fixed 1,400-example subset of MMLU-Pro (Wang et al., 2024) and the full 198-question GPQA Diamond split (Rein et al., 2023).

All methods use the same zero-shot task instructions and answer-extraction procedure within each benchmark. Decoding configurations are held fixed across methods within each model family and are detailed in Appendix B.4.

B.3 PROQ IMPLEMENTATION

DeepSeek-R1-Distill-Qwen-1.5B configuration. ProQ uses $K = 8$ candidate proposals and $B = 16$ reserved value tokens. We apply LoRA to the attention (q, k, v, o) and MLP $(\mathrm{gate}, \mathrm{up}, \mathrm{down})$ projections with rank 16, scaling 32, and dropout 0.05. The shared LM head is retained, with direct output-row updates restricted to the reserved-token coordinates.

The final selection objective is

$$\mathcal{L}_{\mathrm{select}} = \mathcal{L}_{\mathrm{pair}} + \lambda_{\mathrm{top1}} \mathcal{L}_{\mathrm{top1}},$$

with $\lambda_{\mathrm{top1}} = 0.5$. We use pairwise temperature $T_{\mathrm{pair}} = 0.2$, Top-1 margin $m = 0.02$, and margin scale $T_{\mathrm{top1}} = 0.2$. Proposal preservation uses

$$\mathcal{L}_{\mathrm{propose}} = D_{\mathrm{KL}}\left(\pi_B^{\mathrm{std}} \,\|\, \pi_\phi^{\mathrm{std}}\right),$$

with $\lambda_{\mathrm{prop}} = 30$; reserved-token positions are excluded from this KL.

We use $M = 1$ in all experiments. Optimization uses AdamW with peak learning rate 2×10^{-5}, global batch size 48, gradient clipping at 5, and a cosine schedule. The training run contains 300 optimizer steps.

Table A1: **Model families and checkpoints used in our analysis.**

Family	Scale	Base checkpoint	Post-trained checkpoint(s)
Qwen2.5-Math	1.5B	`Qwen2.5-Math-1.5B`	`Qwen2.5-Math-1.5B-Instruct`
DeepSeekMath	7B	`deepseek-math-7b-base`	`deepseek-math-7b-instruct` `deepseek-math-7b-rl`
Qwen2.5-Math	7B	`Qwen2.5-Math-7B`	`Qwen2.5-Math-7B-Instruct`

OLMo-2	7B	`OLMo-2-1124-7B`	`OLMo-2-1124-7B-SFT`
			`OLMo-2-1124-7B-DPO`
			`OLMo-2-1124-7B-Instruct`

OLMo-3-Think	7B	`Olmo-3-1025-7B`	`Olmo-3-7B-Think-SFT`
			`Olmo-3-7B-Think-DPO`
			`Olmo-3-7B-Think`

Qwen3	14B	`Qwen3-14B-Base`	`Qwen3-14B`

Qwen2.5 / QwQ	32B	`Qwen2.5-32B`	`Qwen2.5-32B-Instruct`
			`QwQ-32B-Preview`
			`QwQ-32B`

Qwen3	30B-A3B	`Qwen3-30B-A3B-Base`	`Qwen3-30B-A3B`

At inference, ProQ uses the 97.5th percentile of a label-free standard-token entropy calibration distribution as its intervention threshold. When triggered, it scores the current Top-8 candidates and selects $\arg\max_c Q_\phi(s_t, c)$. No additional value-confidence gate or cooldown is used.

Llama-3.2-3B configuration. We use $K = 8$ candidate proposals and rank-16 LoRA with $\alpha = 32$ and dropout 0.05. We jointly optimize the proposal-side LoRA and reserved-token readout at a learning rate of 5×10^{-6}, updating 24.31M LoRA parameters and 49.2K reserved-token output coordinates (24.36M effective updated parameters in total). Training uses cosine learning-rate decay, 3% warmup, and gradient clipping at 5. The selection objective combines all-pairs ranking, a 0.5-weighted Top-1 objective, and standard-vocabulary preservation with $\lambda_{\text{prop}} = 30$; both ranking objectives use temperature 0.2, with a Top-1 margin of 0.02.

Qwen3-8B configuration. We use $K = 8$ candidate proposals and rank-16 LoRA with scaling 32 and dropout 0.05. We optimize the proposal-side LoRA and reserved-token readout with separate learning rates of 1×10^{-6} and 5×10^{-6}, respectively, updating 43.65M LoRA parameters and 65.5K reserved-token parameters (43.71M in total). Training uses cosine learning-rate decay, 3% warmup, and gradient clipping at 5. The selection objective combines all-pairs ranking, a 0.5-weighted Top-1 objective, and standard-vocabulary preservation with $\lambda_{\text{prop}} = 30$; both ranking objectives use temperature 0.2, with a Top-1 margin of 0.02.

B.4 Evaluation decoding protocol

DeepSeek-R1-Distill-Qwen-1.5B. We use the model's native sampling configuration with temperature 0.6 and nucleus sampling with $p = 0.95$, together with its native reasoning template and `<think>` prefix. At non-triggered positions, generation follows the same sampling policy used for Base and all baselines. When the ProQ trigger fires, the current Top-8 candidates under the adapted standard-token policy are scored using the reserved-token readout, and the highest-scoring candidate is selected. We generate one complete response per problem with a maximum response length of $8{,}192$ tokens. Generation terminates upon the model's EOS/turn-end token or when the maximum length is reached.

Llama-3.2-3B. We use the model's native sampling configuration with temperature 0.6 and nucleus sampling with $p = 0.9$, together with its native instruction template. As for DeepSeek, all methods share the same decoding configuration within the model family, while ProQ replaces the sampled next-token decision with value-guided selection only at triggered positions.

Qwen3-8B. We evaluate Qwen3-8B in thinking mode using its native chat template with `add_generation_prompt=True`, following the recommended sampling configuration of temperature 0.6, top-$p = 0.95$, and top-$k = 20$. We generate one complete response per problem with a maximum response length of 32,768 tokens. Generation stops at EOS (`ignore_eos=False`) or when the maximum generation length is reached.

Because decoding is stochastic, all methods are evaluated with 3 independent decoding seeds, each covering the full benchmark. We average accuracy across the three runs.

B.5 Baseline implementation

We implement GRPO and DAPO with the same LoRA configuration used by ProQ: rank 16, scaling 32, and dropout 0.05 over all attention and MLP projection layers. Both methods train on the same math prompts. Each update samples eight responses per prompt from a global batch of 48 prompts, using temperature 1.0, top-p 1.0, and a maximum response length of 8,192 tokens. Optimization uses AdamW with a constant learning rate of 10^{-6}, betas $(0.9, 0.999)$, epsilon 10^{-8}, zero weight decay, and gradient clipping at 1.0.

GRPO uses symmetric ratio clipping $[0.8, 1.2]$ with group-normalized sequence-level advantages. DAPO uses asymmetric clipping $[0.8, 1.28]$, token-level loss aggregation, and a linear overlength penalty over the final 1,024 response tokens with maximum penalty 1.0. We retain DAPO's dynamic sampling procedure, under which prompt groups without reward variation do not contribute an update and additional groups may be generated to obtain informative training signal. All generated trajectories, including those from discarded groups, count toward the matched generation budget described below. RLPT uses the same GRPO optimization backbone and rollout budget while restricting per-token sampling and likelihood normalization to the stored behavior-policy Top-8 support.

C Extended Evidence for Candidate Inheritance and Capability Formation

This section extends the three empirical steps underlying our local-to-global account of capability formation. We first characterize the graded inheritance of Base-ranked candidates, then trace how post-training reorganizes preferences and progressively promotes final decisions, and finally test whether this inherited local structure persists along newly successful Post-generated trajectories.

C.1 Candidate inheritance across tasks, models, and candidate depths

Fixed-prefix construction. We analyze candidate inheritance through a fixed-prefix comparison between each Base model and its post-trained checkpoint. For every example, the Base model first generates a greedy trajectory. At generation step t, both models are evaluated on the identical state $s_t = (x, y^{\mathrm{B}}_{<t})$, where $y^{\mathrm{B}}_{<t}$ is the prefix generated by the Base model.

We focus on positions where post-training changes the Base Top-1 decision,

$$\mathcal{D} = \{t : a^{\star}_{\mathrm{P}}(s_t) \neq a^{\star}_{\mathrm{B}}(s_t)\}\,, \tag{8}$$

where

$$a^{\star}_{\mathrm{B}}(s_t) = \arg\max_{v} \pi_{\mathrm{B}}(v \mid s_t), \qquad a^{\star}_{\mathrm{P}}(s_t) = \arg\max_{v} \pi_{\mathrm{P}}(v \mid s_t). \tag{9}$$

For each disagreement position, we record the rank of the post-trained choice under the Base policy,

$$r_t = \mathrm{rank}_{\mathrm{B}}\left(a^{\star}_{\mathrm{P}}(s_t) \mid s_t\right). \tag{10}$$

Thus, r_t measures how deep within the Base ranking post-training reaches when it changes the locally preferred decision.

Bullseye visualization. Each point in the bullseye represents one disagreement position. Points are assigned to mutually exclusive Base-rank bands, $r_t = 2$, 3–4, 5–8, 9–16, > 16, which form the concentric regions of the plot. Points nearer the center therefore correspond to post-trained

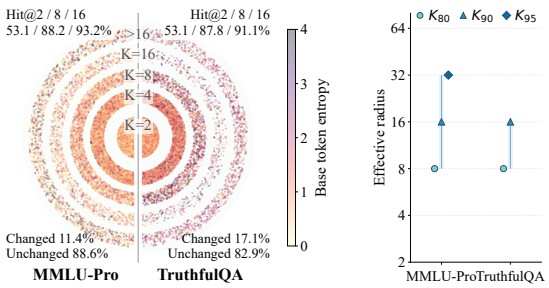


Figure A1: *Candidate inheritance is graded across tasks.* For the same Qwen3-14B lineage, post-trained Top-1 choices at Base–Post disagreement states remain concentrated in shallow Base ranks on both MMLU-Pro and TruthfulQA, with TruthfulQA exhibiting a broader rank tail. The effective candidate radius therefore varies across tasks while preserving the same inheritance pattern.

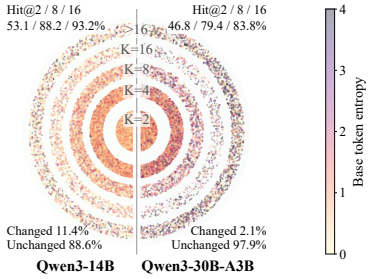


Figure A2: *Candidate depth varies across model lineages.* On MMLU-Pro, Qwen3-30B-A3B changes substantially fewer Base decisions than Qwen3-14B, yet a larger fraction of its changed Top-1 choices lies beyond Base Top-16. Thus, the prevalence of candidate inheritance cannot be inferred from how frequently a checkpoint changes the Base policy.

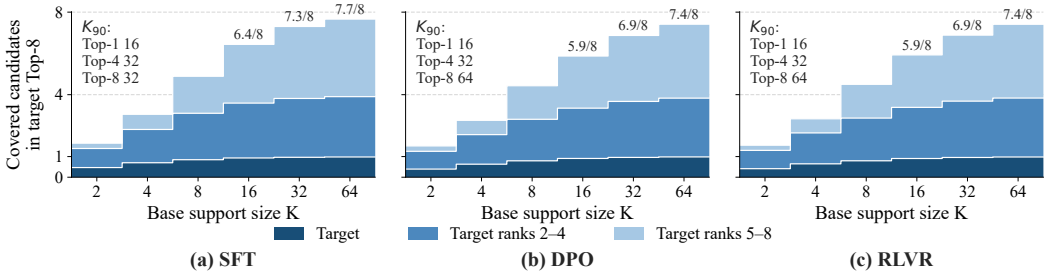


Figure A3: **Candidate inheritance extends beyond the post-trained winner.** For OLMo-2 on MMLU-Pro, expanding Base support progressively recovers the post-trained Top-8 set across SFT, DPO, and RLVR. While 90% of changed winners are recovered by Base Top-16, broader Top-4/Top-8 recovery requires Top-32–64, showing that inheritance persists beyond the selected token but with a graded radius.

choices that were already highly ranked by the Base model, whereas outer rings indicate progressively deeper shifts. Angular position and within-ring jitter are used only to avoid overplotting and carry no quantitative meaning. Point color denotes Base-policy uncertainty at the corresponding state.

We summarize cumulative winner inheritance by

$$\mathrm{Hit@}K = \Pr\left[r_t \le K \mid t \in \mathcal{D}\right], \tag{11}$$

and define the corresponding effective radius as

$$K_\alpha = \min\left\{K \in \{2,4,8,16,32,64\} : \mathrm{Hit@}K \ge \alpha\right\}, \qquad \alpha \in \{0.8, 0.9, 0.95\}. \tag{12}$$

Hence, Hit@K measures what fraction of changed decisions were already available within Base Top-K, while K_{80}, K_{90}, and K_{95} summarize the support depth required to recover increasingly large fractions of those decisions.

The *Changed*/*Unchanged* percentages are computed over all aligned positions rather than only $\mathcal{D}$. They therefore measure how frequently post-training changes the Base Top-1 decision, which is distinct from how deep those changed decisions lie in the Base ranking.

Variation across tasks and model lineages. Candidate inheritance is strong but not governed by a universal rank cutoff. Holding the Qwen3-14B lineage fixed, Figure A1 shows that MMLU-Pro and TruthfulQA exhibit similar shallow-rank concentration, while TruthfulQA has a somewhat heavier tail and therefore a larger effective radius. Conversely, holding MMLU-Pro fixed, Figure A2 shows

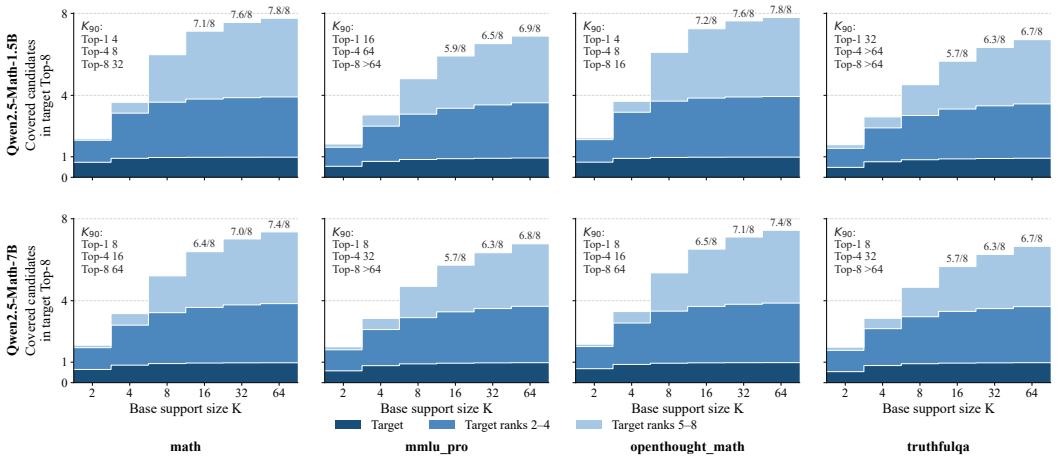


Figure A4: **Candidate inheritance persists across task domains and model scales with a graded effective radius.** Rows compare Qwen2.5-Math-1.5B and 7B, while columns vary the evaluation task. Across both scales and domains, expanding Base support consistently recovers the post-trained candidate set, with the required support depth varying by setting.

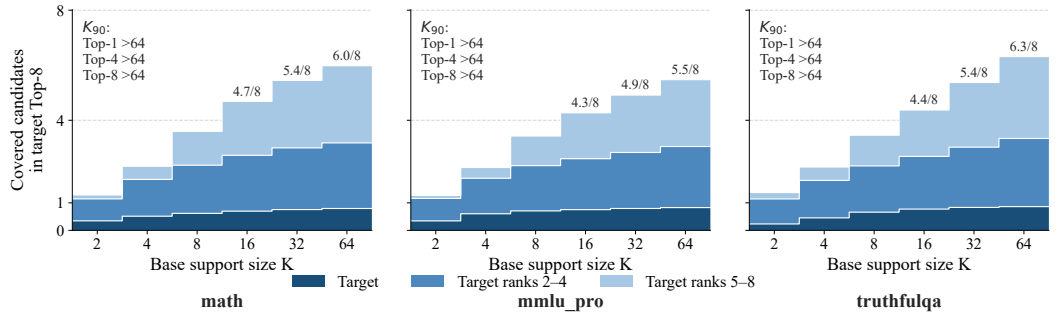


Figure A5: **Candidate inheritance persists in a substantially deeper regime.** For Qwen3-30B-A3B, Base Top-64 continues to recover a substantial fraction of the post-trained Top-8 set across Math, MMLU-Pro, and TruthfulQA, revealing a deeper but still structured inheritance regime.

that Qwen3-30B-A3B changes far fewer Base Top-1 decisions than Qwen3-14B but places a larger fraction of those changed decisions beyond Base Top-16.

Inheritance beyond the winning candidate. To characterize how inheritance extends beyond the selected Top-1 decision, we next examine the broader post-trained Top-L candidate set. For each disagreement state s_t, we measure how much of this set is recovered from progressively larger Base supports.

$$\mathrm{Recall@}K(L) = \frac{|\mathrm{TopL_P}(s_t) \cap \mathrm{TopK_B}(s_t)|}{L}, \tag{13}$$

and report its average over $t \in \mathcal{D}$. For $L = 1$, this reduces to $\mathrm{Hit@}K$. We analogously define

$$K_{90}^{(L)} = \min\{K \in \{2, 4, 8, 16, 32, 64\} : \mathrm{Recall@}K(L) \geq 0.9\}. \tag{14}$$

This measures how much Base support is required to recover not only the post-trained winner, but progressively broader portions of its local candidate set.

Candidate-set accumulation across stages, scales, and domains. The broader candidate set exhibits the same qualitative inheritance pattern, but with a larger and task-dependent effective radius. For OLMo-2 on MMLU-Pro, Base Top-16 already recovers 90% of changed winners across SFT, DPO, and RLVR, whereas recovering 90% of the post-trained Top-8 requires Base Top-32 after SFT and Top-64 after DPO and RLVR (Figure A3). Across Qwen2.5-Math scales, mathematical tasks

Table A2: **Candidate inheritance is broad across models and tasks, but its effective radius is graded.** Hit@K measures the fraction of post-trained Top-1 replacements contained in Base Top-K, while Recall@$K(L)$ measures the fraction of the post-trained Top-L candidate set recovered by Base Top-K. $K_{90}^{(L)}$ denotes the smallest $K \in \{2, 4, 8, 16, 32, 64\}$ reaching 90% recovery, with >64 indicating that the threshold is not reached within the evaluated range. Across all 31 model–dataset–checkpoint settings, Base Top-32 recovers 92.2% of the post-trained Top-4 candidates on average, while Base Top-64 recovers 91.8% of the Top-8. The substantial variation in $K_{90}^{(L)}$ shows that candidate inheritance is a structural tendency rather than a universal fixed-support rule.

Model	Checkpoint	Dataset	Winner inheritance			Candidate-set inheritance			
			Hit@8	Hit@16	$K_{90}^{(1)}$	Recall@32(4)	Recall@64(8)	$K_{90}^{(4)}$	$K_{90}^{(8)}$
Qwen2.5-Math-1.5B	Instruct	OpenThoughts-Math	96.8	97.9	4	98.0	97.5	8	16
		MMLU-Pro	86.8	90.7	16	88.3	86.2	64	>64
		TruthfulQA	85.5	89.7	32	87.2	84.0	>64	>64
Qwen2.5-Math-7B	Instruct	MATH	93.1	95.7	8	94.9	92.0	16	64
		OpenThoughts-Math	94.9	96.8	8	95.7	93.0	16	64
		MMLU-Pro	91.7	95.0	8	90.5	84.7	32	>64
		TruthfulQA	92.4	95.4	8	90.4	83.2	32	>64
DeepSeekMath-7B	Instruct	OpenThoughts-Math	93.2	96.7	8	97.9	98.5	16	16
	RL		91.7	96.3	8	97.2	97.7	16	32
	Instruct	MMLU-Pro	92.9	97.0	8	96.5	96.9	16	32
	RL		92.7	97.2	8	96.3	96.6	16	32
	Instruct	TruthfulQA	87.2	93.3	16	92.8	93.5	32	64
	RL		88.2	95.5	16	91.9	91.8	32	64
OLMo-2-7B	SFT	MATH	93.5	97.0	8	97.9	98.1	16	32
	DPO		89.0	95.4	16	96.5	96.8	16	32
	RLVR		88.4	95.0	16	96.2	96.3	16	32
	SFT	OpenThoughts-Math	94.8	97.8	8	98.6	98.9	8	16
	DPO		89.4	95.5	16	97.1	97.5	16	32
	RLVR		88.8	95.1	16	96.8	97.2	16	32
	SFT	MMLU-Pro	84.8	93.2	16	95.4	95.8	32	32
	DPO		79.4	90.9	16	91.8	92.7	32	64
	RLVR		79.5	90.7	16	92.2	92.7	32	64
	SFT	TruthfulQA	83.0	89.5	32	89.0	90.9	64	64
	DPO		83.8	90.9	16	91.5	92.6	32	64
	RLVR		83.6	90.9	16	91.2	92.3	32	64
Qwen3-14B	Final	MATH	83.2	90.1	16	91.7	91.4	32	64
		OpenThoughts-Math	84.5	91.6	16	93.4	93.3	32	64
		MMLU-Pro	88.2	93.2	16	93.4	92.9	32	64
		TruthfulQA	87.8	91.1	16	83.7	86.5	>64	>64
Qwen3-30B-A3B	Final	MATH	61.8	69.8	>64	74.9	74.8	>64	>64
		MMLU-Pro	79.4	83.8	>64	70.8	68.4	>64	>64
Macro average (31 settings)			**87.4**	**92.9**	–	**92.2**	**91.8**	–	–

generally exhibit shallower accumulation than MMLU-Pro and TruthfulQA, while increasing model scale does not collapse these differences into a universal radius (Figure A4). For Qwen3-30B-A3B, Base Top-64 continues to recover a substantial fraction of the post-trained Top-8 set across Math, MMLU-Pro, and TruthfulQA, revealing a deeper but still structured inheritance regime (Figure A5).

The aggregate results in Table A2 confirm this graded structure across all evaluated settings. Across 31 model–dataset–checkpoint combinations, Base Top-32 recovers 92.2% of the post-trained Top-4 candidates on average, while Base Top-64 recovers 91.8% of the Top-8. At the same time, the effective $K_{90}^{(L)}$ varies substantially across models and tasks, showing that candidate inheritance is broad but does not obey a universal fixed-support radius.

C.2 Preference rewriting across post-training stages

We provide the formal definitions underlying Section 3.2.

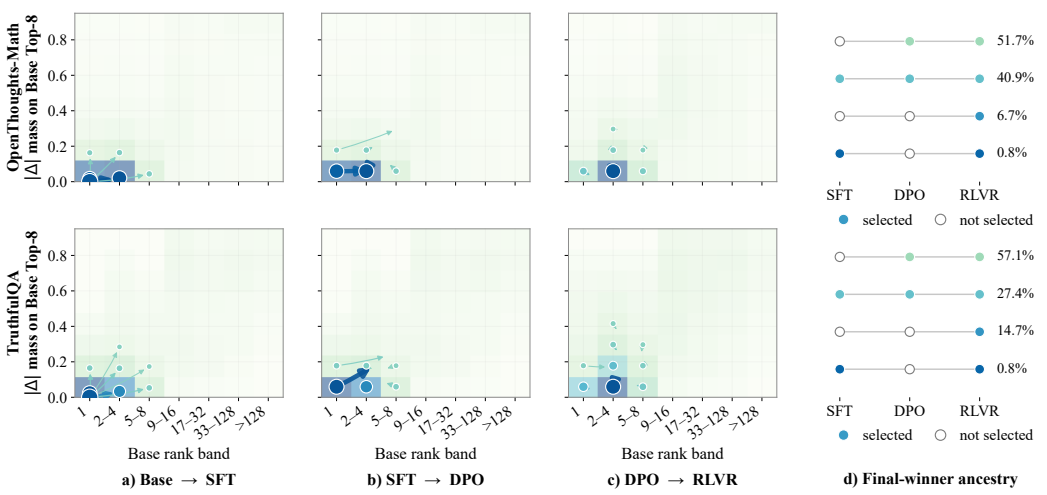


Figure A6: **Candidate inheritance persists despite changes in support-level probability mass.** For the OLMo-2-7B Base→SFT→DPO→RLVR lineage, we measure the Base-relative change in total probability assigned to the fixed Base Top-8 support on OpenThoughts-Math and TruthfulQA. Selected decisions remain concentrated in shallow Base-rank bands even when the mass assigned to this inherited support changes, motivating the conditional within-support analysis that follows.

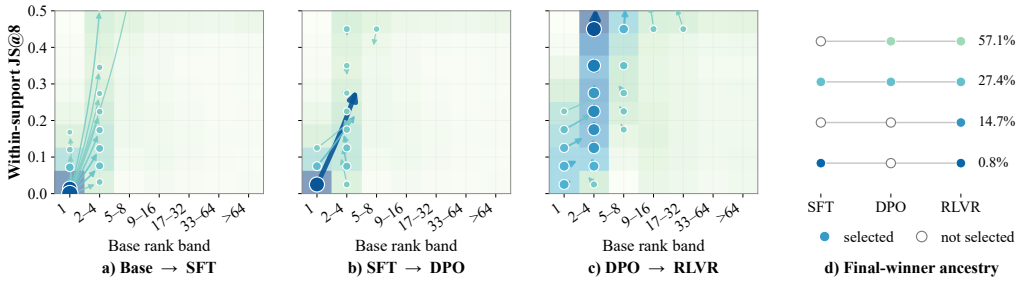


Figure A7: **Preference rewriting persists beyond mathematical reasoning.** For the OLMo-2-7B Base→SFT→DPO→RLVR lineage on TruthfulQA, selected decisions remain concentrated in relatively shallow Base-rank regions while within-support JS@8 increases across post-training stages. The final-winner ancestry further shows progressive decision formation: 27.4% of eventual RLVR winners are already selected from SFT onward, 57.1% first become Top-1 at DPO, and 14.7% only at RLVR.

Probability mass on the inherited support. Before conditioning on the inherited candidate set, we first examine how post-training changes the total probability assigned to this set. For each aligned state s, we fix the Base Top-8 candidates $\mathcal{C}_8^{\mathrm{B}}(s)$ and define

$$M_t^{(8)}(s) = \sum_{c \in \mathcal{C}_8^{\mathrm{B}}(s)} \pi_t(c \mid s). \tag{15}$$

We then measure the Base-relative support-mass shift

$$\Delta M_t^{(8)}(s) = \left| M_t^{(8)}(s) - M_{\mathrm{B}}^{(8)}(s) \right|. \tag{16}$$

Unlike the conditional analysis below, this quantity uses the original probability mass and therefore retains movements into or out of the Base Top-8 support.

Figure A6 places this Base-relative mass deviation against the Base rank of each stage's preferred token. The horizontal axis therefore tracks how deeply the selected decision lies in the Base ranking, whereas the vertical axis tracks how far the total probability assigned to the inherited Top-8 support has moved from its Base value.

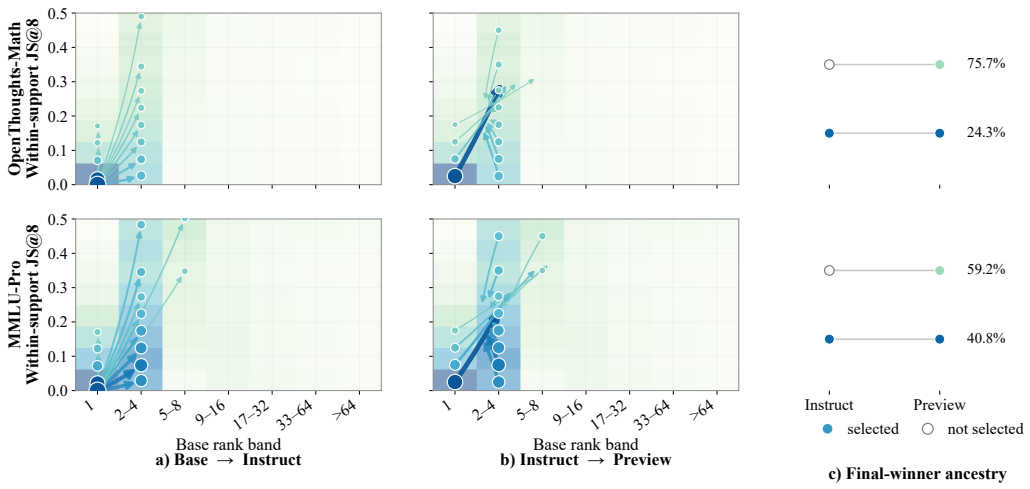


Figure A8: **Preference rewriting generalizes across model lineages and reasoning domains.** For the Qwen2.5-32B Base→Instruct→QwQ-Preview lineage, both OpenThoughts-Math and MMLU-Pro show increasing deviation from the Base preference allocation while selected decisions remain concentrated in relatively shallow Base-rank regions. Final-winner ancestry shows continued promotion at the later stage: only 24.3% of eventual QwQ-Preview winners on OpenThoughts-Math and 40.8% on MMLU-Pro are already Top-1 at the Instruct checkpoint.

Across both OpenThoughts-Math and TruthfulQA, stage-wise winners remain concentrated in shallow Base-rank bands even as the total probability carried by the Base Top-8 support changes across post-training stages. The effect is especially visible on TruthfulQA, where later stages exhibit larger support-mass shifts without a corresponding movement of selected decisions into substantially deeper Base ranks. Thus, candidate inheritance does not rely on the total probability mass of the Base Top-8 remaining numerically fixed.

Preference rewriting conditional on the inherited support. The support-mass analysis above captures whether probability moves into or out of the Base Top-8 set, but it does not isolate how preference is redistributed *among* the inherited candidates themselves. To separate these two effects, we next condition each checkpoint's distribution on the same fixed Base Top-8 support.

For checkpoint t, we renormalize its next-token distribution over this shared support,

$$\widetilde{\pi}_t(v \mid s) = \frac{\pi_t(v \mid s)}{\sum_{u \in \mathcal{C}_8^{\mathrm{B}}(s)} \pi_t(u \mid s)}, \qquad v \in \mathcal{C}_8^{\mathrm{B}}(s), \tag{17}$$

and measure Base-relative within-support preference deviation as

$$J_t^{\mathrm{within}}(s) = \mathrm{JS}\left(\widetilde{\pi}_{\mathrm{B}}(\cdot \mid s), \widetilde{\pi}_t(\cdot \mid s)\right). \tag{18}$$

This normalization deliberately removes changes in the total mass assigned to the Base Top-8 set. Consequently, J_t^{within} isolates changes in the *relative* preference allocation among the same inherited candidates, complementing the support-mass analysis above.

Stage-wise transport visualization. All checkpoints are evaluated on the same Base-generated prefixes. We retain the common set of aligned positions at which at least one post-training stage changes the Base Top-1 decision, so every transition tracks the same decision states throughout the lineage.

For each position and checkpoint, the horizontal coordinate is the Base rank of that checkpoint's preferred token, and the vertical coordinate is $J_t^{\mathrm{within}}(s)$. Background intensity represents the density of aligned positions, while each circle aggregates positions with similar rank and preference deviation. Arrows connect the same positions across consecutive checkpoints and terminate at their average destination. Horizontal movement reflects changes in candidate depth, whereas vertical movement reflects changes in Base-relative preference organization within the fixed inherited support.

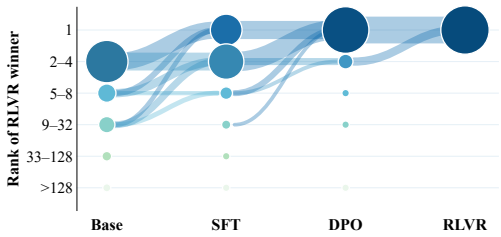


Figure A9: **Final RLVR decisions are progressively promoted from inherited Base candidates.** For each aligned position on OpenThoughts-Math, we fix the eventual RLVR Top-1 decision and trace its rank across Base, SFT, DPO, and RLVR. Most final decisions begin within shallow Base-rank bands and move progressively toward Top-1, showing that later post-training stages primarily strengthen and reorganize pre-existing candidates rather than introducing the final decision abruptly from deep ranks.

Final-winner ancestry. To complement the distributional transport view with a decision-level perspective, we track when the decision ultimately selected by the final checkpoint first becomes Top-1 across the post-training lineage. For every Base–Final disagreement position, we therefore fix the eventual final winner and record whether it is already Top-1 at each intermediate checkpoint. Filled markers indicate stages at which the eventual final winner is selected, while open markers indicate that it has not yet become Top-1. This binary ancestry view identifies when final decisions are acquired across the lineage; the finer-grained rank evolution of these same decisions is analyzed separately in Appendix C.3.

Cross-domain validation. The separation between candidate depth and preference rewriting persists beyond mathematical reasoning. Using the same OLMo-2 Base→SFT→DPO→RLVR lineage on TruthfulQA, Figure A7 shows that disagreement-conditioned Hit@16 remains nearly stable across SFT, DPO, and RLVR (around 90.0%), while J_t^{within} increases from 0.254 to 0.274 and 0.283. Thus, even in a deeper candidate regime, successive post-training stages alter relative preference organization substantially without a comparable expansion of the Base-ranked region supplying the selected decisions. The final-winner ancestry provides a complementary decision-level view: many eventual RLVR winners become Top-1 before the final RLVR stage, indicating progressive decision formation across the lineage. Their rank evolution before selection is examined separately below.

Cross-lineage validation. Figure A8 repeats the preference-rewriting analysis on the Qwen2.5-32B Base→Instruct→QwQ-Preview sequence using OpenThoughts-Math and MMLU-Pro. Across both tasks, later checkpoints remain concentrated in relatively shallow Base-rank regions while moving farther from the Base preference allocation within the fixed Top-8 support.

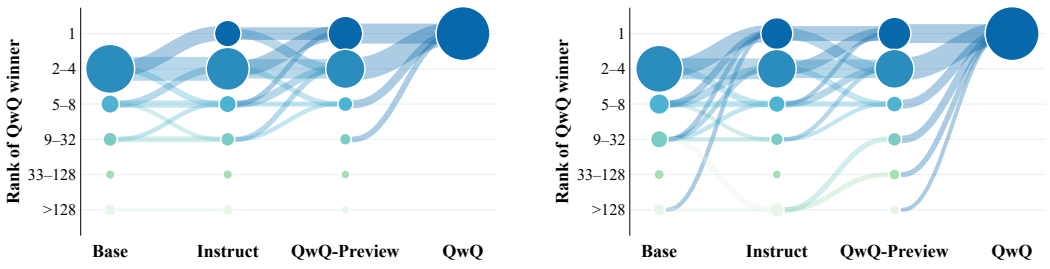
Final-winner ancestry gives the same pattern from a complementary perspective. Only 24.3% of eventual QwQ-Preview winners on OpenThoughts-Math and 40.8% on MMLU-Pro are already Top-1 at the Instruct checkpoint; the remainder become selected only at the later Preview stage. Thus, substantial changes in the final decision can emerge through continued reorganization of candidates already present in the Base ranking.

C.3 Progressive promotion of final decisions

The binary ancestry view identifies when an eventual final winner becomes Top-1. We further refine this picture by tracing the rank of the same candidate before selection, distinguishing progressive promotion from late-stage emergence. Figure A9 visualizes this rank evolution for the OLMo-2 lineage on OpenThoughts-Math.

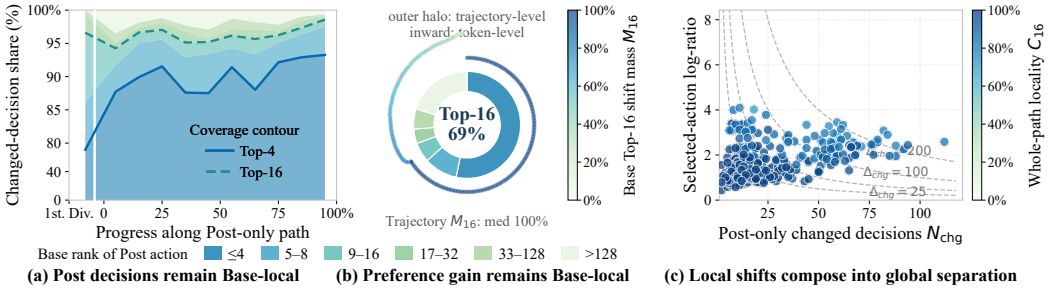
Visualization setup. For every aligned position at which the final checkpoint overturns the Base Top-1 decision, we fix the eventual final winner

$$a_{\mathrm{F}}^{\star}(s) = \arg\max_{v} \pi_{\mathrm{F}}(v \mid s), \tag{19}$$



(a) **OpenThoughts-Math** (b) **MMLU-Pro**

Figure A10: **QwQ decisions are already structured within earlier Qwen2.5 candidate rankings.** For each position, we fix the final QwQ Top-1 decision and trace its rank under the related Base, Instruct, and QwQ-Preview checkpoints on (a) OpenThoughts-Math and (b) MMLU-Pro. Across both tasks, many eventual QwQ decisions are already highly ranked by earlier checkpoints and become progressively concentrated near Top-1, extending the candidate-promotion pattern beyond the OLMo-2 lineage.



(a) **Post decisions remain Base-local** (b) **Preference gain remains Base-local** (c) **Local shifts compose into global separation**

Figure A11: **Candidate inheritance persists under domain shift, with greater trajectory-level heterogeneity.** For Qwen2.5-Math-1.5B on MMLU-Pro, changed Post decisions remain strongly Base-local after divergence, while Base Top-16 captures 69% of aggregate positive shift mass and the median trajectory-level M_{16} remains 100%. The gap between aggregate and trajectory-level locality reflects a subset of trajectories with disproportionately large shifts on deeper-ranked actions.

where F denotes the final checkpoint under analysis. For every checkpoint t, we then compute the rank of this same token under that checkpoint's distribution,

$$r_t^{\mathrm{F}}(s) = \mathrm{rank}_t\left(a_{\mathrm{F}}^{\star}(s) \mid s\right). \tag{20}$$

The analysis therefore follows one fixed eventual decision across the entire checkpoint sequence.

We group the stage-wise rank into six bands: 1, 2–4, 5–8, 9–32, 33–128, and $>$ 128. At each checkpoint, circle area denotes the fraction of aligned positions whose eventual winner occupies the corresponding rank band. Flows connect the same positions across consecutive checkpoints, with width proportional to transition frequency and color indicating rank depth.

A path such as 5–8 $\rightarrow$ 2–4 $\rightarrow$ 1 indicates progressive promotion of an already plausible candidate. By contrast, a transition from a very deep rank directly to 1 would indicate late-stage emergence. The visualization therefore distinguishes gradual promotion from abrupt appearance of the final decision.

Cross-lineage extension. Figure A10 extends the rank-evolution analysis to the related Qwen2.5-32B Base and Instruct checkpoints together with QwQ-Preview and QwQ. Across these related checkpoints, eventual QwQ winners are frequently already located in shallow or intermediate Base-ranked regions and become increasingly concentrated near Top-1. MMLU-Pro exhibits a broader initial rank distribution than OpenThoughts-Math, but the same progressive-promotion pattern remains.

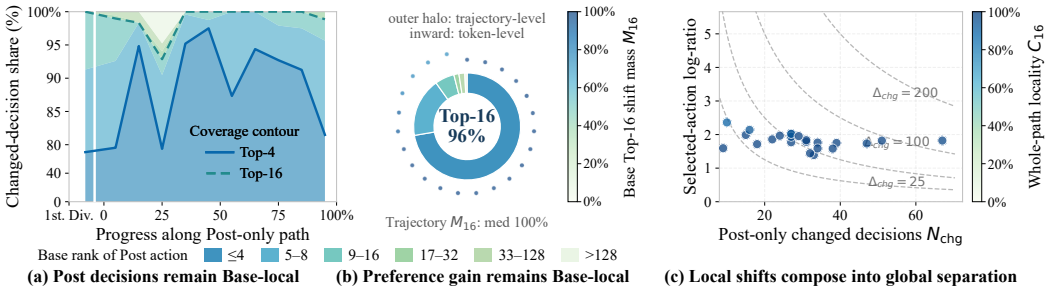


Figure A12: **The local-to-global structure persists on DeepSeekMath-7B and MATH-500.** Changed Post decisions remain predominantly concentrated in shallow Base ranks along newly successful trajectories, while 96% of the positive preference increase occurs on Base Top-16 actions and the trajectory-level median M_{16} reaches 100%. Despite this strong locality, repeated local changes still accumulate into substantial path-level policy differences.

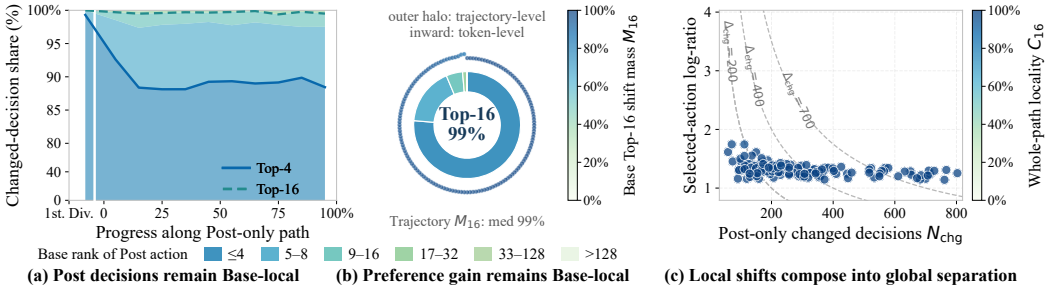


Figure A13: **Local candidate support persists over substantially longer Gain trajectories on OLMo-3-7B-Think and ARC-Challenge.** Base Top-16 remains highly inclusive throughout the Post-only path, with 99% of the positive preference increase occurring on Base Top-16 actions and a trajectory-level median M_{16} of 99%. Even trajectories containing hundreds of changed decisions remain highly local while accumulating large path-level policy differences.

C.4 Candidate inheritance along newly successful trajectories

Section 3.3 moves beyond shared-prefix comparisons by asking whether candidate inheritance persists after the post-trained policy has diverged onto a different, successful reasoning path. Here we provide the full construction of this analysis and test its robustness along three axes: domain shift for a fixed Base–Post pair, replication across distinct model families, and a substantially larger model setting.

Gain trajectories and Post-generated states. To test whether candidate inheritance persists after post-training enters a reasoning trajectory that the Base policy itself does not generate, we focus on *Gain* examples, where the Base model fails but the post-trained model succeeds. Both policies first generate freely,

$$\tau_{\mathrm{B}} = (a_1^{\mathrm{B}}, \ldots, a_{T_{\mathrm{B}}}^{\mathrm{B}}), \qquad \tau_{\mathrm{P}} = (a_1^{\mathrm{P}}, \ldots, a_{T_{\mathrm{P}}}^{\mathrm{P}}), \tag{21}$$

and let d denote their first token-level divergence.

After divergence, the post-trained trajectory induces states

$$s_t^{\mathrm{P}} = (x, a_{<t}^{\mathrm{P}}), \qquad t > d, \tag{22}$$

which need not be visited by greedy Base decoding. At every such Post-generated prefix, we evaluate both $\pi_{\mathrm{B}}(\cdot \mid s_t^{\mathrm{P}})$ and $\pi_{\mathrm{P}}(\cdot \mid s_t^{\mathrm{P}})$.

We focus on post-divergence positions at which the action selected by Post differs from the Base greedy decision,

$$\mathcal{D}_\tau = \left\{ t > d : a_t^{\mathrm{P}} \neq \arg\max_v \pi_{\mathrm{B}}(v \mid s_t^{\mathrm{P}}) \right\}. \tag{23}$$

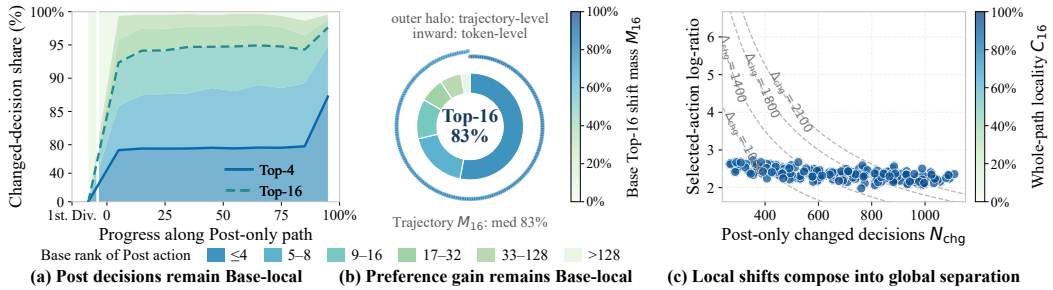


Figure A14: **Candidate inheritance persists at 32B with a broader effective radius.** For Qwen2.5-32B on OpenThoughts-Math, changed Post decisions remain predominantly Base-local after divergence, while Base Top-16 captures 83% of positive preference-shift mass and the median trajectory-level M_{16} is 83%. Hundreds of moderate local changes accumulate into large trajectory-level shifts, showing that scale broadens rather than eliminates the inherited candidate structure.

For each $t \in \mathcal{D}_\tau$, we record the Base rank of the Post-selected action,

$$r_t = \mathrm{rank}_{\mathrm{B}}\left(a_t^{\mathrm{P}} \mid s_t^{\mathrm{P}}\right), \tag{24}$$

and its Post–Base log-probability shift,

$$\ell_t = \log \pi_{\mathrm{P}}\left(a_t^{\mathrm{P}} \mid s_t^{\mathrm{P}}\right) - \log \pi_{\mathrm{B}}\left(a_t^{\mathrm{P}} \mid s_t^{\mathrm{P}}\right). \tag{25}$$

Candidate locality along the Post-only path. For Panel (a), the post-divergence suffix of each Gain trajectory is mapped to normalized progress from 0 to 100%, with the first divergence shown separately. At each progress bin, we report the distribution of r_t over Base-rank bands together with cumulative Top-4 and Top-16 coverage. Aggregation is performed at the trajectory level before averaging across examples so that long reasoning traces do not dominate the statistic.

Where post-training amplifies preference. Panel (b) asks whether the probability increases assigned by Post are concentrated on Base-local actions. We retain the positive part of the selected-action shift,

$$\ell_t^+ = \max(\ell_t, 0), \tag{26}$$

and define the fraction of positive shift mass attributable to Base Top-K as

$$M_K = \frac{\sum_{t\in\mathcal{D}_\tau} \ell_t^+ \, \mathbf{1}[r_t \le K]}{\sum_{t\in\mathcal{D}_\tau} \ell_t^+}. \tag{27}$$

We report both the aggregate token-level M_{16} and the distribution of trajectory-level M_{16} values. The inner ring of Panel (b) shows how positive shift mass is distributed across Base-rank bands, while the outer halo summarizes trajectory-level locality.

Composition along complete trajectories. Panel (c) measures how repeated local changes accumulate within each Gain trajectory. For trajectory τ, let

$$N_{\mathrm{chg}}(\tau) = |\mathcal{D}_\tau| \tag{28}$$

denote the number of post-divergence changed decisions, and define their mean selected-action shift as

$$\bar{\ell}_\tau = \frac{1}{N_{\mathrm{chg}}(\tau)} \sum_{t\in\mathcal{D}_\tau} \ell_t. \tag{29}$$

Their cumulative changed-decision shift is therefore

$$\Delta_{\mathrm{chg}}(\tau) = N_{\mathrm{chg}}(\tau)\bar{\ell}_\tau = \sum_{t\in\mathcal{D}_\tau} \ell_t. \tag{30}$$

We additionally define whole-path Top-16 locality

$$C_{16}(\tau) = \frac{\sum_{t\in\mathcal{D}_\tau} \mathbf{1}[r_t \le 16]}{N_{\mathrm{chg}}(\tau)}. \tag{31}$$

Panel (c) plots N_{chg} against $\bar{\ell}_\tau$, colors each trajectory by C_{16}, and overlays contours of constant Δ_{chg}.

Table A3: **Actionable selection headroom within the Base Top-8 candidate set.**

Model	Dataset	Active (%)	Base Top-1 Success	Top-8 Oracle Success	Headroom
DeepSeek-R1, 1.5B	MATH	54.3	62.1	84.5	**+22.4**
	AIME25	33.3	6.7	33.3	**+26.7**
	MMLU-Pro	39.4	23.4	43.1	**+19.7**
	GPQA-D	32.0	12.5	33.6	**+21.1**
Llama-3.2, 3B	MATH	38.6	44.5	66.9	**+22.4**
	AIME25	16.7	3.3	20.0	**+16.7**
	MMLU-Pro	48.5	41.4	71.9	**+30.5**
	GPQA-D	54.6	23.7	58.6	**+34.9**

Domain shift under a fixed model pair. We first hold the Qwen2.5-Math-1.5B Base–Post pair fixed and replace OpenThoughts-Math with MMLU-Pro, isolating the effect of task domain (Figure A11). Changed Post decisions remain strongly Base-local after trajectory divergence: Top-16 coverage stays above approximately 94% across most of the Post-only path, although the rank distribution is broader than on OpenThoughts-Math.

Preference amplification reveals greater trajectory-level heterogeneity. Base Top-16 accounts for 69% of aggregate positive shift mass, while the median trajectory-level M_{16} is 100%. Thus, the lower aggregate concentration is driven by a subset of trajectories with disproportionately large shifts on deeper-ranked actions rather than by a uniform loss of locality. MMLU-Pro therefore provides a useful boundary case: the local-to-global structure persists under domain shift, while its effective candidate radius remains graded.

Replication across model families. We next repeat the analysis with two distinct model families: DeepSeekMath-7B on MATH-500 and OLMo-3-7B-Think (Olmo et al., 2025) on ARC-Challenge (Clark et al., 2018) (Figures A12 and A13).

For DeepSeekMath-7B, Top-16 coverage remains high throughout the Post-only suffix despite greater variation in Top-4 coverage. Base Top-16 accounts for 96% of positive preference-shift mass, and the median trajectory-level M_{16} is 100%. The same organization therefore appears in a distinct 7B mathematical model.

OLMo-3-7B-Think provides a substantially different setting in both model family and task domain. On ARC-Challenge, Top-16 coverage remains near saturation across the Post-only path, while 99% of positive shift mass lies within Base Top-16 and the median trajectory-level M_{16} is 99%. Moreover, many Gain trajectories contain hundreds of changed decisions while maintaining high C_{16}, showing that the phenomenon persists even when global separation is built from long sequences of local changes.

Scaling to a larger model. Finally, we test whether the same structure persists at substantially larger scale using Qwen2.5-32B on OpenThoughts-Math (Figure A14). The inherited candidate region becomes broader but remains pronounced: after the first divergence, Base Top-16 covers approximately 92–95% of changed Post actions across most of the Post-only trajectory.

Preference amplification broadens accordingly. Base Top-16 contains 83% of aggregate positive shift mass, with a median trajectory-level M_{16} of 83%. At the same time, individual Gain trajectories contain several hundred changed decisions whose moderate local shifts accumulate into large Δ_{chg} while retaining high whole-path locality. Thus, increasing model scale broadens the effective inherited radius without eliminating the local-to-global structure.

Taken together, these extensions establish the same local-to-global structure across task domains, model families, and scales: even after post-training enters a different successful trajectory, its changed decisions remain largely grounded in the Base ranking, while repeated local preference changes accumulate into globally different reasoning behavior.

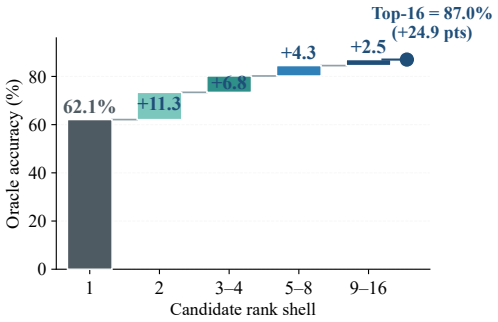


Figure A15: **Actionable headroom concentrates in shallow Base ranks on MATH-500.** Each bar gives the marginal oracle gain from an additional rank shell; most improvement is recovered within Base Top-8.

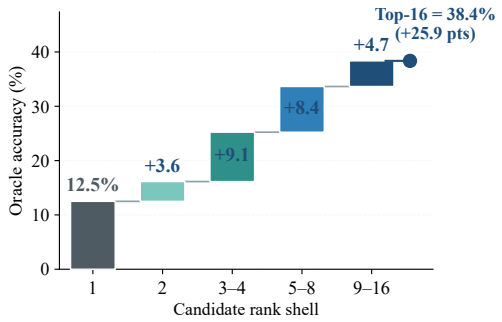


Figure A16: **The same shallow-headroom pattern holds on GPQA-D.** Most oracle improvement again comes from highly ranked Base candidates, with limited additional gain beyond Top-8.

Table A4: **The proposal–value decomposition persists on Qwen3-8B.** Adapted accuracy remains close to Base, while enabling ProQ yields additional end-to-end gains. Rescue/Keep accuracy evaluates local value ordering for beneficial and harmful alternatives, and trajectory Rescue/Degradation reports the corresponding incorrect→correct and correct→incorrect outcome transitions.

	End-to-End Accuracy			Local Value Diagnostics		Trajectory Transitions	
Benchmark	**Base Acc.**	**Adapted Acc.**	**ProQ Acc.**	**Rescue Acc.**	**Keep Acc.**	**Rescue (%)**	**Degradation (%)**
MATH	81.3	81.6	82.1	68.2	61.5	0.8	0.3
AIME25	63.3	66.7	73.3	40.0	58.3	10.0	3.3
MMLU-Pro	70.6	72.6	75.9	44.3	60.9	3.7	0.4
GPQA-D	51.8	52.8	57.1	60.6	60.4	5.8	1.5

D Extended Evaluation and Analysis of ProQ

D.1 Actionable headroom within the Base candidate set

Before evaluating the learned selector, we independently audit whether the model's own candidate set contains meaningful opportunities for better local decisions. At each audited high-uncertainty state s, we evaluate the Base Top-K candidates $\mathcal{C}_K(s) = \{c_1, \ldots, c_K\}$ using their candidate-conditioned downstream success values $R(s, c_j)$. This analysis measures the available selection headroom independently of selector quality.

We report four quantities over the audited high-uncertainty states $\mathcal{S}_{\text{audit}}$. A state is Active if the observed candidate-conditioned outcomes differ:

$$\max_{c_j \in \mathcal{C}_K(s)} R(s, c_j) > \min_{c_j \in \mathcal{C}_K(s)} R(s, c_j).$$

Active (%) is the fraction of audited states satisfying this condition. Base Top-1 Success and Top-K Oracle Success are both averaged over $\mathcal{S}_{\text{audit}}$:

$$A_1 = \mathbb{E}_{s \in \mathcal{S}_{\text{audit}}}[R(s, c_1)], \qquad A_K^{\text{oracle}} = \mathbb{E}_{s \in \mathcal{S}_{\text{audit}}}\left[\max_{c_j \in \mathcal{C}_K(s)} R(s, c_j)\right].$$

Headroom is $A_K^{\text{oracle}} - A_1$, measured in percentage points.

Table A3 reveals substantial selection opportunities across both mathematical and general reasoning tasks. For DeepSeek-R1-Distill-Qwen-1.5B, roughly 32–54% of audited high-uncertainty states are active, indicating that candidate identity frequently changes the eventual outcome. Across all audited states, Top-8 Oracle Success exceeds the Base Top-1 decision by 19.7–26.7 percentage points across all four benchmarks.

Where does the headroom come from? To localize where this selection headroom arises within the candidate ranking, we decompose the Top-K oracle gain by Base-rank shell. Starting from the Base Top-1 decision, we successively admit candidates from rank 2, ranks 3–4, 5–8, and 9–16, and report the marginal oracle improvement contributed by each shell. Equivalently, the cumulative gain after each shell corresponds to the oracle attainable with Base Top-2, Top-4, Top-8, and Top-16 candidates.

Figures A15 and A16 show a consistent pattern across MATH and GPQA-D. Most actionable headroom emerges within the first few Base ranks: rank 2 and ranks 3–4 contribute the largest increments, while candidates from ranks 5–8 add smaller gains and ranks 9–16 contribute only modestly. Thus, the available improvement is not primarily driven by rare deep candidates, but by alternatives that the Base model already ranks highly.

Together, these results show that consequential reasoning failures are often not caused by the absence of useful candidates. A substantial fraction of recoverable value already lies within a shallow Base candidate set, providing the local headroom that value-based selection can exploit.

D.2 From local value interventions to global gains

Candidate-conditioned scoring is useful only if two conditions hold at deployment: the model must identify *when* value-based selection is worth invoking, and the resulting local interventions must accumulate into favorable changes at the level of complete trajectories. We examine these two aspects separately. First, we vary the trigger that determines where ProQ intervenes. We then directly decompose the resulting trajectory-level accuracy change into failures rescued and previously correct trajectories degraded by intervention. Together, these analyses characterize how candidate scores translate into global reasoning improvement.

Selective deployment. We first vary the criterion used to activate value-based selection. For entropy-based triggering, a threshold H_p invokes ProQ only when the standard-token entropy exceeds the p-th percentile of a label-free calibration distribution. We additionally consider a disagreement-based trigger defined by the KL divergence between the current policy and a stronger 32B reference model, providing an alternative signal for locally consequential decisions. For both trigger families, we sweep the activation threshold from permissive to highly selective and measure final trajectory-level accuracy.

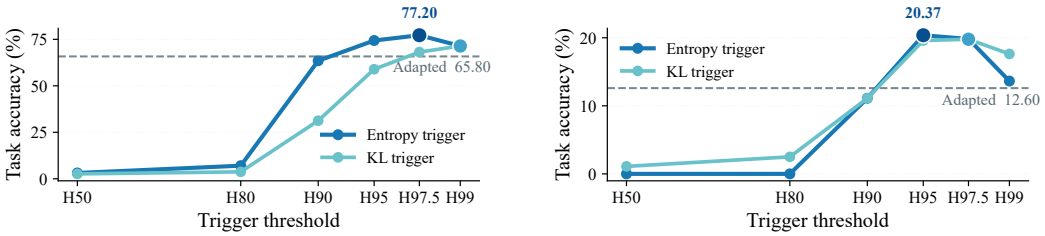
The resulting curves in Figure A17 reveal a pronounced selective-intervention regime. When the trigger is permissive, ProQ is invoked at many positions and final performance can fall substantially below the adapted policy. As the trigger is restricted to a smaller high-uncertainty or high-disagreement subset, accuracy rises sharply and exceeds the adapted baseline. Performance eventually declines again under the most conservative thresholds, producing an interior optimum rather than a monotonic benefit from increasingly sparse intervention.

This non-monotonic behavior is consistent with a coverage–selectivity tradeoff. Permissive triggering exposes generation to many local interventions, including positions where changing the proposal may provide little benefit, so small local errors can accumulate across long trajectories. Increasing the threshold concentrates intervention on a smaller set of more ambiguous decisions, improving the average usefulness of each intervention. At the opposite extreme, however, very sparse triggering leaves fewer opportunities to correct consequential decisions. The strongest performance therefore arises when ProQ acts as a selective local controller rather than a universal replacement for the standard next-token policy.

Replication on Qwen3-8B. Table A4 repeats the decomposition on the substantially stronger Qwen3-8B model. Across benchmarks, Adapted accuracy remains close to Base, while enabling ProQ produces additional end-to-end gains. The corresponding trajectory transitions are consistently favorable, with failure-to-success rescues exceeding success-to-failure degradations in every benchmark.

Table A5: **Comparison of KL- and entropy-based selector triggers.** Accuracy and the fraction of generations reaching the response-length cap are reported in %.

Dataset	Trigger	Metric	H50	H80	H90	H95	H97.5	H99
MATH-500	KL	Accuracy	2.64	3.73	31.25	58.92	68.13	71.42
		Cap rate	96.21	89.57	71.09	37.44	27.49	25.59
	Entropy	Accuracy	**3.12**	**7.04**	**63.47**	**74.36**	**77.20**	**71.42**
		Cap rate	97.16	86.26	24.17	11.37	18.01	24.64
GPQA Diamond	KL	Accuracy	**1.18**	**2.53**	**11.11**	19.53	19.70	**17.68**
		Cap rate	98.48	96.97	87.88	74.75	75.76	82.32
	Entropy	Accuracy	0.00	0.00	**11.11**	**20.37**	**19.87**	13.64
		Cap rate	98.99	96.46	72.73	55.05	56.57	75.76



(a) **MATH-500.** (b) **GPQA Diamond.**

Figure A17: **Local value-based selection is most effective under selective intervention.** We compare two trigger families—standard-token entropy, and KL-based deviation with a stronger 32B reference model—across MATH-500 and GPQA Diamond. Despite using different trigger signals, all exhibit the same qualitative tradeoff: dense intervention introduces too many harmful corrections, while overly conservative triggering misses useful opportunities. The strongest trajectory-level gains arise in an intermediate selective regime.

D.3 Ablations on the ProQ objective

We study the ProQ training objective along four questions: *(i)* whether candidate supervision should be anchored only to the current preferred action or cover all success–failure comparisons, *(ii)* whether selection should be trained through a set-level objective or directly aligned with the final Top-1 decision, *(iii)* whether simply increasing selection-loss strength is sufficient, and *(iv)* how strongly the standard-token proposal distribution should be preserved. Together, these ablations motivate the final objective

$$\mathcal{L}_{\text{ProQ}} = \mathcal{L}_{\text{pair}} + \lambda_{\text{top1}}\mathcal{L}_{\text{top1}} + \lambda_{\text{prop}}\mathcal{L}_{\text{propose}}.$$

How broadly should candidate comparisons be supervised? We first compare supervision anchored only to the Base-preferred candidate with all-pairs success–failure ranking. On the MATH reference, the Base-anchored formulation achieves 65.6% selection accuracy, whereas all-pairs ranking reaches 75.0% under the same early protocol. This indicates that reliable selection benefits from exploiting the full within-state outcome structure rather than learning only whether individual alternatives should override the current preference.

Should training optimize the successful set or the eventual winner? A more consequential design choice is how the objective aligns with the inference rule $c^\star = \arg\max_c Q_\phi(s, c)$. Our set-selection variant encourages probability mass to concentrate on any successful candidate, whereas the final Top-1 margin explicitly separates the strongest successful candidate from the strongest unsuccessful competitor.

On the MATH reference, pairwise ranking with set selection reaches 78.1%, while replacing the set objective with the Top-1 margin improves accuracy to 82.8% under the matched optimization budget.

Increasing the pairwise and set-selection weights does not reproduce this gain, reaching 77.3% despite stronger supervision. This suggests that the gain comes from sharpening the separation between the strongest successful and unsuccessful candidates, rather than simply increasing optimization pressure.

How much proposal preservation is necessary? The selector and standard-token proposer share the same LoRA-adapted representation, so selection learning can also alter ordinary generation. Removing the KL constraint improves training fit but increases standard-vocabulary drift by roughly 20–$38\times$, without improving held-out Top-1 selection. At the other extreme, increasing the KL coefficient provides no consistent gain: representative stronger-preservation settings obtain similar selection accuracy to the default configuration. Increasing the number of positions used for KL regularization likewise does not yield a systematic selector improvement.

These results favor *moderate preservation* rather than either unconstrained adaptation or increasingly strong anchoring. For the current ProQ checkpoint, the adapted standard-token policy remains close to the Base model, while enabling ProQ selection substantially improves performance on the same macro evaluation. Thus, most of the additional reasoning gain is realized through the learned selection mechanism while proposal behavior remains close to Base.

D.4 Proposal-preservation diagnostics across decoding states

Because ProQ uses the same adapted LLM for standard-token proposal and reserved-token candidate scoring, value learning and proposal preservation must coexist within a shared representation. We therefore characterize how the adapted standard-token policy differs from Base across both selector-triggered and sampled non-triggered states. Our diagnostics separate candidate identity and local ordering from full-distribution drift, and further examine where these changes occur and how they evolve during training.

Metrics. For each state s, *Overlap@*K measures the fraction of candidates shared by the Base and adapted Top-K sets,

$$\mathrm{Overlap@}K = \frac{|\mathrm{TopK}_B(s) \cap \mathrm{TopK}_A(s)|}{K},$$

with Overlap@1 indicating whether their preferred token is unchanged. To measure preservation beyond candidate identity, we take the union of the Base and adapted Top-8 sets and compute *Spearman's* ρ and *Kendall's* τ_b over their Base- and adapted-policy rankings; the former measures rank correlation, while the latter measures pairwise ordering agreement. We additionally compare full standard-token distributions using *total variation*,

$$\mathrm{TV}(p_B, p_A) = \frac{1}{2} \sum_v |p_B(v) - p_A(v)|,$$

and *KL divergence*,

$$D_{\mathrm{KL}}(p_B \| p_A),$$

where smaller values indicate less proposal drift. Together, these metrics distinguish preservation of the candidate set and its local ordering from broader changes in the standard-token probability distribution.

Overall preservation profile. Figure A18 summarizes the resulting proposal geometry for DeepSeek-R1-Distill-Qwen-1.5B on OlympiadBench. Across triggered and non-triggered states, candidate overlap and local rank agreement remain high, whereas full-distribution TV and KL are systematically larger at triggered positions. The joint diagnostic therefore reveals a structured form of adaptation: the largest changes occur in probability allocation, while the high-ranked candidate structure changes substantially less.

Where proposal drift occurs. Figure A19 relates Base uncertainty to position-level proposal drift. Each decoding state is represented by

$$\left(H_B(s), \mathrm{TV}\left(\pi_B^{\mathrm{std}}(\cdot \mid s), \pi_\phi^{\mathrm{std}}(\cdot \mid s)\right)\right),$$

where $H_B(s)$ is the Base standard-token entropy. Because the number of states is large, we visualize their density with hexbins and separate non-triggered from selector-triggered positions.

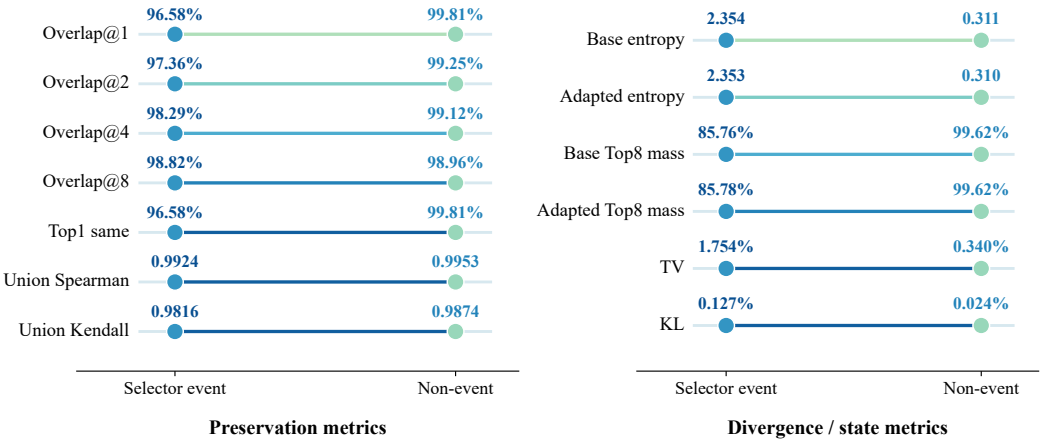


Figure A18: **Multi-metric proposal-preservation diagnostics for ProQ** on DeepSeek-R1-Distill-Qwen-1.5B, evaluated on OlympiadBench. We compare the Base and ProQ-adapted standard-token policies at triggered and non-triggered states using candidate overlap, local ordering, uncertainty, retained probability mass, and full-distribution drift. This provides a joint view of how proposal behavior changes across complementary notions of distributional similarity.

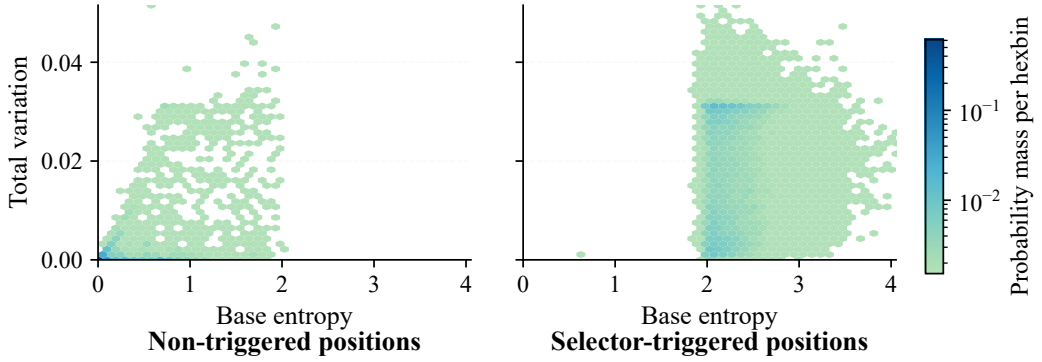


Figure A19: **Proposal drift is concentrated in the high-uncertainty regime where ProQ intervenes.** Hexbins relate Base entropy to Base–ProQ total-variation distance at non-triggered and triggered positions on OlympiadBench. Triggered positions occupy a distinctly higher-entropy region and exhibit somewhat larger drift, revealing where proposal adaptation is concentrated.

Non-triggered states concentrate in a low-entropy, low-drift regime. Triggered states occupy substantially higher entropy and exhibit a broader TV distribution, showing that proposal adaptation is preferentially concentrated at the uncertain states where ProQ intervenes. The position-level view therefore localizes the distributional differences observed in the aggregate statistics rather than attributing them to a small number of isolated outliers.

Probability redistribution versus candidate replacement. Figure A20 compares the marginal distributions of Base entropy, KL divergence, total variation, and Overlap@8. Triggered states shift toward higher entropy and larger KL/TV values, confirming that their standard-token distributions undergo more adaptation. In contrast, Overlap@8 remains concentrated near complete agreement for both triggered and non-triggered states. This separates two effects that aggregate divergence alone cannot distinguish: ProQ can substantially reallocate probability among plausible alternatives while largely retaining the identity of the high-ranked candidate set.

Training dynamics. Figure A21 traces the two optimization signals corresponding to ProQ's proposal–value decomposition. The selection objective combines within-state pairwise ranking with

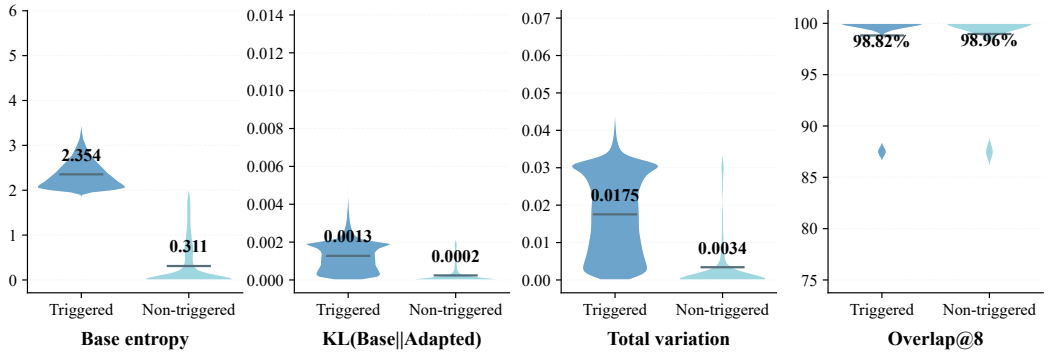


Figure A20: **Distributional drift need not imply a change in the candidate set.** Marginal distributions show that triggered states have higher entropy and larger KL/TV deviation than non-triggered states, while Base–ProQ Top-8 overlap remains concentrated near complete recovery in both regimes. Thus, changes in probability allocation are substantially larger than changes in candidate identity.

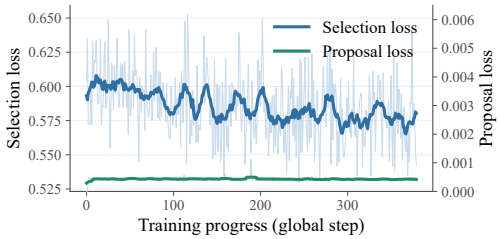


Figure A21: **Value learning progresses without a corresponding growth in proposal divergence.** During training, the selection objective steadily decreases while the standard-vocabulary KL to Base remains low and approximately stable. The two trajectories separate improvement in candidate-value learning from changes in the standard-token proposal distribution.

the Top-1 decision constraint, while the proposal KL measures standard-vocabulary divergence from Base. Over training, the selection objective decreases despite minibatch-level fluctuations, whereas proposal KL remains low and approximately stable. Thus, continued improvement in candidate-value learning does not require progressively increasing divergence in the standard-token proposal policy.

Taken together, these diagnostics show that ProQ does not preserve proposal behavior by preventing representation change altogether. Instead, adaptation is concentrated at uncertain decisions and primarily reorganizes probability among largely stable high-ranked candidates, while standard-token divergence remains controlled as value learning proceeds.

D.5 Training and inference cost

We separate ProQ's computational cost into four components: *counterfactual data construction*, *value-learning optimization*, *parameter overhead*, and *inference-time intervention*. We report DeepSeek-R1-Distill-Qwen-1.5B in detail and apply the same accounting protocol to the remaining models and policy-optimization baselines.

Data construction. We construct the training asset from a 4,096-problem sample of OpenThoughts-Math. After filtering two over-length prompts and reserving 128 examples each for validation and test, 3,838 problems remain for training. For each problem, the Base model first generates one greedy trajectory; two highest-entropy branch states are then selected, and each of the Base Top-8 candidates at each state is forced and greedily continued under the fixed Base policy. This produces 7,676 branch states and 61,408 candidate-conditioned continuations. Among them, 2,005 states are *active*, meaning that their candidate set contains both successful and unsuccessful downstream outcomes. The resulting reward-labeled asset is collected once and reused across ProQ optimization runs.

ProQ optimization. The selected ProQ checkpoint is obtained after 300 optimizer updates with global batch size 48, corresponding to 14,400 total state exposures, of which 11,708 are active-state exposures. Relative to the available pools, these correspond to 1.88 full-state and 5.84 active-state

Table A6: **End-to-end inference measurements for ProQ.** Shown on DeepSeek-R1, 1.5B. Average generated response length and wall-clock inference time are reported for Base and ProQ under matched evaluation settings. Relative changes from Base are shown in parentheses.

Benchmark	Output Length (tokens)		Inference Time (s / question)	
	Base	ProQ	Base	ProQ
MATH-500	3629	3128 (−13.8%)	26.72	36.07 (+35.0%)
OlympiadBench	5983	5290 (−11.6%)	34.99	54.05 (+54.5%)
MMLU-Pro	4445	3625 (−18.4%)	31.48	44.51 (+41.4%)

exposure equivalents. We use exposure equivalents rather than epochs because training employs stratified repeated sampling rather than uniform passes through the dataset.

The Base model contains 1.777B parameters. ProQ introduces 18.46M LoRA parameters together with 24.6K trainable reserved-token output coordinates, for approximately 18.49M effective updated parameters (1.04% of the Base model).

Trajectory-matched policy-optimization baselines. Our primary matching criterion is the number of reward-labeled generations used for learning. ProQ constructs two branch states for each of the 3,838 training problems and evaluates the Base Top-8 candidates at every state, yielding $3,838 \times 2 \times 8 = 61,408$ candidate-conditioned continuations. GRPO samples $48 \times 8 = 384$ trajectories per update, so 160 steps yield $61{,}440$ trajectories; DAPO and RLPT-GRPO use the same rollout budget. Because DAPO dynamically discards uninformative prompt groups, we instead stop it when cumulative generations reach approximately 61.4k. All methods use the same problem pool and task reward.

Inference overhead. Outside triggered positions, ProQ follows ordinary autoregressive decoding. When the entropy trigger fires, the current standard-token Top-K proposals are each appended to the current state and evaluated through one-token branch forwards using the same LLM. For each candidate, ProQ reads the reserved-token logits and maps them to a scalar candidate-conditioned score $Q_\phi(s, c)$. The highest-scoring candidate is selected, after which normal autoregressive generation resumes. Accordingly, ProQ introduces no separate critic and does not generate candidate-level continuations at inference; the additional computation consists only of K one-token candidate evaluations at sparse intervention points.