BENIGN TASKS, UNSAFE ORDERS: THE LOCAL-TO-GLOBAL SAFETY GAP IN CONTINUAL LLM FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual fine-tuning can alter the safety of large language models even when all downstream tasks are benign. We show that this risk is inherently path dependent: applying the same tasks in different orders can lead to substantially different terminal safety outcomes. More importantly, we identify a local-to-global safety gap: state-conditioned measurements can accurately characterize the immediate safety effect of a candidate task, yet even the realized one-step effect can be a poor indicator of safety after the remaining tasks are completed. Thus, reliable local monitoring does not necessarily support reliable long-horizon task-order decisions. Motivated by this observation, we formulate continual task ordering as a terminal path-value estimation problem and introduce **P-DC-SCTR**, which jointly models the current state, local transition effects, and remaining adaptation path. Experiments across multiple continual fine-tuning settings consistently reveal strong order-dependent safety variation and confirm the local-to-global gap. Path-aware modeling further provides more informative estimates of terminal safety consequences than local signals, while reliable end-to-end ordering remains an open challenge. These findings suggest that safety under continual adaptation should be modeled over training trajectories rather than isolated updates.

1 INTRODUCTION

Large language models (LLMs) are commonly fine-tuned for downstream tasks and applications. In continual fine-tuning, a model learns multiple downstream tasks while preserving its safety behavior against harmful requests. However, safety can degrade even when the downstream training data are benign (Qi et al., 2024; Jan et al., 2025). Meanwhile, task order is known to affect continual-learning outcomes (Bell & Lawrence, 2022; Li & Hiratani, 2025), and the ordering of safety and capability tuning can influence safety forgetting in sequential LLM fine-tuning (Ung et al., 2024). Yet it remains unclear whether the order of the same benign downstream tasks alone can systematically alter terminal safety. We isolate this question by holding the task set, training data, and per-task training budget fixed and varying only task order.

We find substantial order-dependent variation in terminal safety. Across eight task trees, terminal-ASR ranges span 20.5–29.5 percentage points, showing that terminal safety is strongly sensitive to task order across the evaluated settings. This variation makes task ordering itself consequential for safety under continual fine-tuning. It therefore raises a natural decision problem: when several tasks remain, how should the next task be selected? Answering this question requires identifying which information about a candidate task is actually informative for its long-horizon safety outcome.

A natural strategy is to estimate the immediate safety effect of each candidate task and greedily choose the locally safer option. This strategy turns a long-horizon ordering problem into a sequence of one-step decisions: if a candidate produces a safer next state, it would appear to be the safer choice for the remaining training trajectory. However, our experiments show that this intuition does not hold. Even when we eliminate local prediction error and use the realized one-step safety outcome after fully training each candidate task, greedy local decisions still fail to reliably recover low-risk terminal paths. Improving local safety prediction alone therefore cannot resolve the ordering problem. The key difficulty is that one-step safety and long-horizon terminal safety are different decision targets.

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.

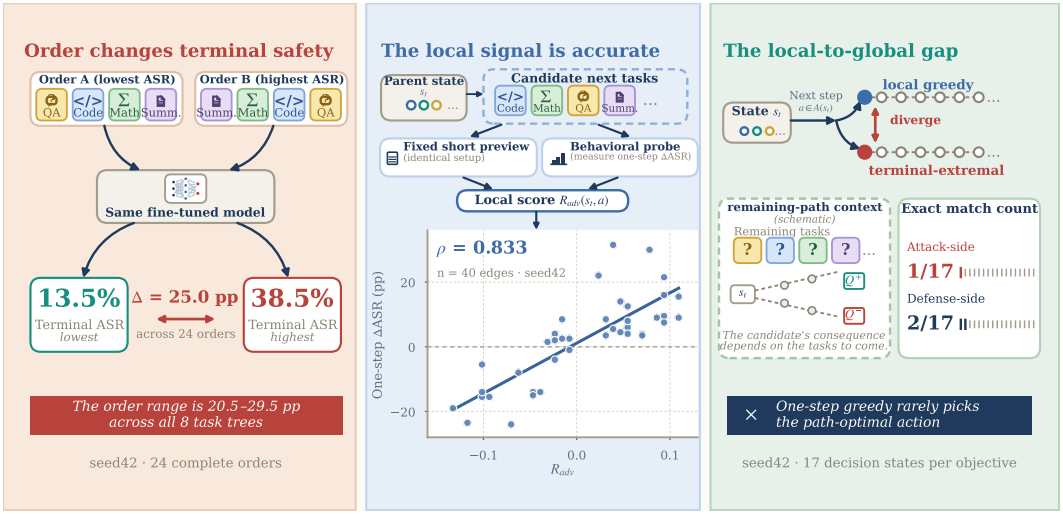


Figure 1: Motivating observations. **Left:** changing only the order of the same benign tasks leads to substantially different terminal safety outcomes. **Right:** accurate local safety information does not reliably identify the terminal path value associated with selecting a task.

Why can even exact one-step safety information fail to predict the eventual outcome? The reason is each task update changes not only the model's current safety behavior, but also the state from which all subsequent tasks are learned. Consequently, the effect of a task is state-conditioned: the same task can induce different safety changes depending on the preceding training path. More importantly, choosing a task at the current state changes the starting point for the next update, which in turn changes the effects of later tasks along the remaining trajectory. An immediate safety improvement can therefore be altered or even reversed by subsequent adaptation, so the local effect of a candidate need not reflect its terminal effect after the remaining tasks are completed. We call this mismatch between immediate safety effects and long-horizon terminal outcomes the *local-to-global safety gap*.

Motivated by this gap, we reformulate continual task ordering as a terminal path-value estimation problem. Rather than predicting only the immediate safety effect of a candidate task, we estimate the extreme terminal safety values attainable after that choice over the remaining adaptation path. We formalize these quantities as high- and low-risk terminal path values, corresponding to the most and least risky terminal outcomes reachable under different orderings of the remaining tasks. Based on this formulation, we introduce P-DC-SCTR, which combines information about the current state, the candidate transition, and the remaining-task context to estimate these path values. In this way, local measurements characterize the immediate transition, while path-value estimation accounts for the remaining adaptation path.

Our contributions are threefold: **1**) We show that task order substantially changes terminal safety under benign continual fine-tuning, even when the task set, training data, and per-task training budget are fixed. **2**) We identify the *local-to-global safety gap*: accurate local transition information, including realized one-step safety effects, is insufficient for reliable long-horizon task-order decisions. **3**) We formulate task ordering as terminal path-value estimation and show that path-aware modeling predicts terminal path values more effectively than local safety signals. However, this prediction advantage does not consistently translate into better complete task orders. This reveals a further distinction between path-value predictability and path controllability: terminal path values contain learnable structure, while reliable end-to-end ordering remains an open challenge.

2 Related Work

Safety-aligned LLMs can lose their safety behavior during downstream adaptation even when the fine-tuning data are benign. Qi et al. (2024) show that fine-tuning aligned models on benign datasets can degrade safety alignment. Subsequent work reveals that this degradation is heterogeneous across both tasks and training examples. Multitask-Bench (Jan et al., 2025) reports substantial

task-dependent differences in safety degradation across benign downstream tasks, while Guan et al. (2025) show that a small subset of training examples can contribute disproportionately to alignment degradation. Together, these studies show that the safety consequences of benign fine-tuning depend on the adaptation data at both the task and sample levels. A complementary line of work seeks to preserve safety during downstream adaptation through constrained parameter updates or perturbation-aware training (Hsu et al., 2024; Huang et al., 2024; 2025; Lu et al., 2025). More recent work considers safety preservation explicitly under continual adaptation. Unforgotten Safety (Alssum et al., 2025) frames safety degradation as catastrophic forgetting and evaluates continual-learning strategies for preserving alignment, while Bach et al. (2026) study alignment drift across continual domain tasks and mitigate it through gradient-based sample selection. These approaches primarily ask how safety degradation can be characterized, prevented, or mitigated during adaptation. Our focus is complementary: given a fixed set of benign tasks, we study how their safety effects depend on the training trajectory and whether local safety information is sufficient to reason about terminal safety.

Continual LLM fine-tuning exposes a model to sequential task updates and can induce interference with previously acquired knowledge, capabilities, or safety behavior. InsCL (Wang et al., 2024a), for example, uses instruction-aware replay to preserve previously acquired knowledge across multiple tasks and training orders. Safety-oriented continual-learning work similarly treats alignment degradation during downstream adaptation as a forgetting problem (Alssum et al., 2025; Bach et al., 2026). Together, these studies highlight the importance of sequential training dynamics for retaining capabilities and safety during continual adaptation. Task ordering has also been studied in continual and sequential learning. Bell & Lawrence (2022) show that reordering tasks can substantially affect catastrophic forgetting and continual-learning performance, while Li & Hiratani (2025) study how task ordering shapes multi-task continual-learning outcomes and how favorable orders can be selected. Related work on language-model adaptation has also examined how the ordering of intermediate fine-tuning tasks affects downstream performance (Chen et al., 2024). Most closely related to our safety setting, Ung et al. (2024) show that the ordering of safety and downstream capability tuning affects safety forgetting in chained LLM fine-tuning. Our setting isolates a more specific question. We start from the same safety-aligned model and vary only the order of a fixed set of benign downstream tasks while holding their data and training budgets fixed. Beyond asking whether task order affects terminal safety, we ask whether the safety effect observed after a candidate next task is informative about terminal safety after all remaining tasks are completed. This distinction motivates reasoning about safety over the remaining training trajectory rather than treating the immediate effect of an isolated update as the task-ordering objective.

3 Problem Formulation and Motivating Analysis

3.1 Controlled Continual Fine-Tuning

We consider continual fine-tuning over a fixed set of benign downstream tasks. At step t, the training state is

$$s_t = (\theta_t, R_t), \tag{1}$$

where θ_t denotes the current model parameters and R_t contains the tasks that remain to be learned. Within each permutation tree, we fix the task set, training data, per-task training budget, safety anchor, and optimization configuration, varying only task order. This controlled setting isolates the effect of task order on terminal safety.

For a candidate task $a \in R_t$, let F denote the full task-level fine-tuning transition. From state s_t, applying a yields

$$s_{t+1} = F(s_t, a) = (\theta_{t+1}, R_t \setminus \{a\}), \tag{2}$$

where θ_{t+1} is obtained by fine-tuning θ_t on the data for task a. For a permutation π of the remaining tasks, $F(s, \pi)$ denotes the state reached by applying the tasks in π sequentially from state s.

Let $\mathcal{A}(s)$ denote the safety risk at state s, measured by attack success rate (ASR). The realized one-step safety change of candidate task a is

$$r(s, a) = \mathcal{A}(F(s, a)) - \mathcal{A}(s). \tag{3}$$

For candidates from the same parent state s, $\mathcal{A}(s)$ is constant, so $r(s, a)$ exactly ranks candidates by successor-state risk $\mathcal{A}(F(s, a))$. This allows us to separate errors in estimating local effects from the

more fundamental question of whether one-step information is the correct target for long-horizon task-order decisions.

3.2 The Local-to-Global Safety Gap

The local-to-global gap does not imply that realized safety changes fail to compose along a completed trajectory. For any completed trajectory $(s_0, a_0, s_1, \ldots, a_{T-1}, s_T)$, the one-step changes telescope exactly:

$$\sum_{t=0}^{T-1} r(s_t, a_t) = \mathcal{A}(s_T) - \mathcal{A}(s_0). \tag{4}$$

The difficulty is therefore not how realized changes aggregate, but that future transition effects depend on the states created by earlier choices. Choosing a_t changes s_{t+1}, which in turn changes the safety effects of tasks applied later in the trajectory.

This limitation persists even when local prediction error is removed: realized one-step rankings need not identify the action associated with the corresponding terminal-extremal continuation. We evaluate this gap in Section 5.3.

The local-to-global safety gap is therefore a target mismatch, not merely a prediction error. The immediate successor-state risk $\mathcal{A}(F(s, a))$ characterizes only the next transition, whereas the long-horizon target must account for terminal outcomes reachable after the remaining tasks are applied. A locally safer successor state therefore need not correspond to a lower-risk terminal continuation. This motivates the terminal path-value formulation below.

3.3 Terminal Path-Value Formulation

We therefore define terminal path values over the remaining tasks. Let $\Pi(R_s)$ denote all permutations of the tasks remaining at state s. The highest- and lowest-risk terminal values reachable from s are

$$V^+(s) = \max_{\pi \in \Pi(R_s)} \mathcal{A}(F(s, \pi)), \qquad V^-(s) = \min_{\pi \in \Pi(R_s)} \mathcal{A}(F(s, \pi)). \tag{5}$$

Here, $V^+(s)$ corresponds to the controlled high-risk stress-test objective, whereas $V^-(s)$ corresponds to the safety-oriented low-risk objective. At a terminal state, $V^+(s) = V^-(s) = \mathcal{A}(s)$.

For a candidate task $a \in R_s$, we define the corresponding terminal action values as

$$Q^+(s, a) = V^+(F(s, a)), \qquad Q^-(s, a) = V^-(F(s, a)). \tag{6}$$

Thus, $Q^+(s, a)$ and $Q^-(s, a)$ are the highest- and lowest-risk terminal values reachable after choosing a first and considering all permutations of the remaining tasks. They characterize terminal-extremal continuations conditioned on the current choice rather than its immediate safety effect. The state and action values satisfy

$$V^+(s) = \max_{a \in R_s} Q^+(s, a), \qquad V^-(s) = \min_{a \in R_s} Q^-(s, a). \tag{7}$$

This formulation shifts the task-order decision from one-step safety change to terminal path value. If $Q^+(s, a)$ and $Q^-(s, a)$ were known, candidate tasks could be compared by the extreme terminal outcomes reachable after each current choice. The resulting methodological question is whether these terminal path values can be estimated without exhaustively evaluating all descendant permutations.

4 Method

Section 3 distinguishes one-step safety effects from terminal path values. Accordingly, we separate local transition measurement from terminal path-value estimation. DC-SCTR uses short, state-conditioned training previews to characterize the local transition induced by each candidate task. Importantly, these measurements describe only the immediate response of the current model. P-DC-SCTR addresses the distinct long-horizon target by combining the current state, local transition information, and remaining-task context to estimate the terminal path values $\hat{Q}^+(s, a)$ and $\hat{Q}^-(s, a)$ defined in Section 3.3. Whether these path-value estimates can support reliable task-order decisions over the complete adaptation trajectory is evaluated separately in Section 5.4.

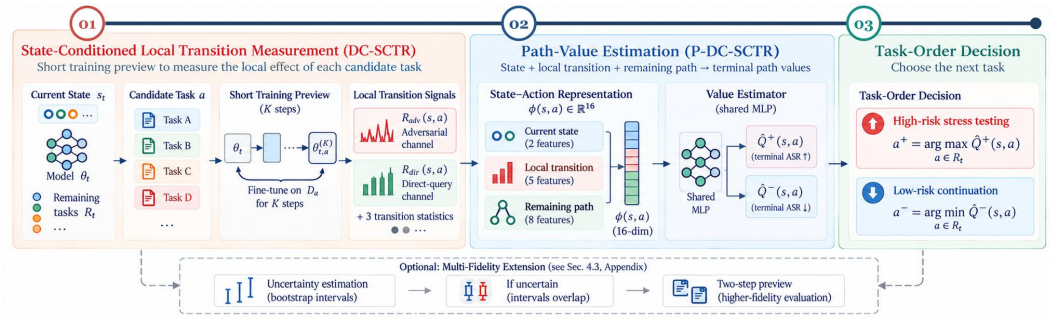


Figure 2: Overview of the proposed framework. DC-SCTR uses short training previews to characterize state-conditioned local transitions. P-DC-SCTR combines information about the current state, candidate transition, and remaining-task context to estimate the terminal extremal path values $\hat{Q}^+(s,a)$ and $\hat{Q}^-(s,a)$. Local transition measurement and terminal path-value estimation are treated as distinct problems.

4.1 State-Conditioned Local Transition Measurement

Computing the realized one-step effect $r(s,a)$ requires completing the full task-level transition $F(s,a)$ for every candidate. We instead obtain a local measurement from a K-step training preview. For state $s_t = (\theta_t, R_t)$ and candidate $a \in R_t$, we define

$$\tilde{\theta}_{t,a}^{(K)} = \mathrm{FT}_K(\theta_t, \mathcal{D}_a), \tag{8}$$

where FT_K follows the full fine-tuning procedure but stops after K updates. The preview is used only for measurement and does not modify the actual continual fine-tuning trajectory.

Each preview starts from the current parameters θ_t rather than from a common base model, conditioning the measurement on the training prefix that produced the current state. The same candidate can therefore induce different local responses after different preceding task orders. DC-SCTR captures this state dependence rather than assigning each task a fixed safety effect.

From each preview, DC-SCTR extracts two behavioral signals. The adversarial channel measures the change in free-generation behavior under harmful prompts, producing $R_{\mathrm{adv}}(s,a)$, while the direct-query channel measures the directional safety change under direct harmful queries, producing $R_{\mathrm{dir}}(s,a)$. We also retain three prompt-level transition statistics, B, Q, and C, from the same measurement pipeline. Together, these quantities characterize the local behavioral transition induced by candidate a at the current state, rather than its terminal path value.

4.2 P-DC-SCTR: Terminal Path-Value Estimation

P-DC-SCTR estimates the terminal path values $Q^+(s,a)$ and $Q^-(s,a)$ defined in Section 3.3. Estimating these values requires the current model state, the candidate's local effect, and the remaining adaptation context. Accordingly, we organize the representation as

$$\phi(s,a) = \Big[\underbrace{\phi_{\mathrm{state}}(s)}_{\text{where we are}}, \underbrace{\phi_{\mathrm{trans}}(s,a)}_{\text{local response}}, \underbrace{\phi_{\mathrm{path}}(s,a)}_{\text{what remains}}\Big]. \tag{9}$$

The resulting 16-dimensional representation contains two parent-state behavioral statistics, the candidate-specific signals R_{adv} and R_{dir}, the three transition statistics B, Q, and C, the current depth, a four-dimensional candidate-task indicator, and a four-dimensional remaining-task indicator. This grouping separates information about the current state and local transition from the context of the remaining adaptation path.

Path-value targets. During development, the complete permutation tree provides exact targets $Q^+(s,a)$ and $Q^-(s,a)$ with respect to the enumerated tree, computed from the terminal descendants of each state–action pair. This enumeration is used only to construct supervision. At held-out

inference time, P-DC-SCTR has no access to descendant states, transitions, or terminal outcomes; it receives only the current-state features, candidate-task information, remaining-task context, and DC-SCTR preview measurements.

Estimator and objective. We use a shared multilayer perceptron with two hidden layers of width 32 to predict both terminal path values:

$$\left(\hat{Q}^{+}(s,a), \hat{Q}^{-}(s,a)\right) = f_{\psi}(\phi(s,a)) \,. \tag{10}$$

The regression component uses Huber loss with $\delta = 0.1$. To also encourage correct comparisons among candidates sharing the same parent state, we add a pairwise ranking loss. For $\sigma \in \{+,-\}$ and candidates $a_i, a_j \in R_s$, let

$$y_{ij}^{\sigma} = \operatorname{sign}\left(Q^{\sigma}(s,a_i) - Q^{\sigma}(s,a_j)\right). \tag{11}$$

For non-tied pairs, the ranking loss is

$$\mathcal{L}_{\text{rank}} = \sum_{\sigma\in\{+,-\}} \sum_{(s,a_i,a_j)\in\mathcal{P}_{\sigma}} \log\left[1 + \exp\left(-y_{ij}^{\sigma}\left(\hat{Q}^{\sigma}(s,a_i) - \hat{Q}^{\sigma}(s,a_j)\right)\right)\right], \tag{12}$$

where $\mathcal{P}_{\sigma}$ contains non-tied candidate pairs sharing the same parent state. The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{value}} + 0.1\,\mathcal{L}_{\text{rank}}, \tag{13}$$

where $\mathcal{L}_{\text{value}}$ is the Huber loss over Q^{+} and Q^{-}. The two terms respectively encourage accurate path-value estimation and correct within-state candidate comparisons.

Development and inference. Hyperparameters are selected on the development tree using leave-one-parent-out validation, keeping all outgoing edges from the same parent in a single split. The selected estimator is then fitted on the complete development tree, frozen, and evaluated without refitting on whole held-out trees. For the two continuation objectives, the point-estimate choices are

$$a^{+} = \arg\max_{a\in R_s} \hat{Q}^{+}(s,a), \qquad a^{-} = \arg\min_{a\in R_s} \hat{Q}^{-}(s,a). \tag{14}$$

Here, a^{+} corresponds to the high-risk stress-test objective and a^{-} to the low-risk safety-oriented objective. These choices test whether path-value prediction transfers to sequential decision making; they are not assumed to recover an extremal complete task order. Accordingly, path-value prediction and complete-order selection are evaluated separately. An exploratory multi-fidelity extension is described in Appendix E.5.

5 Experiments

Our experiments address three questions. **Order effect.** Does changing only the order of the same benign tasks systematically alter terminal safety? **Local-to-global gap.** If local safety effects can be measured accurately, are they sufficient for long-horizon task-order decisions? **Path-value predictability.** Do terminal path values contain learnable structure beyond local safety signals, and does improved path-value prediction translate into reliable end-to-end ordering?

5.1 Experimental Setup

Setup. Our main experiments use Qwen3-0.6B (Yang et al., 2025) with a rank-8 safety-anchor LoRA (Hu et al., 2022) and four benign downstream tasks: code, mathematics, question answering, and summarization. Each task is trained for 125 steps, and we evaluate all $4! = 24$ task orders while holding the remaining training settings fixed within each permutation tree. We use seed42 for development and seeds43–46 as whole-tree holdouts. Additional experiments with a modified task composition, Qwen3-1.7B (Yang et al., 2025), and SmolLM2-1.7B (Ben Allal et al., 2025) are used to examine the robustness of the order-effect phenomenon rather than the cross-setting generalization of P-DC-SCTR.

Implementation Details. We measure terminal safety by attack success rate (ASR) on 200 HarmBench prompts (Mazeika et al., 2024) using HarmBench-Llama-2-13B-cls. DC-SCTR characterizes

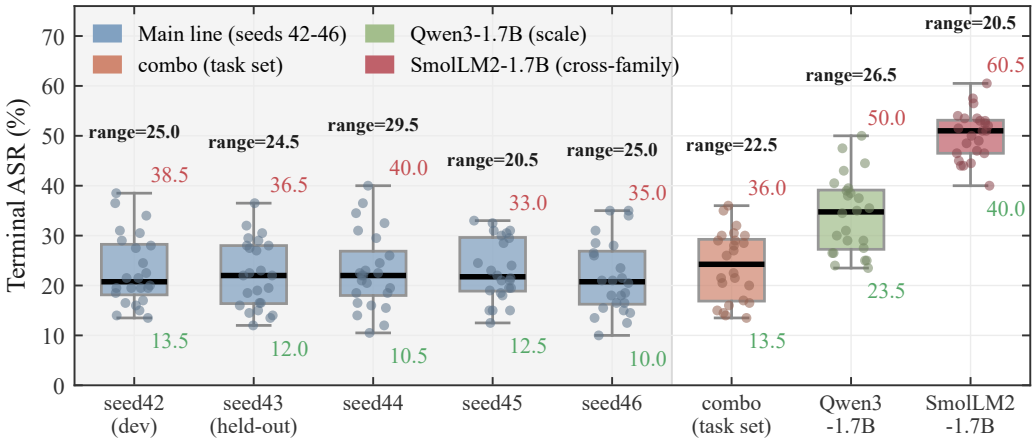


Figure 3: **Terminal safety varies substantially with benign task order.** Each distribution contains all 24 terminal permutations of one tree. Within each tree, only task order changes. Terminal-ASR ranges span 20.5–29.5 percentage points.

Table 1: Local information characterizes the immediate transition but provides limited guidance about terminal path value. The immediate-transition correlation uses 40 decision-relevant transitions from seed42; path-value correlations are reported as seed42 / seed43.

Signal	$\rho(\cdot, \Delta\text{ASR})$	$\rho(\cdot, Q^+)$	$\rho(\cdot, Q^-)$
True one-step ΔASR	self	$0.247 / 0.252$	$-0.128 / -0.082$
R_{adv}	0.833	$0.233 / 0.183$	$0.002 / -0.172$
R_{dir}	0.778	$-0.340 / -0.391$	$-0.017 / -0.148$

local transitions using fixed 20-step previews initialized from the current model state. Each preview follows the same training configuration as the corresponding full task update but is used only for measurement and does not modify the actual continual fine-tuning trajectory. From these previews, we obtain the local behavioral signals R_{adv} and R_{dir}. Decision-level analyses cover 40 transitions from 17 parent states. Full fine-tuning, safety evaluation, prompt-isolation, and preview protocols are provided in Appendix A.

5.2 Order-Dependent Terminal Safety

Changing only task order produces substantial variation in terminal safety. Across the eight evaluated trees, within-tree terminal-ASR ranges span 20.5–29.5 percentage points (Figure 3). For example, the 24 permutations of seed42 range from 13.5% to 38.5%. Because the task set, data, safety anchor, training configuration, and budget are fixed within each tree, task order is the only varying factor.

The observed variation is larger than expected from finite-prompt evaluation noise for all seven Qwen trees under a binomial-noise null ($p \leq 0.0005$). Evidence is weaker for SmolLM2, whose Bonferroni-corrected p-value is 0.086. Terminal rankings are also strongly correlated across the evaluated Qwen trees (pairwise Spearman $\rho = 0.778$–0.948), although no single extreme permutation transfers consistently across all settings. Full rankings, noise analyses, and uncertainty results are reported in Appendix C.

5.3 The Local-to-Global Safety Gap

Local transitions are measurable. On the 40 decision-relevant transitions in seed42, R_{adv} and R_{dir} correlate with realized one-step ΔASR at $\rho = 0.833$ and 0.778, respectively (Table 1). Thus, both signals capture substantial information about the immediate transition, suggesting that local measurement itself is not the primary bottleneck.

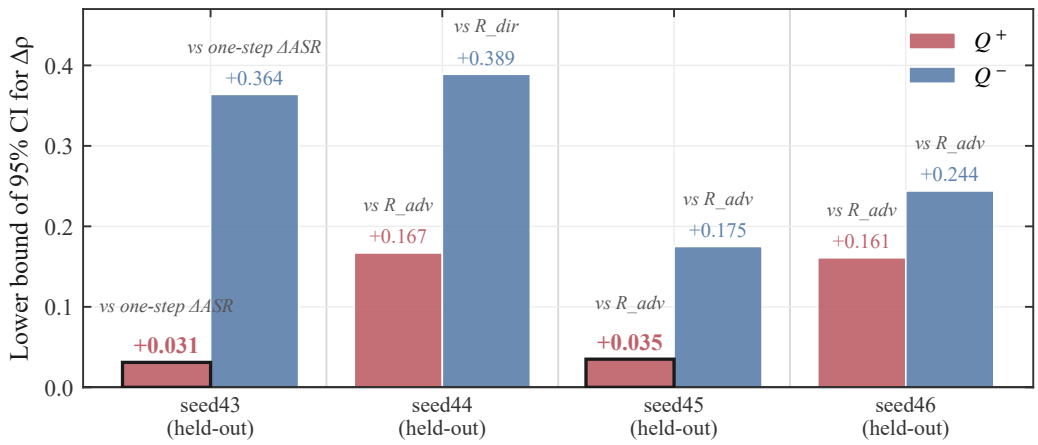


Figure 4: **Path-value prediction on whole-tree holdouts.** P-DC-SCTR is trained on seed42 and evaluated without refitting on seed43–46. Bars show the smallest nominal 95% bootstrap lower bound for the Spearman correlation gain over four local baselines. They quantify path-value prediction, not terminal-ASR improvement.

Local information provides limited guidance about terminal path value. The association weakens substantially when the target shifts from the immediate transition to terminal path value. R_{adv} correlates with Q^+ at only $\rho = 0.233$ on seed42 and 0.183 on seed43. More importantly, even realized one-step ΔASR correlates with Q^+ at only 0.247 and 0.252, and with Q^- at -0.128 and -0.082, respectively (Table 1). Thus, the limited correspondence with terminal path value cannot be explained solely by error in the local behavioral measurements.

The gap persists even with exact one-step information. We remove local prediction error entirely by retrospectively ranking candidates using their realized one-step ΔASR after full candidate training. By point estimate, this oracle local rule agrees with the corresponding terminal-extremal action at only 1/17 high-risk and 2/17 low-risk states on seed42, and 5/17 and 4/17 states on seed43. Rolling the same rule through the complete four-task trajectory remains 18–28 percentage points below the observed high-risk maxima and 14–17.5 points above the low-risk minima across seed42–46. The limitation therefore lies in the decision target, not only in its estimation: better prediction of the same one-step target does not close the local-to-global gap. Additional diagnostics reveal substantial state–task interactions, ambiguity among locally similar candidates, and a need for multi-step lookahead; details are reported in Appendix D.3.

5.4 Terminal Path-Value Estimation

Path values contain learnable structure. P-DC-SCTR is trained on seed42, with hyperparameters selected by leave-one-parent-out validation, and then frozen before evaluation on seed43–46. Across these whole-tree holdouts, Spearman correlation ranges from 0.684 to 0.780 for $\hat{Q}^+$ and from 0.691 to 0.763 for $\hat{Q}^-$. Parent-clustered bootstrap lower bounds for the correlation gains over all four evaluated local baselines are positive on every held-out tree and for both objectives; the narrowest margins are $+0.031$ and $+0.035$ (Figure 4).

Remaining-path context carries much of the predictive structure. Removing the remaining-task indicator reduces held-out Spearman correlation by 0.13–0.22, whereas removing R_{adv} and R_{dir} has little effect. This does not make local measurements uninformative about immediate transitions. Rather, it suggests that, in the evaluated setting, much of the predictive structure for terminal path value comes from remaining-path context rather than simple extrapolation of local-risk signals. Full feature-group ablations and initialization-sensitivity results are reported in Appendix E.2.

Path-value predictability does not imply path controllability. Despite improved path-value correlation, greedy rollouts based on the predicted path values do not show a consistent complete-order advantage over the whole-order transfer baseline on seed43–46. Under the low-risk objective, all paired per-prompt 95% confidence intervals for terminal-ASR differences include zero, while the high-risk results are mixed across trees. Thus, stronger path-value prediction does not by itself guarantee recovery of an extremal complete order. In this sense, path-value predictability is empirically separable from path controllability. Complete-order comparisons are reported in Appendix E.4. Training on multiple trees further improves path-value correlations, but still does not yield a consistent complete-order advantage; full results are reported in Appendix E.3. An exploratory multi-fidelity analysis is provided in Appendix E.5.

5.5 Robustness and Scope

The main qualitative findings persist on an independent safety-prompt set: terminal safety remains order dependent, and path-value estimates retain meaningful agreement with the independently evaluated targets. Cross-judge evaluation further shows partial agreement in terminal rankings but substantial disagreement in absolute ASR, indicating that fine-grained risk estimates remain evaluator dependent. Capability evaluation also shows that lower terminal ASR does not universally imply capability preservation, so safety-oriented ordering should not be interpreted as optimizing overall model quality. Finally, additional task and model settings provide complementary evidence for the order-effect phenomenon, but they do not establish method-level cross-task or cross-model generalization of P-DC-SCTR. Full prompt, judge, capability, and cross-setting analyses are reported in Appendix F.

6 Discussion and Limitations

Our results distinguish three aspects of safety under continual adaptation: local transition measurement, terminal path-value prediction, and path control. Accurate state-conditioned measurements can characterize immediate safety changes without reliably identifying terminal path values, while improved path-value prediction does not consistently translate into better complete task orders. Together with the dominant role of remaining-task context in our ablations, these findings suggest that safety under continual fine-tuning is fundamentally trajectory dependent: assessing an update in isolation is insufficient when that update also changes the state from which subsequent adaptation proceeds.

Several limitations bound these conclusions. Method-level evaluation is restricted to four-task permutation trees with Qwen3-0.6B, and the additional task and model settings support the order-effect phenomenon rather than broad generalization of P-DC-SCTR. Absolute ASR is evaluator dependent, and lower ASR does not universally imply capability preservation. Longer horizons and larger task sets may exhibit path structures not captured by the current setting. P-DC-SCTR should therefore be viewed as a path-aware terminal value estimator rather than a validated task-order optimizer, and the high-risk objective should be interpreted only as a controlled stress test.

7 Conclusion

We study safety under continual LLM fine-tuning from a trajectory-level perspective. Our results show that changing only the order of the same benign downstream tasks can substantially alter terminal safety. We further identify a local-to-global safety gap: even realized one-step safety effects provide limited guidance about the terminal outcomes induced by the remaining adaptation path. This indicates that the challenge cannot be reduced to improving local safety measurement alone, because each task update also changes the model state from which subsequent adaptation proceeds. Motivated by this gap, we formulate task ordering as a terminal path-value estimation problem and introduce P-DC-SCTR to jointly model the current state, local transition, and remaining-task context. Experiments show that terminal path values contain learnable structure beyond local safety signals, although improved path-value prediction does not consistently translate into more favorable complete task orders. This distinction between path-value predictability and path controllability suggests that reliable continual safety requires reasoning over the full adaptation trajectory rather than isolated updates. Developing methods that translate path-aware estimates into reliable long-horizon control remains an important direction for future work.

AI USE STATEMENT

Generative AI tools were used solely for language editing and polishing. All scientific content, experimental results, analyses, and conclusions were developed and verified by the authors.

REFERENCES

Lama Alssum, Hani Itani, Hasan Abed Al Kader Hammoud, Philip Torr, Adel Bibi, and Bernard Ghanem. Unforgotten safety: Preserving safety alignment of large language models with continual learning. *arXiv preprint arXiv:2512.10150*, 2025.

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

Thong Bach, Dung Nguyen, Thao Minh Le, and Truyen Tran. Continual safety alignment via gradient-based sample selection. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18870–18887. Association for Computational Linguistics, 2026.

Samuel J. Bell and Neil D. Lawrence. The effect of task ordering in continual learning. *arXiv preprint arXiv:2205.13323*, 2022.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025.

Sahil Chaudhary. Code alpaca: An instruction-following LLaMA model for code generation. GitHub repository, 2023. Code Alpaca dataset.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. 2021.

Qihong Chen, Jiawei Li, Hyunjae Suh, Lianghao Jiang, Zheng Zhou, Jingze Chen, Jiri Gesi, and Iftekhar Ahmed. Does the order of fine-tuning matter and why? *arXiv preprint arXiv:2410.02915*, 2024.

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

Zihan Guan, Mengxuan Hu, Ronghang Zhu, Sheng Li, and Anil Vullikanti. Benign samples matter! fine-tuning on outlier benign samples severely breaks safety. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20572–20597. PMLR, 2025.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe LoRA: The silver lining of reducing safety risks when fine-tuning large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *International Conference on Learning Representations*, 2025.

Essa Jan, Nouar AlDahoul, Moiz Ali, Faizan Ahmad, Fareed Zaffar, and Yasir Zaki. Multitask-bench: Unveiling and mitigating safety gaps in LLMs fine-tuning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 9025–9043. Association for Computational Linguistics, 2025.

Ziyan Li and Naoki Hiratani. Optimal task order for continual learning of multiple tasks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34578–34603. PMLR, 2025.

Ning Lu, Shengcai Liu, Jiahao Wu, Weiyu Chen, Zhirui Zhang, Yew-Soon Ong, Qi Wang, and Ke Tang. Safe delta: Consistently preserving safety when fine-tuning LLMs on diverse datasets. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 40537–40559. PMLR, 2025.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 2024.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400. Association for Computational Linguistics, 2024.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083. Association for Computational Linguistics, 2017.

Megan Ung, Alicia Sun, Samuel J. Bell, Bhaktipriya Radharapu, Levent Sagun, and Adina Williams. Chained tuning leads to biased forgetting. *arXiv preprint arXiv:2412.16469*, 2024.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. InsCL: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 663–677. Association for Computational Linguistics, 2024a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

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

APPENDIX

The appendix is organized as follows:

- **Appendix A** provides the complete experimental protocol, including model initialization, downstream training data, continual fine-tuning settings, safety and behavioral measurements, state-conditioned previews, path-value supervision, and capability evaluation.
- **Appendix B** describes the datasets used for safety, over-refusal, and capability evaluation.
- **Appendix C** provides additional analysis of order-dependent terminal safety, including complete results across task trees, cross-tree ranking structure, and the statistical resolution of terminal differences.
- **Appendix D** provides additional evidence for the local-to-global safety gap, including local measurements over all transitions, complete oracle-local trajectories, state–task interactions, and limited-horizon diagnostics.
- **Appendix E** provides additional analyses of terminal path-value estimation, including held-out prediction results, feature and initialization ablations, multi-tree training, complete-order selection, and the exploratory multi-fidelity extension.
- **Appendix F** examines the robustness and scope of the findings under independent safety prompts, alternative automatic judges, capability and over-refusal evaluation, and additional task and model settings.

A EXPERIMENTAL PROTOCOL

This appendix provides the complete experimental protocol used throughout the paper, including model initialization, downstream training data, permutation-tree construction, safety and behavioral measurements, state-conditioned previews, path-value supervision, and capability evaluation.

A.1 MODELS, TRAINING DATA, AND CONTINUAL FINE-TUNING

Base model and safety-anchor initialization. The main experiments use Qwen3-0.6B (Yang et al., 2025). All continual fine-tuning runs start from the same safety-aligned model. We obtain this initialization by training a rank-8 LoRA (Hu et al., 2022) on a safety-anchor dataset constructed from the training split of WildGuardMix (Han et al., 2024). The anchor contains 1,800 training examples and 150 validation examples. The training set consists of 900 harmful prompt–refusal pairs and 900 benign prompt–compliant pairs. The harmful examples are stratified by prompt subtype, with at most 40 examples sharing the same 30-character refusal-template prefix. The benign half contains 450 adversarially styled but benign prompts and 450 ordinary benign prompts. The anchor data are constructed with seed 42.

The resulting safety anchor has 2.5% ASR on the primary safety evaluation and 9.6% over-refusal on XSTest (Röttger et al., 2024). The same safety-anchor initialization is used for every task order within the main experiments.

Benign downstream tasks. We continually fine-tune on four benign downstream tasks, using 2,000 training examples per task sampled with seed 42: mathematics from GSM8K (Cobbe et al., 2021), code generation from Code Alpaca (Chaudhary, 2023), question answering from the `auxiliary_train` split of MMLU (Hendrycks et al., 2021), and summarization from CNN/DailyMail 3.0.0 (Hermann et al., 2015; See et al., 2017). All datasets are converted to a common instruction–output format.

Each task is fine-tuned for one epoch with a learning rate of 1×10^{-4}, batch size 4, gradient accumulation of 4, and a maximum sequence length of 1024. This corresponds to exactly 125 optimization steps per task. The optimizer is reinitialized before each task update. Within each permutation tree, the task set, sampled training data, per-task training budget, safety anchor, and optimization configuration are fixed; only task order varies. Training and evaluating one complete tree requires approximately 16 GPU hours.

Permutation trees and data splits. The four tasks yield $4! = 24$ complete orders. Representing a state by the ordered prefix of completed tasks gives 4, 12, 24, and 24 non-empty states at successive depths, for 64 non-root states and 24 terminal states. Including the root safety anchor gives 65 states in total. The root and the first two non-empty depths contain $1 + 4 + 12 = 17$ parent states with at least two available actions, yielding $4 + 12 + 24 = 40$ decision-relevant edges. The final 24 edges have only one available action and are therefore excluded from action-selection analyses.

Seed42 is used for development, path-value model fitting, and hyperparameter selection. Seeds43–46 are evaluated as whole-tree holdouts under the frozen protocol. Additional phenomenon-level experiments use a modified task composition that replaces question answering with an instruction task, Qwen3-1.7B (Yang et al., 2025), and SmolLM2-1.7B (Ben Allal et al., 2025). These additional settings are used only to test the robustness of the order-effect phenomenon and do not constitute cross-task, cross-scale, or cross-family evaluation of P-DC-SCTR.

A.2 Safety and Behavioral Measurements

Primary safety evaluation. The primary safety metric is attack success rate (ASR) on 200 standard HarmBench prompts (Mazeika et al., 2024), judged by HarmBench-Llama-2-13B-cls. Each prompt therefore corresponds to 0.5 percentage points of ASR resolution. This evaluation is used to measure state-level and terminal safety throughout the permutation trees.

Local behavioral channels. Local transition measurements use two isolated 128-prompt behavioral channels drawn from the WildGuardMix training split (Han et al., 2024). The adversarial channel measures the change in attack success behavior under adversarially formulated prompts and defines R_{adv}. The direct-query channel uses ordinary harmful queries and defines R_{dir} after aligning its sign with the risk direction. These channels are used for local transition measurement and P-DC-SCTR features, but not for the primary terminal ASR evaluation.

Prompt isolation. We audit separation among the safety-anchor data, behavioral probes, and primary safety-evaluation prompts using SHA-256 hashes, exact text matching, and 60-character prefix matching. The safety-anchor audit also covers XSTest (Röttger et al., 2024) and HarmBench (Mazeika et al., 2024); no overlap is detected under these checks. An independent safety evaluation additionally uses 200 PKU-derived harmful prompts with new templates and no overlap with the primary evaluation set. The independent evaluation is used only as a robustness check and does not replace HarmBench as the primary terminal-safety metric.

A.3 State-Conditioned Preview and Path-Value Protocol

State-conditioned previews. For each state–candidate pair (s, a), DC-SCTR uses a fixed $K = 20$-step preview initialized from the current model state. The preview follows the same training configuration as the corresponding 125-step full task update but stops after 20 optimization steps. Preview models are used only to measure local transitions and do not enter or modify the actual continual fine-tuning trajectory.

Starting every preview from the current state is essential because the same candidate task can induce different behavioral changes after different training prefixes. The resulting R_{adv} and R_{dir} measurements are therefore state-conditioned transition signals rather than fixed task-level attributes.

The fixed $K = 20$ preview used to construct P-DC-SCTR features is distinct from the exploratory $K \in \{20, 50, 125\}$ multi-fidelity analysis reported in Appendix E. The latter introduces additional online computation and is not part of the frozen P-DC-SCTR evaluation.

Path-value supervision. Exact Q^+ and Q^- supervision on the development tree is constructed from the exhaustive terminal outcomes of its 24 complete task orders, following the definitions in Section 3.3. For each state–candidate pair, these labels summarize the terminal-extremal continuations available after taking that candidate action. Descendant states and terminal outcomes are used only to construct supervision on the development tree. They are not available to P-DC-SCTR when evaluating the held-out trees.

Seed42 is the only tree used for model fitting and hyperparameter selection. After development, the P-DC-SCTR protocol is frozen and evaluated on seeds43–46. This separation ensures that the held-out path-value results measure transfer across independently trained permutation trees rather than in-tree fitting.

A.4 Capability and Over-Refusal Evaluation

We evaluate general capability using GSM8K, MMLU-Pro, HumanEval, MBPP, and CNN/DailyMail, and measure over-refusal using XSTest. The evaluation uses 300 GSM8K examples, 298 MMLU-Pro examples, all 164 HumanEval problems, 150 MBPP examples, and 100 CNN/DailyMail examples. We report accuracy for GSM8K and MMLU-Pro, pass@1 for HumanEval and MBPP, and ROUGE-L for CNN/DailyMail. Dataset descriptions and citations are provided in Appendix B.

Capability and over-refusal results are used only as diagnostics of the selected endpoints and are not objectives for fitting P-DC-SCTR.

B Benchmark Datasets

This appendix summarizes the datasets used for safety, over-refusal, and capability evaluation. The datasets used for continual fine-tuning are described separately in Appendix A.1.

- **HarmBench** (Mazeika et al., 2024) is a standardized benchmark for evaluating harmful compliance and robust refusal. We use 200 HarmBench prompts as the primary safety evaluation and report attack success rate (ASR) using HarmBench-Llama-2-13B-cls.
- **WildGuardMix** (Han et al., 2024) contains benign and harmful interactions with safety-related annotations. We use isolated subsets from its training split to construct the safety anchor and the behavioral channels used by DC-SCTR.
- **XSTest** (Röttger et al., 2024) evaluates exaggerated safety behavior by testing whether models unnecessarily refuse benign prompts that may appear superficially unsafe. We use it to measure over-refusal.
- **PKU-derived safety set** is an independently constructed evaluation set derived from a PKU safety dataset. We use 200 harmful prompts with new templates and no overlap with the primary HarmBench evaluation as an independent robustness check.
- **GSM8K** (Cobbe et al., 2021) is a grade-school mathematical reasoning benchmark. We use 300 examples for capability evaluation, with the evaluation data kept disjoint from the GSM8K examples used for mathematics fine-tuning.
- **MMLU-Pro** (Wang et al., 2024b) evaluates knowledge and reasoning across diverse academic and professional domains. We use 298 examples for capability evaluation. MMLU-Pro is distinct from the MMLU `auxiliary_train` split used for question-answering fine-tuning.
- **HumanEval** (Chen et al., 2021) contains 164 Python function-synthesis problems with executable tests. We evaluate all 164 problems and report pass@1.
- **MBPP** (Austin et al., 2021) contains natural-language programming problems paired with test cases. We evaluate 150 examples and report pass@1.
- **CNN/DailyMail** (Hermann et al., 2015; See et al., 2017) is a large-scale news summarization dataset. We use 100 examples for capability evaluation and report ROUGE-L, with the evaluation data kept separate from the examples used for summarization fine-tuning.

C Additional Analysis of Order Effects

C.1 Complete Order-Effect Results

Table 2 summarizes the 24 complete task orders in each of the eight evaluated trees. Absolute ASR values should not be used to compare model quality across different base models; the relevant quantity

Table 2: Terminal ASR summary for the eight task trees (24 terminal orders per tree).

Task tree	Min. ASR (%)	Max. ASR (%)	Range (pp)	Mean (%)
seed42 (development)	13.5	38.5	25.0	23.06
seed43 (holdout)	12.0	36.5	24.5	22.19
seed44	10.5	40.0	29.5	22.88
seed45	12.5	33.0	20.5	23.12
seed46	10.0	35.0	25.0	21.67
combo (qa→instruct)	13.5	36.0	22.5	23.90
Qwen3-1.7B	23.5	50.0	26.5	34.17
SmolLM2-1.7B	40.0	60.5	20.5	50.19

here is the within-tree variation induced by changing task order while holding the remaining training configuration fixed.

To assess whether the observed ranges can be explained by finite-prompt evaluation noise, we simulate 24 terminal estimates under a common $\mathrm{Binomial}(200, \hat{p})$ null. Across the seven Qwen trees, the observed ranges exceed this reference distribution ($p \leq 0.0005$ before correction). SmolLM2 provides weaker evidence: its unadjusted p-value is 0.0108 and its Bonferroni-corrected value across eight trees is 0.086.

For seed42–46, where per-prompt labels are available, we additionally use a prompt-paired permutation test that preserves each prompt's success count while shuffling terminal identity. All five trees have $p < 10^{-4}$. A moment-based variance decomposition estimates order-attributable standard deviations of 5.5–7.2 percentage points for the Qwen trees, compared with 2.9–3.3 points of single-endpoint sampling noise. For SmolLM2, the corresponding values are 3.4 and 3.5 points. We therefore interpret the cross-family evidence more cautiously.

C.2 Cross-Tree Ranking Structure

The order effect exhibits substantial but incomplete ranking structure across trees. Among the six Qwen trees, pairwise Spearman correlations of terminal-order rankings range from 0.778 to 0.948; within the five Qwen3-0.6B main-line trees, they range from 0.850 to 0.948. Correlations between the Qwen trees and SmolLM2 are substantially lower, ranging from 0.502 to 0.569.

The highest-risk seed42 order, `summarization_math_code_qa`, ranks first on seed44–46 and Qwen3-1.7B but eighth on seed43. The lowest-risk seed42 order, `qa_code_math_summarization`, ranks 24th, 24th, 23rd, and 22nd on seed43–46, respectively, where higher rank denotes lower ASR. Thus, complete-order rankings contain transferable structure, but an extreme order on the development tree is not guaranteed to remain extreme on every held-out tree.

C.3 Resolution of Terminal Differences

Terminal ASR is estimated from 200 binary judgments, giving a resolution of 0.5 percentage points. Across the 46 adjacent-terminal comparisons in seed42 and seed43, every prompt-paired bootstrap 95% interval includes zero, and the smallest exact McNemar p-value is 0.272. Fine distinctions between neighboring terminal ranks should therefore be interpreted as point-estimate ordering rather than statistically resolved differences.

This limitation does not imply that all state transitions are unresolved. Among 120 parent–child transitions in seed42 and seed43 for which paired labels are available, 71 (59%) have paired one-step $\Delta\mathrm{ASR}$ intervals excluding zero. Thus, many individual transitions are measurable even though small differences between neighboring terminal endpoints fall below the resolution of the 200-prompt evaluation.

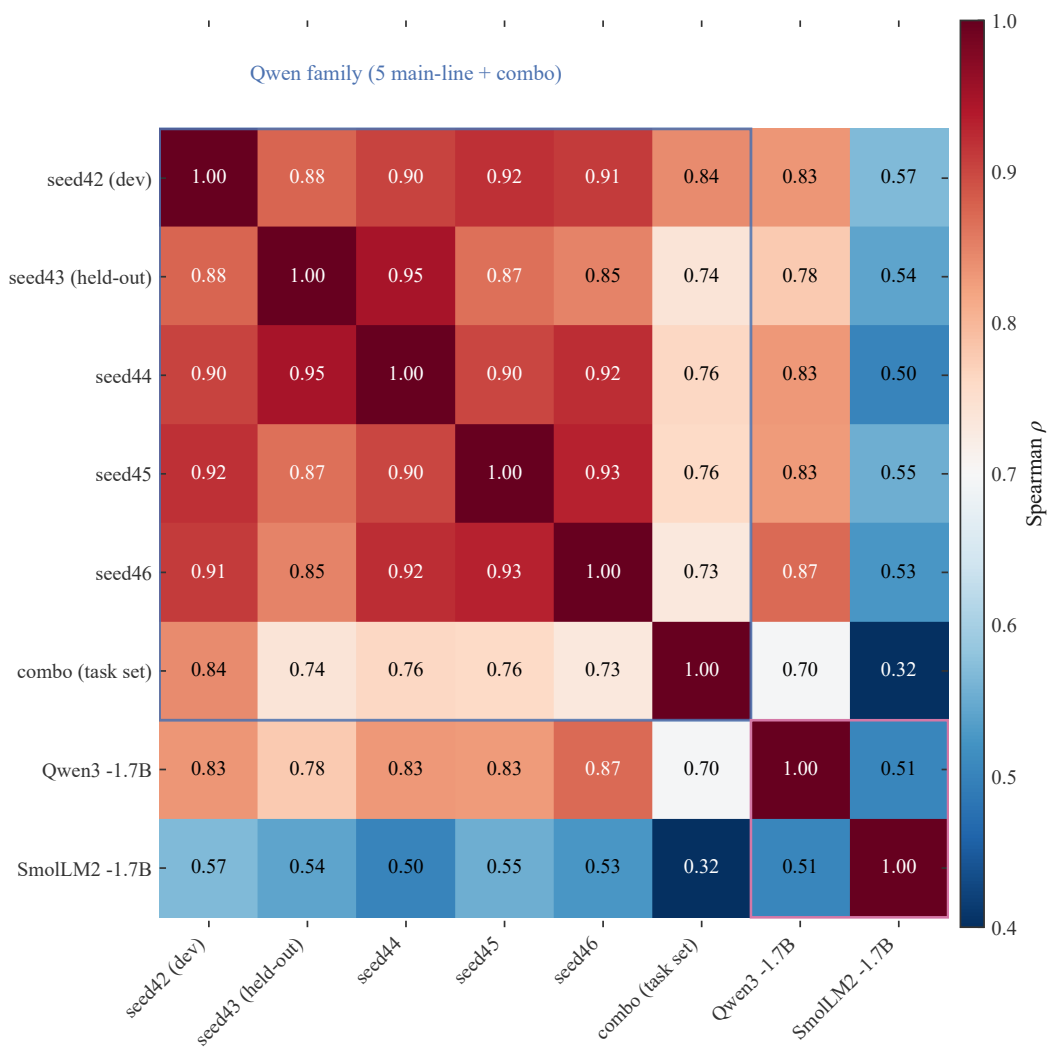


Figure 5: **Pairwise Spearman correlations of terminal-order rankings across the eight trees.** The five Qwen3-0.6B main-line trees exhibit strong ranking agreement. Agreement is weaker across model families. The combo tree replaces `qa` with `instruct` and is role-key aligned before computing the corresponding correlations.

D Additional Analysis of the Local-to-Global Gap

D.1 Local Measurement over All Transitions

The main text reports local-transition results on the 40 decision-relevant edges. Over all 64 transitions in seed42, including the 24 forced final-step transitions, R_{adv} and R_{dir} correlate with realized one-step ΔASR at Spearman $\rho = 0.812$ and 0.797, respectively. These values are close to the decision-edge correlations of 0.833 and 0.778, showing that the local measurement result is not specific to the decision-edge subset.

Importantly, this result concerns the accuracy of local transition measurement only. As shown in Section 5.3, accurate one-step information does not reliably identify terminal path values.

D.2 Complete Oracle-Local Trajectories

The retrospective one-step oracle in the main text selects actions using realized one-step $\Delta\mathrm{ASR}$ after full candidate-task training. It therefore removes proxy-prediction error while retaining the one-step decision target.

Across seed42–46, greedy maximization of realized one-step $\Delta\mathrm{ASR}$ produces the same tie-free complete order,

$$\texttt{code} \rightarrow \texttt{qa} \rightarrow \texttt{math} \rightarrow \texttt{summarization}.$$

Its terminal ASR ranges from 12.0% to 15.0% across the five trees, placing it at ranks 21–23 of 24 when orders are ranked from higher to lower ASR. These endpoints remain 18–28 percentage points below the observed high-risk maxima.

Conversely, greedy minimization of realized one-step $\Delta\mathrm{ASR}$ produces

$$\texttt{qa} \rightarrow \texttt{summarization} \rightarrow \texttt{math} \rightarrow \texttt{code}.$$

Its terminal ASR ranges from 24.5% to 30.0%, corresponding to ranks 6–8 of 24 under the same ranking convention. These endpoints remain 14–17.5 percentage points above the observed low-risk minima.

The failure therefore persists even when the local signal is measured using the same primary ASR metric as the terminal evaluation. The issue is not simply inaccurate estimation of the next transition; it is the mismatch between one-step selection and the terminal value induced by the remaining adaptation path.

D.3 State–Task Interactions and Limited-Horizon Diagnostics

State–task interaction analysis provides additional evidence that task effects depend on the preceding training path. For seed42 and seed43, the median absolute second-order parent-state-by-next-task interactions are 6.0 and 6.5 percentage points, respectively, compared with median main effects of 9.0 and 9.25 points. Interactions larger than 5 percentage points occur in 55.0% and 63.3% of the evaluated cases.

Local measurements also frequently leave multiple candidate actions difficult to distinguish. Under the frozen uncertainty band, candidates fall within one uncertainty region at 15/17 and 14/17 seed42 parent states for R_{adv} and R_{dir}, respectively, and at 12/17 states for both channels on seed43.

Finally, perfect-information limited-horizon comparisons show that a single step is insufficient to recover the relevant terminal-extremal continuation. For the high-risk stress-test objective, at least three steps of lookahead are required to enter the best 20% of terminal orders on both seed42 and seed43. For the low-risk safety-oriented objective, the corresponding horizons are two and three steps. These analyses are diagnostic and cover only seed42 and seed43.

E Additional Path-Value Analyses

E.1 Held-Out Path-Value Prediction

We use the frozen estimator and training protocol described in Section 4.2. Hyperparameters are selected on seed42 by leave-one-parent-out validation, after which the final estimator is fitted on the 64 seed42 edges and frozen. No held-out descendant states or terminal outcomes are used for fitting or model selection.

On seed43–46, Spearman correlations between predicted and exact terminal path values are 0.684–0.780 for Q^{+} and 0.691–0.763 for Q^{-}. Parent-clustered bootstrap comparisons against the four evaluated local baselines have positive nominal 95% lower bounds on every tree and for both objectives. The narrowest high-risk margins are $+0.031$ on seed43 against realized one-step $\Delta\mathrm{ASR}$ and $+0.035$ on seed45 against R_{adv}.

Table 3 reports both edge-level prediction quality and the complete-order outcomes obtained by greedily applying the corresponding point estimates. The two quantities are reported together to make explicit that path-value prediction and sequential path control are different evaluation targets.

Table 3: Path-value prediction and complete-order outcomes on the four held-out trees. Higher edge-level correlation does not consistently translate into a more favorable terminal order.

Holdout	$\rho(\hat{Q}^+, Q^+)$	$\rho(\hat{Q}^-, Q^-)$	Optimal-action agreement, high/low	Greedy rollout of $\hat{Q}$ ASR, high/low (%)	`copy-dev` ASR, high/low (%)
seed43	0.684	0.721	$10/17,\ 10/17$	27.5 / 15.0	27.5 / 12.0
seed44	0.758	0.759	$14/17,\ 11/17$	31.0 / 14.0	40.0 / 10.5
seed45	0.714	0.691	$11/17,\ 13/17$	30.5 / 12.5	33.0 / 15.0
seed46	0.780	0.763	$13/17,\ 13/17$	35.0 / 10.0	35.0 / 13.5

Table 4: Feature-group ablation. Entries are high-/low-risk Spearman correlations after removing the indicated feature group.

Removed feature group	seed43 high/low	seed44 high/low
None (full 16 features)	$0.684/0.721$	$0.758/0.759$
Parent-state behavioral statistics	$0.646/0.715$	$0.743/0.761$
$R_{\mathrm{adv}}, R_{\mathrm{dir}}$	$0.673/0.730$	$0.740/0.777$
Behavioral-transition statistics $B/Q/C$	$0.698/0.734$	$0.756/0.756$
Depth	$0.666/0.707$	$0.734/0.771$
Candidate-task one-hot	$0.683/0.729$	$0.740/0.779$
Remaining-task indicators	$0.549/0.590$	$0.543/0.587$

Relative to `copy-dev`, the terminal-ASR point estimates are mixed across trees and objectives. On the low-risk objective, the differences between the greedy $\hat{Q}$ rollout and `copy-dev` are $+3.0$, $+3.5$, -2.5, and -3.5 percentage points on seed43–46, respectively. On the high-risk objective, the two procedures have identical terminal point estimates on seed43 and seed46, while the greedy $\hat{Q}$ rollout has lower ASR on seed44 and seed45. These complete-order outcomes should not be interpreted as following directly from the edge-level correlation gains.

E.2 Feature and Initialization Ablations

Across five torch initializations (42, 7, 13, 100, and 202), the standard deviation of the four seed43/44 path-value correlations is at most 0.015. Complete-order endpoints are also nearly invariant; the only observed exception is torch seed 7, which selects a sibling high-risk seed43 endpoint at 28.0% rather than 27.5%, without changing the qualitative conclusion.

Table 4 reports feature-group ablations on seed43 and seed44.

The largest degradation occurs when the remaining-task indicators are removed, reducing correlation by approximately 0.13–0.22 across the two trees and objectives. In contrast, removing the aggregate $R_{\mathrm{adv}}/R_{\mathrm{dir}}$ features has little effect in these settings. This suggests that much of the predictive structure in the evaluated setting is associated with remaining-path context rather than simple extrapolation of aggregate local-risk signals. This does not imply that all local transition information is unnecessary: the full representation also contains parent-state and behavioral-transition information.

E.3 Multi-Tree Training

We additionally examine whether increasing the path-value training pool changes the held-out results. In this analysis, one tree is held out while the estimator is fitted on 256 edges from the other four same-task trees. Hyperparameters and the random seed remain fixed, feature standardization uses only the training pool, and parent states are namespaced by tree. Because seed42 participated in the original hyperparameter selection, its rotated result is descriptive; independent held-out comparisons use seed43–46.

Compared with frozen single-tree training, high-risk correlations increase on all four method holdouts and low-risk correlations increase on three, remaining unchanged on seed44. The two narrow high-

Table 5: Leave-one-tree-out training: held-out path-value correlations and complete-order outcomes.

Held-out tree	High/low corr.	Min. high-risk CI lower bound	Action agreement high/low	Greedy rollout ASR, high/low (%)
seed42 (descriptive)	0.814/0.683	+0.304	13/17, 11/17	38.5 / 15.0
seed43	0.772/0.735	+0.168	9/17, 10/17	27.5 / 15.0
seed44	0.794/0.759	+0.218	14/17, 11/17	40.0 / 14.0
seed45	0.822/0.763	+0.154	13/17, 14/17	33.0 / 12.5
seed46	0.811/0.780	+0.207	13/17, 12/17	35.0 / 10.0

Table 6: Paired low-risk terminal comparisons between the greedy $\hat{Q}$ rollout and `copy-dev`.

Holdout	Point diff. (pp)	Paired bootstrap 95% CI (pp)	Exact McNemar p
seed43	+3.0	$[-1.5, +7.5]$	0.286
seed44	+3.5	$[-1.0, +8.5]$	0.210
seed45	−2.5	$[-7.0, +2.0]$	0.405
seed46	−3.5	$[-8.5, +1.5]$	0.248

risk lower bounds from the single-tree experiment increase from $+0.031$ and $+0.035$ to $+0.168$ and $+0.154$, respectively. Optimal-action agreement does not improve systematically.

Across seed43–46, the multi-tree high-risk rollouts have the same terminal ASR as `copy-dev`, whereas the low-risk point estimates remain mixed. Increasing the training pool therefore strengthens path-value prediction without resolving complete-order control.

E.4 Prediction versus Complete-Order Selection

To quantify the uncertainty in the complete-order comparison, we evaluate the low-risk terminal-ASR difference between the greedy $\hat{Q}$ rollout and `copy-dev` on the same prompts. Table 6 reports P-DC-SCTR minus `copy-dev`. All four paired bootstrap 95% intervals include zero.

The compared low-risk endpoints are identical under the frozen and multi-tree training analyses, so the same paired terminal comparisons apply to both. The point-estimate directions differ across trees, and none of the four differences is statistically resolved. This provides a direct example of the distinction emphasized in the main text: improved path-value prediction does not by itself establish improved end-to-end task ordering.

E.5 Exploratory Multi-Fidelity Extension

We treat the following analysis as exploratory and separate it from the frozen P-DC-SCTR evaluation in the main text. The multi-fidelity extension first compares bootstrap intervals for candidate $\hat{Q}$ values. When the intervals overlap, it retains the top two candidates and performs additional two-step previews at $K = 20, 50, 125$. A branch is removed only when the behavioral-probe ASR difference between preview endpoints exceeds $\tau_A = 10.77$ percentage points; otherwise both branches are retained, with a maximum beam width of two.

"Optimistic" and "conservative" denote retrospective readings of unresolved retained branches under the corresponding objective. They are not separate algorithms, and favorable branches from different objectives cannot be combined into a single realized policy.

On seed43, the observed high-risk maximum is 36.5% ASR. The optimistic and conservative retained endpoints reach 36.5% and 32.0%, respectively, compared with 27.5% for both the greedy $\hat{Q}$ rollout and `copy-dev`. The 4.5-point difference between the retained endpoints is not statistically resolved.

For the low-risk objective, the observed minimum and `copy-dev` reach 12.0% ASR, compared with 15.0% for the greedy $\hat{Q}$ rollout and optimistic branch, and 23.0% for the conservative branch. These results show that online preview and branch retention can alter trajectory selection, but do not establish reliable complete-order control.

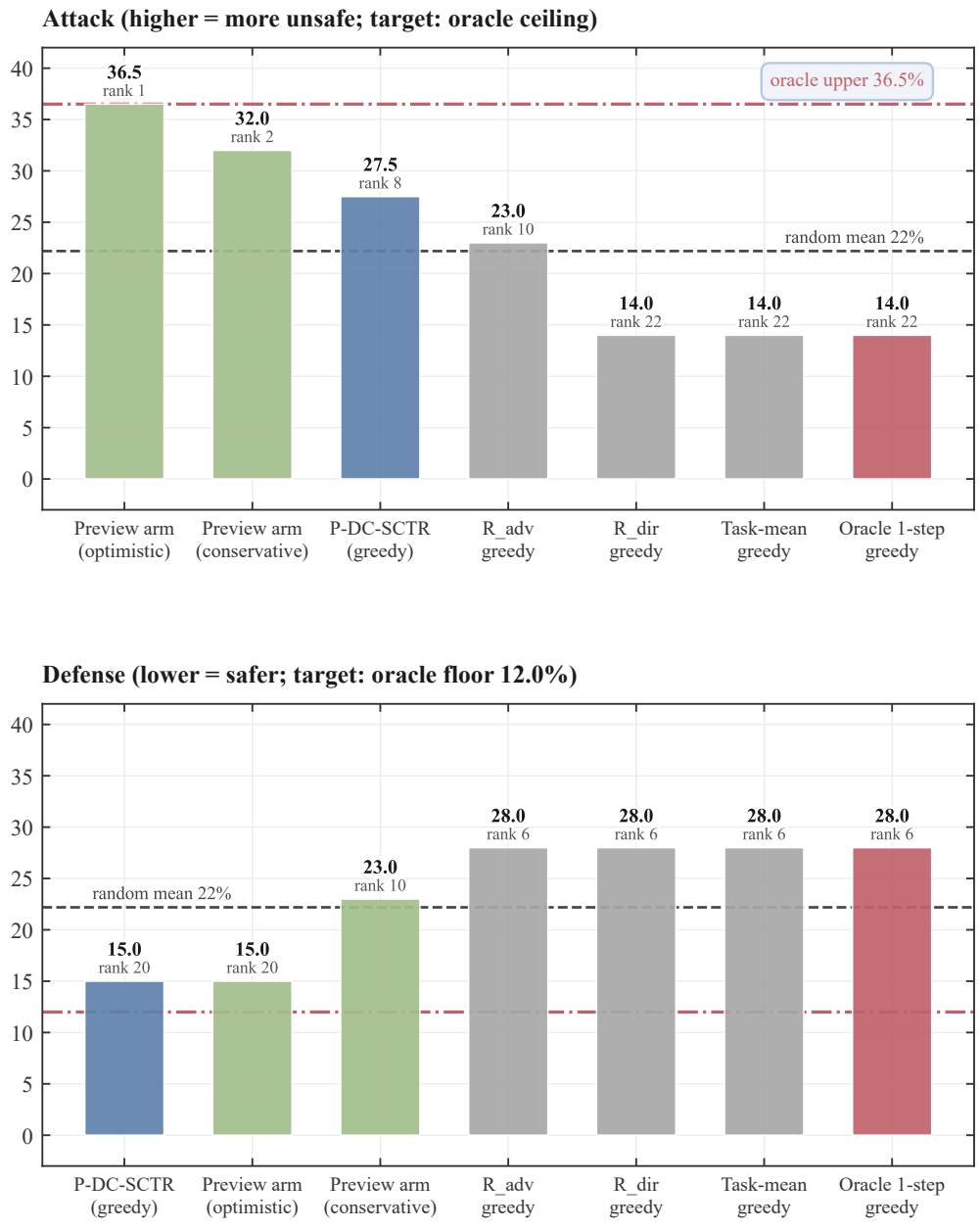
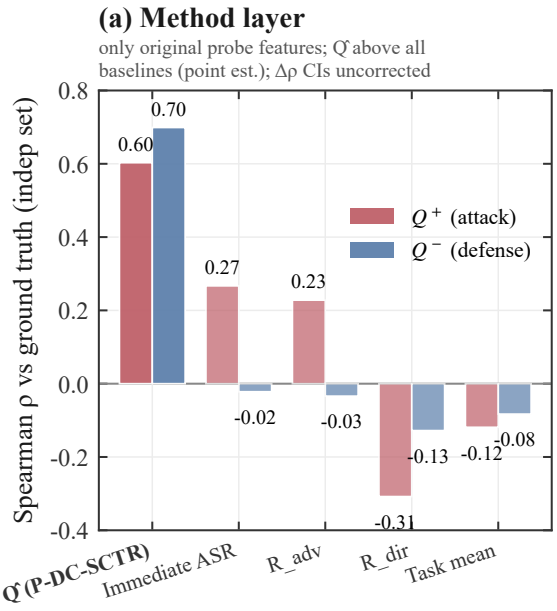


Figure 6: **Exploratory multi-fidelity selection on the seed43 holdout.** Additional online previews alter the retained high-risk trajectories, whereas low-risk outcomes remain mixed. Nearby terminal ranks should not be interpreted as statistically resolved; in particular, the 4.5-point difference between the two retained high-risk endpoints has exact McNemar $p = 0.272$. This experiment is separate from the frozen P-DC-SCTR evaluation.

F Robustness and Scope

This appendix examines the robustness and scope of the main findings. We evaluate sensitivity to the safety-prompt set and automatic judge, examine capability and over-refusal for representative endpoints, and finally clarify the scope of the resulting evidence.



(b) Order effect at 3 levels

(out-of-source PKU + new templates, zero overlap)
$\tau_{\text{indep}} = 6.50$ pp

Level	ρ	n
Terminals (24)	0.804	24
States (64)	0.825	64
Edges ΔASR (64)	0.838	64

ρ: primary- vs independent-set Spearman
64 edges incl. 24 terminal single-choice edges
(not the 40 planning-analysis edges)

Figure 7: **Robustness to an independent safety prompt set.** The independent evaluation uses 200 PKU-derived harmful prompts with new templates and no overlap with the primary safety set. The primary and independent evaluations retain substantial rank agreement across terminal models, intermediate states, and transitions. Predictions from the frozen P-DC-SCTR estimator also remain correlated with terminal path values recomputed under the independent evaluation.

F.1 Independent Safety Prompts

The primary experiments evaluate safety using 200 HarmBench prompts. To test whether the observed structure depends on this particular prompt set, we construct an independent evaluation containing 200 PKU-derived harmful prompts with new templates and no overlap with the primary set. We use this independent set to re-evaluate the common root and all 64 non-root prefix states.

The relative safety structure is largely preserved under the independent evaluation. Spearman correlations between the primary and independent measurements are

$$\rho_{\text{terminal}} = 0.804, \qquad \rho_{\text{state}} = 0.825, \qquad \rho_{\text{edge}} = 0.838,$$

computed over the 24 terminal orders, 64 non-root states, and 64 parent–child transitions, respectively. Thus, the order-dependent structure observed under the primary evaluation is not specific to a single safety prompt set.

We also evaluate whether the frozen path-value estimator retains predictive structure when the target path values are recomputed from the independent safety measurements. P-DC-SCTR is not retrained on the independent prompts. Its predictions obtain

$$\rho(\hat{Q}^+, Q^+_{\text{indep}}) = 0.603, \qquad \rho(\hat{Q}^-, Q^-_{\text{indep}}) = 0.699.$$

These correlations indicate that part of the learned terminal path-value structure persists when safety is measured with a different prompt set.

Under the independent evaluation, the complete order selected by the frozen low-risk rollout reaches 10.5% terminal ASR and ranks 22nd among the 24 orders when larger rank denotes lower ASR; the observed minimum is 8.5%. This result is consistent with the distinction made in the main text: path-value prediction can retain useful structure without reliably recovering the lowest-risk complete order.

This experiment changes the safety evaluation but not the underlying continual-training trajectories. It therefore provides evidence of *prompt-set robustness*, rather than an independent-tree, independent-task, or cross-model replication of P-DC-SCTR.

Table 7: Pairwise Cohen's κ between four model-based safety judgments on the stratified 149-response subset.

	Claude ref.	HarmBench	Qwen	Mistral
Claude ref.	1.000	0.464	0.338	0.400
HarmBench	0.464	1.000	0.103	0.219
Qwen	0.338	0.103	1.000	0.155
Mistral	0.400	0.219	0.155	1.000

Table 8: Capability of the selected low-risk endpoint relative to the mean over all 24 terminal task orders. Positive values indicate higher capability scores. XSTest reports the over-refusal rate of the selected endpoint.

Evaluation	Difference from 24-order mean
GSM8K	$+4.54$ pp
MMLU-Pro	-0.16 pp
HumanEval	$+3.73$ pp
MBPP	$+0.67$ pp
CNN/DailyMail ROUGE-L	$+1.58$ pp
XSTest over-refusal	0%

F.2 Alternative Automatic Judges

The primary experiments use HarmBench-Llama-2-13B-cls as the automatic safety evaluator. Because absolute ASR can depend on the evaluator, we compare it with Qwen2.5-7B, Mistral-7B-Instruct-v0.3, and a Claude reference model on a stratified subset of 149 responses. The Claude reference labels are model-generated and are not treated as human annotations.

The four evaluators produce substantially different positive rates: 25.5% for HarmBench, 12.1% for Qwen, 58.4% for Mistral, and 34.9% for the Claude reference. Pairwise Cohen's κ values are shown in Table 7.

Agreement among the non-reference automatic judges is limited. Over all 4,800 prompt-paired judgments from the 24 terminal models, HarmBench and Qwen have raw binary agreement of 0.761 but Cohen's $\kappa = 0.154$. HarmBench reports ASR values that are on average 13.4 percentage points higher than Qwen, with a 95% confidence interval of $[12.1, 14.8]$ percentage points. Nevertheless, the rankings of the 24 terminal models retain a positive association, with Spearman $\rho = 0.594$.

These results show that absolute safety levels are meaningfully evaluator-dependent. Some relative terminal-order structure is preserved, but the agreement is not strong enough to treat ASR as judge invariant. We therefore use this analysis to characterize measurement sensitivity, rather than as evidence for a judge-independent safety ground truth.

F.3 Capability and Over-Refusal

Lower terminal ASR does not necessarily imply that a complete training trajectory is preferable if the reduction is accompanied by capability degradation or excessive refusal. We therefore evaluate downstream capability and XSTest over-refusal for the low-risk endpoint produced by the frozen P-DC-SCTR rollout,

$$\texttt{code} \rightarrow \texttt{math} \rightarrow \texttt{qa} \rightarrow \texttt{summarization}.$$

Because all 24 terminal models are trained on the same four tasks with the same per-task budget, we report capability differences relative to the mean over all 24 terminal orders.

The selected endpoint does not exhibit broad degradation across the evaluated capability benchmarks: four of the five capability differences are positive, while MMLU-Pro differs from the 24-order mean by only -0.16 percentage points. Its XSTest over-refusal rate is 0% under the reported evaluation. These results characterize this particular endpoint; they do not establish a general capability benefit of low-ASR task orders or of P-DC-SCTR-based selection.

Indeed, capability preservation is not universal across low-risk trajectories. The alternative endpoint retained by the exploratory multi-fidelity extension,

$$\texttt{math} \rightarrow \texttt{code} \rightarrow \texttt{qa} \rightarrow \texttt{summarization},$$

reduces GSM8K by 6.46 percentage points relative to the 24-order mean. Thus, lower terminal ASR does not by itself imply capability preservation. Safety and capability should therefore be evaluated jointly when characterizing complete adaptation paths.

F.4 Scope of the Evidence

The robustness analyses support different levels of evidence and should not be interpreted as establishing the same form of generalization. Across additional seeds, a modified task composition, Qwen3-1.7B, and SmolLM2-1.7B, we observe substantial order-dependent variation in terminal safety. The evidence is strongest across the Qwen settings and weaker for SmolLM2 after multiplicity correction (Bonferroni-adjusted $p = 0.086$). These experiments therefore support phenomenon-level robustness of the order effect, with more limited cross-family evidence.

Method-level evidence is narrower. P-DC-SCTR is developed on the seed42 Qwen3-0.6B tree and evaluated without refitting on seed43–46, which retain the same model and four-task composition while varying training randomness. The additional task-composition and model settings are not used to establish equivalent cross-task or cross-model transfer of P-DC-SCTR. Likewise, the independent-prompt analysis changes the safety evaluation while keeping the underlying training trajectories fixed, and therefore tests prompt-set robustness rather than independent path-level replication.

Finally, the current results distinguish path-value prediction from path control. Held-out path values exhibit learnable structure, but improved prediction does not consistently yield more favorable complete task orders. The present evidence therefore supports P-DC-SCTR as a path-aware terminal value estimator, rather than a validated task-order optimizer. Reliable control over longer horizons, larger task sets, and unseen task compositions remains open.