SELECTING LONG-HORIZON TRAJECTORIES FOR RELIABLE AND EFFICIENT TERMINAL-AGENT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Terminal agents are commonly trained by imitating long teacher trajectories, yet how much of each trajectory to supervise remains unexplored. We study the *supervision horizon*, the number of trajectory tokens retained for training, and show that it is a key design axis for reliability and cost. Reliability improves with longer horizons but saturates: on Terminal-Bench, a 12K-token horizon solves more tasks than 16K (29 ± 0.7 vs. 26 ± 0.8) while requiring 30% less training time. The horizon also shapes agent behavior: short horizons cause premature termination, intermediate horizons yield productive error recovery, and long horizons induce over-persistence. We analyze this saturation through a bias–complexity bound, in which longer supervision reduces temporal supervision bias but increases finite-sample estimation error from more heterogeneous late-stage histories. Guided by this analysis, we propose *selective long-horizon refinement*, which first trains on short prefixes and then refines only on continuations that are most likely under the warm-start model. It consistently outperforms full long-horizon training. At 16K, it raises successful attempts from 110 ± 2.7 to 126 ± 2.1 and tasks solved in at least six of eight attempts from 9 ± 0.7 to 14 ± 0.6; with half of the long-horizon data, it still reaches 122 ± 2.4 while cutting training time by 23%. The gains transfer across benchmarks, from 64 ± 2.6 to 73 ± 2.1 on Terminal-Bench v2.0 and from 137 ± 2.7 to 155 ± 2.2 on OpenThoughts-TBLite. For long-horizon supervision, selecting the right trajectories matters more than training on all of them.

1 INTRODUCTION

Terminal-based coding agents such as OpenAI Codex (OpenAI, 2025) and Anthropic Claude Code (Anthropic, 2026) are becoming a common interface for software development, operating directly over terminals and repositories to execute commands, modify code, and iteratively solve tasks. Recent works have therefore begun to study how to train smaller, capable terminal agents efficiently (Pi et al., 2026; Cheng et al., 2026; Gandhi et al., 2026; Wu et al., 2026).

Table 1: **Training time and accuracy** across supervision horizons.

Cutoff	4K	8K	12K	16K
Time (h) ↓	**10.17**	21.65	35.44	50.81
Accuracy (%) ↑	24	25	**29**	26

A common way to train terminal agents is imitation learning from teacher-generated trajectories, where the student is supervised to reproduce the teacher's actions conditioned on the preceding interaction history. These long-horizon terminal trajectories involve extended sequences of interdependent decisions in which later actions depend on earlier observations, modifications, errors, and intermediate outcomes. As trajectories unfold, the agent may need to recover from failed commands, revise hypotheses, verify intermediate results, or switch strategies based on accumulated interaction history. Training with a short horizon can therefore omit these later-stage behaviors and provide insufficient supervision for recovery, correction, verification, and strategy switching. In contrast, longer horizons expose the student to richer temporal dependencies and may improve persistence and long-range reasoning, but they also incur substantially higher training cost and introduce increasingly heterogeneous histories and more complex decision situations. As shown in Table 1, increasing the cutoff from 8K to 16K more than *doubles training time*, from 21.65 to 50.81 hours, while Terminal-Bench accuracy improves only marginally from 25 to 26. Moreover, the 12K model achieves the highest accuracy of 29 with 35.44 hours of training, suggesting that simply extending the supervision horizon does not yield proportional performance gains and may

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.

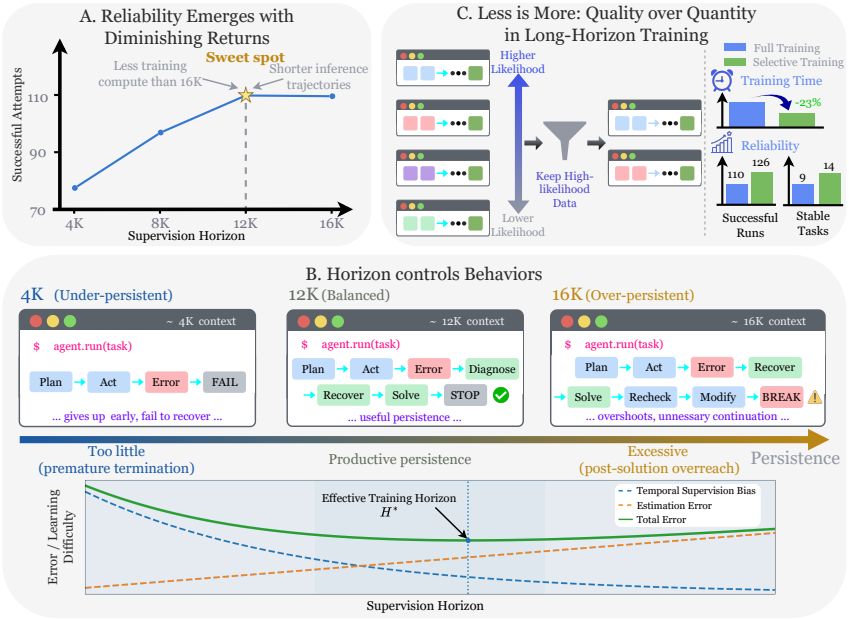


Figure 1: **Overview of our three main findings on effective training horizons for terminal agents.** **(A)** ***Reliability Improves with Diminishing Returns*:** increasing the training horizon substantially improves reliability, but the marginal gains diminish beyond an intermediate sweet spot, where 12K provides a favorable balance between successful attempts, training compute, and inference length. **(B)** ***Horizon Controls Behaviors*:** increasing the training horizon shifts agents from under-persistence at 4K, to productive persistence at 12K, and over-persistence at 16K; theoretically, this reflects a trade-off between decreasing temporal supervision bias and increasing estimation difficulty. **(C)** ***Less is More: Quality over Quantity in Long-Horizon Training*:** simply training on more long-horizon trajectories is not necessarily beneficial; training on a smaller set of appropriately selected long-horizon trajectories can achieve higher reliability than using the full long-horizon data, showing that data quality matters more than data quantity for exploiting long-horizon supervision.

even exhibit diminishing returns. This motivates **two important design questions** that remains largely unresolved:

Q1: *Is supervising the longer teacher trajectory always better?*

Q2: *Can selectively chosen long-horizon trajectories provide better and more efficient supervision than using all long-horizon data?*

To answer these questions, we study the *effective training horizon* for terminal agents: a horizon that is sufficiently long to capture useful temporal dependencies, yet not so long that additional supervision becomes inefficient or counterproductive. As summarized in Figure 1, **our study reveals three main findings.** ❶ *First*, increasing the training horizon substantially improves reliability, but the gains diminish beyond an intermediate sweet spot, indicating that full-horizon supervision is not necessarily optimal. ❷ *Second*, the training horizon systematically shapes the agent's behavioral regime: short horizons induce under-persistence, intermediate horizons encourage productive persistence, while excessively long horizons lead to over-persistence, causing agents to continue acting unnecessarily and resulting in longer test-time inference trajectories. We further explain this intermediate optimum theoretically by decomposing the total error into two competing quantities: *temporal supervision bias*, which decreases as longer horizons expose the model to richer later-stage behaviors, and *finite-sample estimation error*, which increases as the learner must model a more complex set of long-horizon histories. The effective training horizon therefore emerges at an intermediate point that balances the decreasing temporal supervision bias against the increasing estimation error. These findings lead to a declarative answer to **Q1**: **No**, supervising a longer teacher trajectory is not always better; the effective training horizon lies in the middle. ❸ *Third*, we show that more long-horizon data is not always better. A two-stage strategy that first trains on the full short-horizon data and then

refines the model using only half of the long-horizon trajectories achieves higher reliability than full long-horizon training while reducing both training and inference costs. This leads to a declarative answer to **Q2**: **Yes**, selectively chosen long-horizon trajectories can provide more effective and efficient supervision than using the full long-horizon dataset.

Our contributions are fourfold and organized as follows. ❶ *First*, we characterize the relationship between training horizon and agent reliability, showing that reliability improves with longer supervision but exhibits diminishing returns (Section 4). ❷ *Second*, we explain this scaling behavior empirically through the notion of persistence, revealing a transition from under-persistence to productive persistence and eventually over-persistence (Section 5.1). ❸ *Third*, we theoretically explain why longer supervision eventually exhibits diminishing returns, characterizing the effective training horizon through a trade-off between temporal supervision bias and finite-sample estimation error (Section 5.2). ❹ *Fourth*, we propose a simple, efficient two-stage training strategy that combines full short-horizon training with selective long-horizon refinement to further improve reliability (Section 6).

2 Related Works

Terminal-Agent Training and Data Engineering. Recent work improves terminal agents through data scaling, environment construction, and richer supervision. Pi et al. (2026) study long-context training and show that extending context does not always improve performance, while Gandhi et al. (2026), Cheng et al. (2026), and Shrivastava et al. (2026) improve environments and supervision signals. *In contrast, we isolate the supervision horizon while fixing the model and trajectory set, study its effect on repeated task success and agent behavior, and use this analysis to motivate selective refinement of newly exposed long-horizon trajectory segments.*

Likelihood-based data selection for SFT. Prior work selects fine-tuning data by its likelihood under the student, favoring responses that fit its distribution (Ren et al., 2024; Zhang et al., 2025), while Cuong et al. (2026) show that this preference depends on model capacity and training duration. These methods score whole responses; *we instead score only the newly exposed continuation* $(H_1, H_2]$ *of each long-horizon trajectory under a warm-start model, treating likelihood as compatibility with the learned short-horizon policy.* Further detailed related works are in Appendix A.

3 Background

3.1 Terminal Agents as Sequential Decision Making

In this work, we formulate the interaction following the *Terminus-2* scaffold (Merrill et al., 2026), which provides the model with a single headless terminal interface and requires it to complete tasks through Bash commands. This lightweight setup minimizes additional scaffold-specific mechanisms, allowing the learned sequential behavior to be more directly attributed to the underlying language model. At each interaction step t, the agent observes the current terminal output o_t and selects an action a_t, such as executing a command, inspecting or modifying a file, installing a dependency, or running a test. The action is sampled from a policy π_θ conditioned on the complete interaction history $h_t = (o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t)$, i.e., $a_t \sim \pi_\theta(\cdot \mid h_t)$. A complete interaction forms a trajectory $\tau = (o_1, a_1, o_2, a_2, \ldots, o_T, a_T)$, where T denotes the number of environment-level actions.

3.2 Terminal Agent Training as Imitation Learning

A common paradigm for training terminal agents is to learn from trajectories generated by a stronger teacher model, which can be viewed as *offline imitation learning*. Let π_{T} denote the teacher policy and π_θ the student policy. At each step t, the teacher selects an action according to $a_t \sim \pi_{\mathrm{T}}(\cdot \mid h_t)$, where $h_t = (o_1, a_1, \ldots, o_t)$ denotes the interaction history. Rolling out the teacher policy produces a trajectory $\tau = (o_1, a_1, o_2, a_2, \ldots, o_T, a_T)$. A dataset of N teacher trajectories is denoted by $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^{N}$, where $\tau^{(i)} = (o_1^{(i)}, a_1^{(i)}, \ldots, o_{T_i}^{(i)}, a_{T_i}^{(i)})$.

Given these offline demonstrations, behavior cloning trains the student to imitate the teacher action at each visited history. The student assigns probability $\pi_\theta(a_t^{(i)} \mid h_t^{(i)})$ to the teacher action observed at

step t, with the standard log-loss objective

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\log \pi_\theta\left(a_t^{(i)} \mid h_t^{(i)}\right). \tag{1}$$

3.3 From Decision Horizon to Token-Level Supervision

In practice, LLM-based agents are trained with supervised fine-tuning (SFT) over tokenized interaction sequences. In SFT, each action $a_t = (y_{t,1}, \ldots, y_{t,\ell_t})$ is generated autoregressively at the token level. Let $p_\theta(y_{t,j} \mid h_t, y_{t,<j})$ denote the next-token distribution. Then,

$$\pi_\theta(a_t \mid h_t) = \prod_{j=1}^{\ell_t} p_\theta(y_{t,j} \mid h_t, y_{t,<j}), \qquad \log \pi_\theta(a_t \mid h_t) = \sum_{j=1}^{\ell_t} \log p_\theta(y_{t,j} \mid h_t, y_{t,<j}). \tag{2}$$

A common cutoff at a particular decision step does not correspond to a common sequence length across trajectories. Instead, the supervision horizon is implemented as a token budget K, retaining only the prefix of the serialized trajectory within the first K tokens. Letting $n_t(\tau)$ denote the cumulative number of tokens through action a_t, a token budget K induces the decision horizon $H_K(\tau) = \max\{t : n_t(\tau) \le K\}$, which is non-decreasing in K. Throughout the empirical analysis, we therefore use the *token supervision horizon* K as the controllable training variable, while the theoretical analysis is expressed in terms of the induced *decision horizon* H.

4 Emergence and Saturation of Reliability

We first investigate how the amount of trajectory supervision affects the reliability of terminal agents. We vary only the token-level supervision horizon while keeping the base model, underlying training trajectories, and evaluation protocol fixed. The resulting trend is clear: extending the supervision horizon initially improves both the breadth and repeatability of task solving, but the gains diminish beyond an intermediate regime. In particular, substantially longer supervision incurs considerably higher training cost without a corresponding improvement in aggregate reliability.

4.1 Experimental Setting

Model and Training Data. Following prior work on terminal-agent training (Pi et al., 2026; Gandhi et al., 2026; Cheng et al., 2026; Shrivastava et al., 2026), we use `Qwen/Qwen3-8B` as the base model throughout our experiments. We train on trajectories from `nvidia/Nemotron-Terminal-Corpus`, which contains terminal-interaction trajectories generated by DeepSeek-V3.2 (Liu et al., 2025). Following Pi et al. (2026), we restrict training to completed trajectories, allowing us to isolate the effect of supervision horizon from confounding effects introduced by incomplete or imperfect trajectories. All training variants are constructed from the same underlying trajectory set, with supervision truncated at 4096, 8192, 12288, or 16384 response tokens. We refer to these settings as 4K, 8K, 12K, and 16K, respectively.

Evaluation Protocol. We evaluate the resulting models on Terminal-Bench v1.0 (Merrill et al., 2026), using the Terminus-2 agent scaffold throughout. Each task is evaluated with eight independent attempts. Repeated evaluation allows us to distinguish between *coverage* — whether a model can solve a task at all — and *repeatability* — whether it can solve the same task consistently across independent runs.

Reliability Metrics. Let $s_i = \sum_{r=1}^{8} \mathbf{1}\{\text{attempt } r \text{ succeeds on task } i\}$ be the number of successes for task i. We define

$$\mathrm{Rel}_{\ge k} = \sum_{i=1}^{M} \mathbf{1}\{s_i \ge k\}, \qquad \bar{s}_+ = \frac{\sum_{i:s_i\ge 1} s_i}{\mathrm{Rel}_{\ge 1}}, \tag{3}$$

where $\mathrm{Rel}_{\ge k}$ counts tasks solved in at least k of eight attempts, and $\bar{s}_+$ is the mean success count among solved tasks. We also report fragile tasks, defined by $1 \le s_i \le 2$.

4.2 Experimental Results

Reliability emerges with longer supervision. As shown in Table 2, increasing the supervision horizon from 4K to 12K substantially improves reliability. Successful attempts increase from 78±2.8 to 111±2.3, while task coverage rises from $\mathrm{Rel}_{\geq 1} = 24\pm0.8$ to 29±0.7. The improvement also extends to repeated success: $\mathrm{Rel}_{\geq 6}$ increases from 4±0.5 to 9±0.6, and $\mathrm{Rel}_{\geq 8}$ from 2±0.4 to 6±0.5. In every case, the gain far exceeds the variation across evaluations. Thus, longer supervision improves both the breadth and consistency of task solving.

Table 2: **Reliability across supervision horizons.** Values are mean ± variation over four evaluations.

Metric	4K	8K	12K	16K
		Aggregate		
Succ. ↑	78±2.8	97±2.6	**111±2.3**	110±2.7
$\mathrm{Rel}_{\geq 1}$ ↑	24±0.8	25±0.8	**29±0.7**	26±0.8
$\bar{s}_+$ ↑	3.25±0.1	3.88±0.1	3.83±0.1	**4.23±0.1**
		Repeatability		
$\mathrm{Rel}_{\geq 4}$ ↑	9±0.7	10±0.7	13±0.8	**14±0.8**
$\mathrm{Rel}_{\geq 6}$ ↑	4±0.5	8±0.6	**9±0.6**	**9±0.7**
$\mathrm{Rel}_{\geq 8}$ ↑	2±0.4	5±0.5	**6±0.5**	4±0.5
Fragile ↓	15±0.8	11±0.7	12±0.8	**9±0.7**

Reliability saturates beyond an intermediate horizon. As shown in Table 2, extending supervision from 12K to 16K yields little additional aggregate benefit. Successful attempts remain nearly unchanged (111±2.3 vs. 110±2.7, within variation), while task coverage decreases from 29±0.7 to 26±0.8 and $\mathrm{Rel}_{\geq 8}$ from 6±0.5 to 4±0.5. However, the reliability profile shifts: the number of fragile tasks decreases from 12±0.8 to 9±0.7, while $\mathrm{Rel}_{\geq 4}$ increases slightly from 13±0.8 to 14±0.8. Together with the higher mean success count ($\bar{s}_+ = 3.83\pm0.1$ vs. 4.23±0.1), this suggests that after the reliability peak, additional supervision primarily *redistributes* success toward more consistent performance on already-solvable tasks rather than expanding overall coverage.

We defer task-level reliability and further analyses in Appendix C.

Takeaway 1 (*Emergence-and-Saturation of Reliability*). Reliability improves markedly up to an intermediate horizon, after which additional supervision mainly redistributes success from fragile to more repeatable task solving rather than increasing aggregate reliability.

5 Horizon Controls Behaviors

5.1 Empirical Analysis

Longer horizons strengthen reactive problem solving. We classify a successful trajectory as proactive if the target failure never manifests, and as reactive if the failure manifests but the agent subsequently diagnoses and recovers from it to solve the task; details in Section B.4. As shown in Table 3, increasing the supervision horizon consistently increases the number of successful reactive trajectories, from 61 at 4K to 83 at 16K. This suggests that longer-horizon supervision supports successful diagnosis and recovery after errors occur. Proactive successes also increase from 17 at 4K to 29 at 12K before slightly decreasing to 27 at 16K, with the proactive share peaking at 26.13% at 12K. Overall, longer horizons are associated with more successful recovery trajectories while preserving strong proactive solving.

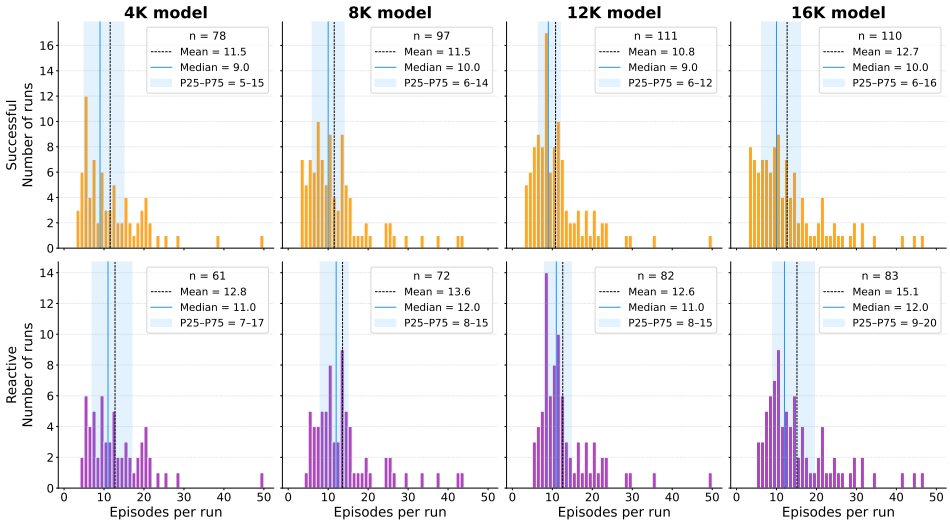
Table 3: **Proactive vs. reactive** successful trajectories across four context lengths. Proactive share peaks at 12K and declines at 16K.

Context length	4K	8K	12K	16K
Proactive	17	25	29	27
Reactive	61	72	82	83
Proactive share	**21.79%**	**25.77%**	**26.13%**	**24.55%**

Longer horizons can induce over-persistence. Together, Table 3 and Figure 2 suggest that 12K provides a favorable balance between proactive and reactive behavior. At 12K, proactive successes peak at 29 (26.13% of successful trajectories), while reactive successes reach 82. It also produces the shortest successful trajectories, averaging 10.8 episodes overall and 12.6 episodes among reactive runs. Scaling to 16K yields only a marginal increase in reactive successes (82→83), while proactive successes decrease (29→27) and successful trajectories become longer on average, both overall (10.8→12.7 episodes) and among reactive runs (12.6→15.1 episodes). These results suggest that the additional persistence induced by longer supervision becomes less efficient beyond 12K and is consistent with over-persistence at 16K. We leave detailed task-specific analyses in Appendix D.

Takeaway 2 (*Persistence Emerges and Eventually Overshoots*). Increasing the supervision horizon is associated with more successful recovery trajectories, with the most favorable balance around 12K. Further scaling provides little additional reactive success while producing longer trajectories, suggesting over-persistence at 16K.

Figure 2: **Episodes per run for successful trials, all and reactive only across training horizons.**



5.2 Why Does Reliability Saturate?

Longer supervision exposes later-stage behavior absent under short-horizon training, but these stages involve increasingly diverse histories. We formalize this tension as a trade-off between temporal supervision bias and finite-sample estimation difficulty. Let $L_H(\pi)$ be the expected imitation loss over the first H teacher decisions, and $\widehat{\pi}_H$ the empirical risk minimizer at horizon H.

Definition 5.1 (Temporal supervision bias)**.** Let $\Pi_H^\star$ be the set of prefix-optimal policies and $\pi_T^\star$ a full-horizon optimal policy. The temporal supervision bias at horizon H is

$$\begin{aligned} \Pi_H^\star &= \arg\min_{\pi\in\Pi} L_H(\pi), \qquad \pi_T^\star \in \arg\min_{\pi\in\Pi} L_T(\pi), \\ B(H) &:= \sup_{\pi\in\Pi_H^\star} L_T(\pi) - L_T(\pi_T^\star), \qquad B(H)\geq 0, \quad B(T)=0. \end{aligned} \tag{4}$$

$B(H)$ is the worst-case full-horizon risk left undetermined by supervising only the first H decisions. We assume it is non-increasing in H, while the horizon-induced policy complexity $\mathcal{C}(H)$ is non-decreasing (Lemma F.3).

Theorem 5.2 (Bias–Complexity Bound)**.** *Assume N independent teacher trajectories and a bounded loss $0 \leq \ell \leq M$. Let $L_H^\star := \min_{\pi\in\Pi} L_H(\pi)$, and assume the prefix-to-full-horizon transfer coefficient is bounded:*

$$\Gamma_H := \sup_{\pi\in\Pi:\, L_H(\pi)>L_H^\star} \frac{\left[L_T(\pi) - \sup_{\pi'\in\Pi_H^\star} L_T(\pi')\right]_+}{L_H(\pi) - L_H^\star} \leq \Gamma < \infty, \qquad \forall H \leq T. \tag{5}$$

Then, with probability at least $1-\delta$,

$$L_T(\widehat{\pi}_H) - L_T(\pi_T^\star) \leq B(H) + \sqrt{2}\,\Gamma M \sqrt{\frac{\mathcal{C}(H) + \log(2/\delta)}{N}}. \tag{6}$$

The proof, assumptions, and scope are in Appendix F. The result bounds imitation risk rather than task success, so we use it as a lens on the trade-off and test it empirically.

Empirical diagnostics. The two terms in Theorem 5.2 are not directly observable, so we measure one diagnostic for each on training trajectories longer than 16K tokens. For each horizon-H model θ_H, we compute

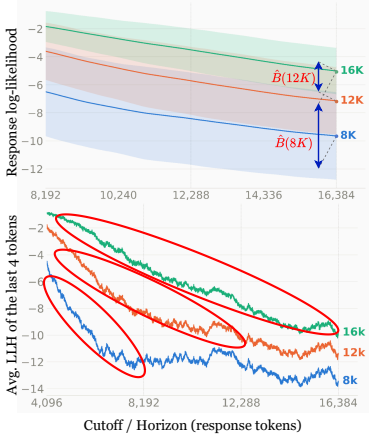
$$\begin{aligned} \bar{\ell}_H(K) &= \frac{1}{K}\sum_{t\leq K} \log p_{\theta_H}(x_t \mid x_{<t}), \\ \widehat{B}(H) &= \bar{\ell}_{16K}(K_{\max}) - \bar{\ell}_H(K_{\max}), \qquad \ell_H^{\mathrm{loc}}(K;w) = \frac{1}{w}\sum_{t=K+1}^{K+w} \log p_{\theta_H}(x_t \mid x_{<t}), \end{aligned} \tag{7}$$

where $K_{\max} = 16\text{K}$ and we use a local window of $w = 4$ tokens throughout. The bias proxy $\widehat{B}(H)$ targets the first term, measuring how much worse a model supervised up to H fits the full 16K trajectory than one supervised on all of it, thereby mirroring the truncation-induced excess risk in $B(H)$. The local difficulty $\ell_H^{\text{loc}}(K;4)$ targets the second term: tracking a short window's likelihood as K increases reveals how difficult each position becomes to model, reflecting the growing heterogeneity of late-stage histories that drives estimation burden. These quantities are diagnostics, not direct estimates of the theoretical terms (Section B.3).

As shown in Figure 3, $\ell_H^{\text{loc}}(K;4)$ drops substantially as the trajectory progresses, even within supervised regions, so later-stage behavior is harder to model. Meanwhile, $\widehat{B}(12K) < \widehat{B}(8K)$: longer supervision narrows the gap to the 16K reference, consistent with the assumption that temporal supervision bias is non-increasing in H. Together with Theorem 5.2, this suggests that longer supervision reduces truncation bias but adds behavior that is harder to estimate from finite data.

Interpretation. Theorem 5.2 and Figure 3 point to the same tension: longer supervision reduces the deficit caused by truncating later-stage behavior, but the newly introduced behavior is harder to estimate from finite data. This is consistent with the saturation observed in Section 4, and it motivates the selective refinement strategy in Section 6. The bound does not explain the behavioral shift toward over-persistence, which we regard as an empirical finding.

Figure 3: **Supervision Bias $\widehat{B}(\cdot)$ and Temporal Difficulty $\ell_H^{\text{loc}}(\cdot;4)$ for the 8K, 12K and 16K models.**



Takeaway 3 (*Estimation Burden*). Longer supervision reduces temporal bias but increases estimation burden by exposing the model to more heterogeneous late-stage behaviors. Empirically, supervision beyond the effective horizon coincides with over-persistence (Section 5.1).

6 Selective Long-Horizon Refinement

Motivation & Method Overview. Longer supervision exposes useful later-stage behaviors, but training on all long-horizon continuations does not consistently improve reliability. Motivated by the bias–complexity trade-off in Theorem 5.2 and increasing temporal difficulty in Figure 3, we propose a two-stage selective refinement strategy. We first train on all trajectories up to a shorter horizon H_1, then extend supervision to H_2 using continuations with high likelihood under the warm-start model. We interpret likelihood as *compatibility* with behavior learned at H_1, rather than as a universal measure of trajectory quality. This provides a conservative way to introduce long-horizon behavior and improve the reliability–efficiency trade-off without uniformly training on difficult continuations.

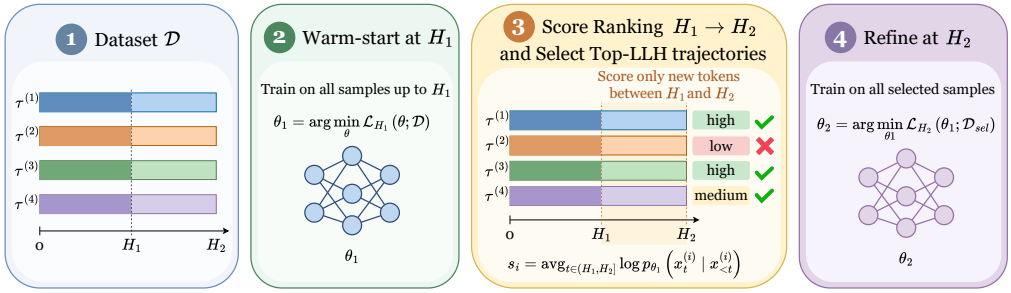


Figure 4: **Overview of the proposed two-stage selective long-horizon training method.**

Stage I: Data Preparation. Following Section 3, let $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^N$ denote the offline teacher dataset, where each trajectory $\tau^{(i)} = (x_1^{(i)}, \ldots, x_{L_i}^{(i)})$ contains L_i supervised response tokens. We consider two supervision horizons, $H_1 < H_2$, where H_1 is used for short-horizon warm-start training

and H_2 for subsequent long-horizon refinement. Since scoring requires observing the complete continuation from H_1 to H_2, we restrict the selection pool to trajectories of length at least H_2, $\mathcal{D}_{H_2} = \{\tau^{(i)} \in \mathcal{D} : L_i \geq H_2\}$. For each candidate trajectory, the interval $(H_1, H_2]$ is used for likelihood-based scoring and selection.

Stage II: Short-Horizon Warm Start. We train on the full dataset with supervision truncated at H_1:

$$\mathcal{L}_{H_1}(\theta; \mathcal{D}) = -\frac{\sum_{i=1}^{N} \sum_{t=1}^{\min(H_1, L_i)} \log p_\theta(x_t^{(i)} \mid x_{<t}^{(i)})}{\sum_{i=1}^{N} \min(H_1, L_i)}, \qquad \theta_1 = \arg\min_\theta \mathcal{L}_{H_1}(\theta; \mathcal{D}). \tag{8}$$

The resulting model θ_1 provides both the initialization for long-horizon training and the reference model for selecting learnable continuations.

Stage III: Continuation Scoring and Selection. For each $\tau^{(i)} \in \mathcal{D}_{H_2}$, we score the continuation between H_1 and H_2 using the warm-start model:

$$s_i(H_1, H_2) = \frac{1}{H_2 - H_1} \sum_{t=H_1+1}^{H_2} \log p_{\theta_1}(x_t^{(i)} \mid x_{<t}^{(i)}). \tag{9}$$

Given a selection ratio ρ, we retain the highest-scoring trajectories to form $\mathcal{D}_{\text{sel}} \subseteq \mathcal{D}_{H_2}$.

Stage IV: Selective Long-Horizon Refinement. Starting from the warm-start checkpoint θ_1, we continue training on $\mathcal{D}_{\text{sel}}$ with the supervision horizon extended to H_2. Since all selected trajectories satisfy $L_i \geq H_2$, the refinement objective is

$$\mathcal{L}_{H_2}(\theta; \mathcal{D}_{\text{sel}}) = -\frac{1}{|\mathcal{D}_{\text{sel}}| H_2} \sum_{\tau^{(i)} \in \mathcal{D}_{\text{sel}}} \sum_{t=1}^{H_2} \log p_\theta \left(x_t^{(i)} \mid x_{<t}^{(i)} \right), \qquad \theta_2 = \arg\min_{\theta\,;\,\theta \leftarrow \theta_1} \mathcal{L}_{H_2}(\theta; \mathcal{D}_{\text{sel}}). \tag{10}$$

7 Experiments

We compare selective long-horizon refinement with one-stage training at 4K, 8K, 12K, and 16K. For 12K and 16K refinement, we warm-start from the 8K checkpoint, rank candidate continuations by likelihood under the warm-start model, and retain the top or bottom 25%, 50%, or 75%for training. All models are trained for two epochs, with each stage of two-stage training run for two epochs. Following Section 4, we report successful attempts, Rel$\geq$ 1, Rel$\geq$ 4, Rel$\geq$ 6, Rel$\geq$ 8, $\bar{s}_+$, and fragile tasks. For each checkpoint, we evaluate eight attempts per task, reporting the mean and run-to-run variation across four evaluations. Our main results use Terminal-Bench v1.0, while results on Terminal-Bench v2.0 and OpenThoughts-TBLite (Merrill et al., 2026) are deferred to Appendix E.4.

7.1 Main Results

Selective refinement improves long-horizon training. With $\rho \geq 50\%$, Top-ρ refinement improves over one-stage training at the same horizon (Table 5). At 12K, Top-50 increases successful attempts from 111±2.3 to 121±2.3, raises $\text{Rel}_{\geq 6}$ from 9±0.6 to 11±0.6, and reduces fragile tasks from 12±0.8 to 9±0.7. At 16K, Top-75 performs best, increasing successful attempts from 110±2.7 to 126±2.1, $\text{Rel}_{\geq 6}$ from 9±0.7 to 14±0.6, and $\text{Rel}_{\geq 8}$ from 4±0.5 to 7±0.5. These gains exceed run-to-run variation, showing that selective long-horizon supervision is more effective than uniform training.

Table 4: **Training cost of one-stage and selective two-stage training.** **Top**-50% models are warm-started from 8K; *Total* includes the warm-start stage. Parentheses give the change vs. one-stage training at the same horizon.

Hor.	Strategy	#Samples	Time (h)	Total (h)↓
8K	Full	59,408	21.65	21.65
12K	Full	59,408	35.44	35.44
12K	Top-50%	19,568	11.83	**33.48** (−5.5%)
16K	Full	59,408	50.81	50.81
16K	Top-50%	19,600	17.48	39.13 (−23.0%)

Selective refinement also reduces training cost. The refinement stage uses only a third of the samples of one-stage training (19,568 vs. 59,408; Table 4). Including the 8K warm start, 8K→12K **Top**-50% takes 33.48 h in total, less than one-stage 12K (35.44 h) and 16K (50.81 h) training, while solving more attempts (121 vs. 111 and 110). At 16K, selective refinement cuts total training time by 23.0% (39.13 vs. 50.81 h).

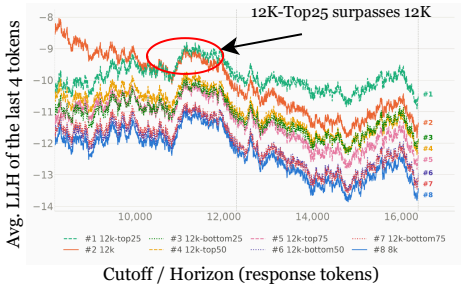
Table 5: **Reliability of one-stage and selective two-stage training on Terminal-Bench v1.0.** Two-stage models warm-start from 8K and refine at 12K or 16K on the highest- (**Top**) or lowest-likelihood (Bot.) $\rho\%$ of continuations. Values are reported with estimated run-to-run variation. Shaded : improves over one-stage training at the same horizon; **green**: best in row.

	One-stage				12K refinement (from 8K)						16K refinement (from 8K)					
	training horizon				ρ=25%		ρ=50%		ρ=75%		ρ=25%		ρ=50%		ρ=75%	
Metric	4K	8K	12K	16K	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.
Aggregate reliability																
Succ. attempts ↑	78±2.8	97±2.6	111±2.3	110±2.7	125±2.2	108±2.8	121±2.3	112±2.6	119±2.5	117±2.7	114±2.6	108±2.9	122±2.4	120±2.5	**126±2.1**	114±2.7
$\mathrm{Rel}_{\geq 1}$ ↑	24±0.8	25±0.8	29±0.7	26±0.8	26±0.7	28±0.8	27±0.7	28±0.8	29±0.7	**31±0.8**	27±0.7	24±0.8	28±0.7	25±0.8	26±0.7	30±0.8
$\bar{s}_+$ ↑	3.25±0.1	3.88±0.1	3.83±0.1	4.23±0.1	4.81±0.1	3.86±0.1	4.48±0.1	4.00±0.1	4.10±0.1	3.77±0.1	4.22±0.1	4.50±0.1	4.36±0.1	4.80±0.1	**4.85±0.1**	3.80±0.1
Repeatability																
$\mathrm{Rel}_{\geq 4}$ ↑	9±0.7	10±0.7	13±0.8	14±0.8	**19±0.7**	13±0.8	15±0.7	15±0.8	16±0.8	14±0.8	14±0.7	13±0.8	15±0.7	16±0.8	15±0.7	15±0.8
$\mathrm{Rel}_{\geq 6}$ ↑	4±0.5	8±0.6	9±0.6	9±0.7	10±0.6	7±0.6	11±0.6	8±0.6	9±0.6	9±0.7	9±0.6	9±0.7	11±0.6	12±0.7	**14±0.6**	11±0.7
$\mathrm{Rel}_{\geq 8}$ ↑	2±0.4	5±0.5	6±0.5	4±0.5	5±0.5	4±0.5	6±0.5	3±0.4	6±0.5	5±0.5	5±0.5	4±0.5	6±0.5	6±0.5	**7±0.5**	4±0.5
Fragile ($1 \leq s_i \leq 2$) ↓	15±0.8	11±0.7	12±0.8	9±0.7	**6±0.6**	13±0.8	9±0.7	9±0.7	10±0.7	14±0.8	10±0.7	9±0.7	9±0.7	8±0.6	9±0.7	13±0.8

7.2 Ablation Study

Does high-likelihood selection improve long-horizon learnability? Figure 5 provides direct evidence that high-likelihood selection improves long-horizon learnability. At the same selection ratio, Top-likelihood models consistently achieve higher local likelihood than their Bottom-likelihood counterparts (Top-25% > Bottom-25%, Top-50% > Bottom-50%, and Top-75% > Bottom-75%). Notably, Top-25% even surpasses full 12K training around the target horizon. These results suggest that warm-start likelihood identifies continuations that are more compatible with the learned short-horizon policy and therefore easier to model during long-horizon refinement.

Figure 5: **Temporal Supervision Difficulty $\ell_H^{\mathrm{loc}}(\cdot; 4)$ of the base, selective 12K-refined, and full 12K models.**



How much long-horizon data should be retained? Table 5 and Figure 5 show a horizon-dependent quality–quantity trade-off. At 12K, Top-25 gives the most successes (125±2.2), while Top-50 yields slightly stronger $\mathrm{Rel}_{\geq 6}$ (11 vs. 10); at 16K, Top-75 is best (126±2.1 successes, $\mathrm{Rel}_{\geq 6} = 14$). This matches Theorem 5.2: smaller ρ improves continuation compatibility but reduces N_{sel}, increasing the estimation burden $O\left(\sqrt{\mathcal{C}(H)/N_{\mathrm{sel}}}\right)$. Since $\mathcal{C}(H)$ also grows with horizon, longer horizons require retaining more data to control estimation error.

Does likelihood-based selection matter? Likelihood-based selection matters, but its effect depends on the selection ratio and target horizon (Table 5). At 12K with $\rho = 50\%$, Top-50 clearly outperforms Bottom-50, with 121±2.3 vs. 112±2.6 successful attempts, $\mathrm{Rel}_{\geq 6} = 11 \pm 0.6$ vs. 8±0.6, and $\mathrm{Rel}_{\geq 8} = 6 \pm 0.5$ vs. 3±0.4. These gaps exceed the estimated run-to-run variation, whereas at other settings the differences narrow or reverse. Thus, likelihood is a useful compatibility signal, but should be considered jointly with the retained data and target horizon.

Is a longer refinement horizon always beneficial? No. At $\rho = 50\%$, 12K-Top and 16K-Top are statistically comparable (121±2.3 vs. 122±2.4 successes; both $\mathrm{Rel}_{\geq 6} = 11 \pm 0.6$). At $\rho = 75\%$, however, extending to 16K improves 119±2.5 → 126±2.1 successes and $\mathrm{Rel}_{\geq 6}$ from 9±0.6 to 14±0.6. Thus, longer refinement helps only when enough continuation data are retained.

Appendix E provides task-level and selection analyses, schedule/data-quantity controls, and cross-benchmark results on Terminal-Bench v2.0 and OpenThoughts-TBLite.

8 Conclusion

We study the supervision horizon in terminal-agent training and identify an emergence-and-saturation pattern: reliability improves with longer supervision but plateaus beyond an intermediate horizon. Based on this, we propose selective long-horizon refinement, using continuation likelihood to retain more learnable long-horizon trajectories while reducing unnecessary training burden.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, LaTeX formatting, and code debugging. All research ideas, experimental design, analyses, results, and conclusions were developed and verified by the authors.

ETHICS STATEMENT

This work studies the training and evaluation of terminal agents using existing datasets and benchmarks. It does not involve human subjects or personally identifiable information. While more capable terminal agents could potentially be misused for harmful automation, our work focuses on improving training reliability and efficiency rather than expanding access to harmful capabilities.

REFERENCES

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L Leavitt, and Mansheej Paul. Perplexed by perplexity: Perplexity-based pruning with small reference models. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2024. URL https://openreview.net/forum?id=wNseMYxwpq.

Anthropic. Introducing claude opus 4.8. https://www.anthropic.com/news/claude-opus-4-8, May 2026. Accessed: 2026-09-20.

Zihao Cheng, Hongru Wang, Zeming Liu, Xinyi Wang, Xiangrong Zhu, Yuhang Guo, Wei Lin, Jeff Z Pan, and Yunhong Wang. Terminal-world: Scaling terminal-agent environments via agent skills. *arXiv preprint arXiv:2605.20876*, 2026.

Dang Cao Cuong, Hoang Anh Just, and Ruoxi Jia. Capacity-dependent effects of data selection for mathematical reasoning. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=2QZkhzstnL.

Kanishk Gandhi, Shivam Garg, Noah D Goodman, and Dimitris Papailiopoulos. Endless terminals: Scaling rl environments for terminal agents. *arXiv preprint arXiv:2601.16443*, 2026.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7602–7635, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.421. URL https://aclanthology.org/2024.naacl-long.421/.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023.

Mike A Merrill, Alexander Glenn Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Kwesi Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Jenia Jitsev, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak.

OpenAI. Introducing codex. https://openai.com/index/introducing-codex/, May 2025. Accessed: 2026-09-20.

Renjie Pi, Grace Lam, Mohammad Shoeybi, Pooya Jannaty, Bryan Catanzaro, and Wei Ping. On data engineering for scaling LLM terminal capabilities. In *Deep Learning for Code: Towards Human-Centered Coding Agents*, 2026. URL https://openreview.net/forum?id=BykU9KPyQ1.

Xuan Ren, Biao Wu, and Lingqiao Liu. I learn better if you speak my language: Understanding the superior performance of fine-tuning large language models with LLM-generated responses. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10225–10245, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.571. URL https://aclanthology.org/2024.emnlp-main.571/.

Vaishnavi Shrivastava, Piero Kauffmann, Ahmed Awadallah, and Dimitris Papailiopoulos. Echo: Terminal agents learn world models for free. *arXiv preprint arXiv:2605.24517*, 2026.

Siwei Wu, Yizhi LI, Yuyang Song, Wei Zhang, Yang Wang, Riza Batista-Navarro, Xian Yang, Mingjie Tang, Bryan Dai, Jian Yang, and Chenghua Lin. Large-scale terminal agentic trajectory generation from dockerized environments. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=PeFSCRulgy.

Dylan Zhang, Qirun Dai, and Hao Peng. The best instruction-tuning data are those that fit. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4jFSekBaDT.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

Appendix Contents

A Extended Related Works **13**

B Setting **14**

B.1 Training Setting 14

B.2 Evaluation Protocol and Metrics 14

B.3 Empirical Measures of the Bias–Complexity Trade-off 14

B.4 Proactive vs. Reactive Classification 15

C Task-Level Emergence and Saturation of Reliability **15**

C.1 From Fragile to Repeatable Success 15

C.2 Saturation as Cancellation, Not a Plateau 15

D Horizon and Agent Behavior **17**

D.1 Longer Supervision Enables Recovery 17

D.2 Beyond the Effective Horizon: Over-Persistence 18

E Additional Results for Selective Refinement **18**

E.1 Task-Level Results 18

E.2 Top- vs. Bottom-Likelihood Selection 19

E.3 How do training schedule, data quantity, and likelihood selection interact? 19

E.4 Results on Other Benchmarks 19

F Missing Proofs **21**

A Extended Related Works

We expand on Section 2 and discuss additional lines of related work.

Terminal-Agent Training and Data Engineering. Recent work improves terminal agents through data scaling, environment construction, and richer supervision. Gandhi et al. (2026), Cheng et al. (2026), and Shrivastava et al. (2026) improve environments and supervision signals, while Pi et al. (2026) study long-context training and observe that extending context does not always improve performance. However, their analysis measures aggregate accuracy and leaves open *how longer supervision changes agent reliability*. We show that aggregate accuracy hides a qualitative shift: beyond an intermediate horizon, longer supervision unlocks few new tasks but makes already-solvable tasks more repeatable, redistributing success from fragile to consistent solving. Building on this observation, we characterize how the supervision horizon shapes agent behavior, from under-persistence to productive recovery and eventually over-persistence, explain the saturation theoretically through a bias–complexity trade-off, and use this analysis to motivate selective refinement of newly exposed long-horizon trajectory segments.

Likelihood-based data selection for SFT. Data quality often matters more than quantity: LIMA (Zhou et al., 2023) shows that a small, well-chosen dataset can rival much larger ones. One line of work selects data by how well it fits the model being trained. Ren et al. (2024) show that LLM-generated responses outperform human-written ones largely because their lower perplexity makes them easier to learn, and Zhang et al. (2025) select, among candidate responses, those with

the highest likelihood under the target model. Other criteria instead favor harder examples, e.g., instruction-following difficulty (Li et al., 2024), and perplexity-based pruning in pretraining finds that the best subset depends on the reference model and the retained fraction (Marion et al., 2023; Ankner et al., 2024). Reconciling these views, Cuong et al. (2026) show that the preference for high-likelihood data is not universal: in mathematical reasoning, it yields faster early gains, especially for small students, whereas low-likelihood data becomes more beneficial for larger students under longer training. These methods score *whole* examples, typically in single-turn settings. *In contrast, we score only the newly exposed continuation $(H_1, H_2]$ of each long-horizon trajectory, using a warm-start model trained on shorter prefixes, and treat likelihood as compatibility with the learned short-horizon policy rather than as intrinsic quality.*

B Setting

This section provides the detailed setup used throughout our paper, including the training configuration, evaluation protocol, reliability metrics, and auxiliary analyses underlying the main results.

B.1 Training Setting

For the experiments in Section 4, Section 5 and Section 6, we train all models with full-parameter supervised fine-tuning using LLaMA-Factory on the Nemotron terminal-agent corpus, which contains 59,407 ShareGPT-format trajectories. The base model for one-stage training is `Qwen/Qwen3-8B`. Unless otherwise specified, all runs use the same hyperparameters: two training epochs, BF16 precision, DeepSpeed ZeRO-3 on four NVIDIA H100 GPUs, per-device batch size 1 with no gradient accumulation (global batch size 4), AdamW with learning rate 1×10^{-5}, cosine decay without warmup, gradient clipping at 1.0, and no sequence packing. We train complete-data models with supervision cutoffs of 4,096, 8,192, 12,288, and 16,384 tokens, respectively, while keeping the underlying 59,407 trajectories fixed so that these runs isolate the effect of supervision horizon. For selective refinement, we warm-start from the complete 8K checkpoint and continue training for two additional epochs at either 12K or 16K on likelihood-filtered subsets of long trajectories. Candidate trajectories are those extending beyond 8,192 tokens and are ranked by the average teacher-forced log-probability assigned by the 8K model to assistant tokens in the newly exposed continuation, i.e., 8K–12K or 8K–16K. We retain either the highest- or lowest-likelihood 25%, 50%, or 75% of the scored trajectories, yielding approximately 9.8K, 19.6K, and 29.4K training examples, respectively. Trajectories not extending beyond 8K are excluded from the selective-refinement stage. As shown in Table 4, the 12K Top-50% setting contains slightly fewer samples than the 16K Top-50% setting because some candidate trajectories do not contain enough tokens in the required 8K–12K continuation and are therefore skipped during scoring and selection.

B.2 Evaluation Protocol and Metrics

We evaluate all models on Terminal-Bench v1.0 using the Terminus-2 scaffold, with eight independent attempts per task. For each task i, $s_i \in \{0, \dots, 8\}$ denotes the number of successful attempts; the reliability metrics $\mathrm{Rel}_{\geq k}$, $\bar{s}_+$, and fragile tasks ($1 \leq s_i \leq 2$) are defined in Section 4. Because each task has only eight attempts, per-task differences of one or two successes are within sampling noise; in task-level analyses we therefore interpret only large or consistent shifts.

B.3 Empirical Measures of the Bias–Complexity Trade-off

The metrics in Equation 7 are empirical diagnostics rather than estimates of the quantities in Theorem 5.2. The bias $B(H)$ is defined through population risks over the unknown prefix-optimal set $\Pi_H^\star$ and the full-horizon optimum $\pi_T^\star$, while the complexity term depends on the policy class. In particular, $\widehat{B}(H)$ measures the gap to the trained 16K model, itself a finite-sample estimate, rather than to $\pi_T^\star$, and corresponds to a single element of $\Pi_H^\star$ rather than the supremum in $B(H)$. We therefore construct observable metrics that capture the same intuition. Specifically, $\widehat{B}(H)$ measures the full-horizon modeling gap between the empirically trained horizon-H policy and the 16K reference policy, mirroring the population risk gap in $B(H)$. Meanwhile, $\ell_H^{\mathrm{loc}}(K; w)$ measures the average log-likelihood of a local token window on the training trajectories. Tracking this quantity as

K increases reveals how the empirical learnability of behavior changes deeper into the trajectory: decreasing local likelihood indicates that later-stage behavior is increasingly difficult for the learned policy to model. Thus, these metrics do not numerically instantiate the theoretical terms, but provide empirical diagnostics for the same bias–difficulty trade-off.

B.4 Proactive vs. Reactive Classification

For the experiment in Section 5.1, we classify each trial according to whether the agent encounters an execution error during interaction. A trial is *proactive* if no genuine command error appears in any terminal observation returned to the agent, and *reactive* if at least one such error occurs and the agent must respond to it. We inspect the terminal observation after every interaction step, excluding the initial task prompt, and match lines against common error signatures, including Python tracebacks and `*Error`/`*Exception`, `command not found`, `No such file or directory`, permission and access failures, `fatal`/`error`, segmentation faults, missing modules, syntax errors, linker failures, `FAILED`, `Killed`, connection failures, and `apt E:` messages. To avoid false positives, we ignore lines that merely echo the agent's own commands or code, harness/parser messages about response formatting, and warnings. This classification is independent of task outcome: a resolved trial may therefore be either proactive, meaning it succeeds without encountering an error, or reactive, meaning it succeeds after encountering and recovering from an error. The procedure is intentionally mechanical and identifies whether an error occurred, rather than inferring how the agent reasoned about it.

C Task-Level Emergence and Saturation of Reliability

Aggregate metrics in Section 4 summarize how reliability changes with the supervision horizon but not which tasks drive the change. Table 6 reports per-task successes for all twelve configurations; this section uses the four one-stage columns, and Appendix E discusses the selective ones. With eight attempts per task, differences of one or two successes are within sampling noise (Appendix B.2), so we focus on large or consistent shifts.

C.1 From Fragile to Repeatable Success

At 4K, success is sparse and fragile: 15 of the 24 solved tasks succeed in only one or two of eight attempts. Extending the horizon to 12K mostly moves tasks up the reliability distribution rather than adding isolated successes. For example, `incompatible-python-fasttext.base_with_hint` improves from $2/8$ to $8/8$, `git-workflow-hack` from $2/8$ to $7/8$, `new-encrypt-command` from $2/8$ to $5/8$, and `sqlite-with-gcov` from $1/8$ to $4/8$.

Some tasks become solvable only once the horizon is long enough. `crack-7z-hash` is never solved at 4K or 8K but reaches $4/8$ at 12K and $5/8$ at 16K; `conda-env-conflict-resolution` is first solved at 12K; and `blind-maze-explorer-5x5` is solved only at 16K. Successful runs on these tasks take 20–33 episodes of sustained trial and adjustment (Table 8), so short-horizon training rarely observes the teacher behavior they require. Consistently, the 16K model produces 21 successful trials lasting at least 20 episodes, spread across 12 tasks, whereas the 8K model produces 11 across 6 tasks.

C.2 Saturation as Cancellation, Not a Plateau

From 12K to 16K, the aggregate success count barely moves ($111 \rightarrow 110$), but this hides substantial task-level movement. Of the 34 tasks solved by at least one one-stage model, 11 improve, 13 worsen, and 10 are unchanged. The improving tasks gain 18 successes and the degrading tasks lose 19:

$$\underbrace{+18}_{\text{gained successes}} + \underbrace{(-19)}_{\text{lost successes}} = -1. \tag{11}$$

The gains concentrate on tasks that require long recovery, such as `modernize-fortran-build` ($3/8 \rightarrow 7/8$) and `blind-maze-explorer-5x5` ($0 \rightarrow 2/8$). The losses fall mainly on tasks that peak at 12K: `git-workflow-hack` ($7/8 \rightarrow 3/8$),

Table 6: **Task-level reliability across one-stage and selective training.** Each cell reports the number of successful attempts out of eight; shading scales with success count (1 3 5 8), and · denotes zero. T/B denote selection of the highest-/lowest-likelihood continuations during refinement. **#Runs** counts the configurations (out of 12) that solve each task at least once; **green** marks the best value in each summary row.

	One-stage				12K refinement				16K refinement				
Task	**4K**	**8K**	**12K**	**16K**	**T50**	**B50**	**T75**	**B75**	**T50**	**B50**	**T75**	**B75**	**#Runs**
Solved by all 12 configurations													
crack-7z-hash.easy	4	1	6	4	3	6	6	5	4	8	6	7	12
create-bucket	1	3	2	3	2	3	4	2	3	3	1	6	12
csv-to-parquet	8	8	8	7	8	7	8	8	8	8	8	8	12
extract-safely	5	7	8	8	7	7	8	6	8	8	8	7	12
fibonacci-server	2	2	3	4	8	5	5	5	7	6	7	6	12
fix-pandas-version	7	7	8	8	8	8	8	7	8	8	8	8	12
fix-permissions	7	8	8	8	8	8	8	8	7	8	8	8	12
git-workflow-hack	2	3	7	3	5	4	3	3	3	5	7	6	12
grid-pattern-transform	1	2	1	2	5	1	1	3	2	2	1	1	12
hello-world	8	8	8	8	8	8	8	8	8	8	8	8	12
heterogeneous-dates	5	8	4	6	8	5	3	4	6	6	6	6	12
incompatible-python-fasttext	2	3	2	2	1	3	2	6	3	5	3	3	12
incompatible-python-fasttext.base_with_hint	2	4	8	6	6	3	6	8	8	7	8	5	12
modernize-fortran-build	2	3	3	7	5	5	5	2	4	6	7	1	12
new-encrypt-command	2	2	5	4	5	6	6	6	6	6	7	5	12
simple-web-scraper	2	8	3	5	6	5	5	8	7	4	7	7	12
sqlite-with-gcov	1	2	4	2	6	4	5	4	5	5	3	4	12
swe-bench-langcodes	5	6	7	6	7	7	8	5	8	7	8	4	12
Solved by a subset of configurations													
configure-git-webserver	·	1	1	2	2	1	1	2	2	·	2	1	10
crack-7z-hash	·	·	4	5	3	4	4	3	3	1	4	3	10
vim-terminal-task	4	5	1	3	1	1	4	1	·	·	1	1	10
conda-env-conflict-resolution	·	·	1	1	1	1	2	1	1	·	2	1	9
blind-maze-explorer-algorithm	·	1	·	·	1	2	·	2	1	1	·	1	7
processing-pipeline	·	1	1	·	2	3	3	2	1	·	·	·	7
blind-maze-explorer-algorithm.hard	1	2	·	1	·	·	·	1	2	·	·	1	6
get-bitcoin-nodes	2	·	1	1	·	1	1	1	·	·	·	·	6
tmux-advanced-workflow	2	·	1	·	·	·	1	1	·	1	2	·	6
blind-maze-explorer-5x5	·	·	·	2	3	1	·	1	·	·	·	1	5
oom	2	·	1	·	·	2	·	2	·	·	1	·	5
simple-sheets-put	·	·	·	·	·	·	1	1	1	2	2	·	5
git-multibranch	1	1	·	·	·	·	·	·	1	·	1	·	4
openssl-selfsigned-cert	·	·	3	2	·	·	·	1	4	·	·	·	4
fix-git	·	·	1	·	·	1	·	·	·	·	·	1	3
password-recovery	·	1	·	·	·	·	·	·	·	2	·	1	3
blind-maze-explorer-algorithm.easy	·	·	1	·	1	·	·	·	·	·	·	·	2
prove-plus-comm	·	·	·	·	·	·	1	·	1	·	·	·	2
crack-7z-hash.hard	·	·	·	·	·	·	·	·	·	1	·	·	1
decommissioning-service-with-sensitive-data	·	·	·	·	·	·	·	·	·	·	·	1	1
eval-mteb	·	·	·	·	·	·	·	·	·	·	·	1	1
eval-mteb.hard	·	·	·	·	·	·	·	·	·	·	·	1	1
organization-json-generator	·	·	·	·	·	·	1	·	·	·	·	·	1
path-tracing	·	·	·	·	·	·	·	·	·	2	·	·	1
path-tracing-reverse	·	·	·	·	1	·	·	·	·	·	·	·	1
swe-bench-fsspec	·	·	·	·	·	·	1	·	·	·	·	·	1
Correct tasks ($\mathrm{Rel}_{\geq 1}$)	**24**	**25**	**29**	**26**	**27**	**28**	**29**	**31**	**28**	**25**	**26**	**30**	
Successful attempts	78	97	111	110	121	112	119	117	122	120	**126**	114	

`incompatible-python-fasttext.base_with_hint` (8/8 → 6/8), `sqlite-with-gcov` (4/8 → 2/8), and `openssl-selfsigned-cert` (3/8 → 2/8). In addition, five tasks solved once at 12K become unsolved at 16K.

This cancellation explains the mixed aggregate profile in Table 2. Coverage ($\mathrm{Rel}_{\geq 1}$) and perfect repeatability ($\mathrm{Rel}_{\geq 8}$) fall because marginal and 12K-peak tasks lose ground. Meanwhile, $\mathrm{Rel}_{\geq 4}$, $\bar{s}_{+}$, and the number of fragile tasks improve because other tasks become more consistent. Saturation is therefore not a uniform plateau: beyond 12K, additional supervision redistributes reliability across tasks rather than increasing it.

Caveat. A few tasks are solved most often at the shortest horizon; for example, `oom` and `tmux-advanced-workflow` succeed in $2/8$ attempts at 4K and in at most $1/8$ at longer horizons. These differences are within sampling noise, and we do not interpret them as horizon effects.

Table 7: **Recovery persistence across supervision horizons.** *Taxonomy*: successful trajectories over the 36 manually analyzed tasks, classified as proactive (pitfall anticipated) or reactive (error observed, then repaired). *Reactive tasks*: successes out of eight on tasks that require recovery; every `modernize-fortran-build` success is reactive. *Pass rate*: over the five tasks whose eight trials are all classified. **Green**: best in row.

Metric	4K	8K	12K	16K
Taxonomy (36 tasks)				
Proactive successes	42	53	**56**	54
Reactive successes	27	38	41	**42**
Reactive share	39%	42%	42%	**44%**
Reactive tasks				
`modernize-fortran-build`	2/8	3/8	3/8	**7/8**
`fibonacci-server`	2/8	2/8	3/8	**4/8**
`simple-web-scraper` (reactive only)	0/8	2/8	0/8	**4/8**
Pass rate (5 fully classified tasks)	50%	70%	63%	**80%**

Table 8: **Task-level behavior across supervision horizons.** Successes out of eight, with mean episodes per run in parentheses (shown for solved settings). Counts are a subset of Table 6; **green**: best in row.

Task	4K	8K	12K	16K
Long-horizon emergence				
crack-7z-hash	0	0	4 (27.1)	**5** (32.6)
conda-env-conflict-resolution	0	0	**1** (32.5)	**1** (25.1)
blind-maze-explorer-5x5	0	0	0	**2** (20.6)
Efficient recovery				
modernize-fortran-build	2 (10.5)	3 (15.5)	3 (11.6)	**7** (12.9)
Over-persistence				
simple-web-scraper	2 (6.9)	**8** (7.4)	3 (7.3)	5 (9.3)
vim-terminal-task	4 (28.3)	**5** (13.0)	1 (18.8)	3 (23.1)

D Horizon and Agent Behavior

This section complements Section 5.1 with task-level evidence on how the horizon shapes *persistence*: the agent's tendency to keep acting after an error, diagnose it, and recover rather than terminate. Table 3 in the main text uses the automatic error-signature classifier of Appendix B.4 over all runs. Here we additionally use a finer manual taxonomy over 36 tasks, which labels a successful trajectory *proactive* if the relevant pitfall is anticipated before it manifests and *reactive* if the error first occurs and is then diagnosed and repaired (Table 9).

D.1 Longer Supervision Enables Recovery

Reactive successes grow steadily with the horizon, from 27 at 4K to 42 at 16K, and their share of successful trajectories rises from 39% to 44% (Table 9). Because some tasks are structurally proactive or reactive, this ratio alone is not evidence of persistence; the more informative comparison is across checkpoints on the *same* task.

`modernize-fortran-build` is structurally reactive: the build must fail before it can be repaired, so every success is a recovery. Successes rise from $2/8$ at 4K to $3/8$ at 8K and 12K, and then to $7/8$ at 16K. Mean episodes at 16K (12.9) remain below those at 8K (15.5), so the gain reflects better-targeted recovery rather than more attempts (Table 8). `fibonacci-server` shows a similar trend, rising from $2/8$ to $4/8$.

The contrast is sharpest on `simple-web-scraper`. The 4K model never recovers after observing the target bug: it dismisses the observed behavior as acceptable, edits the downstream report instead of the corrupted CSV, or spends its remaining budget on an unrelated error. At 16K, four of the five successful trajectories recover reactively, one after 22 episodes of revisiting its diagnosis. Across the five tasks whose eight trials are all classified, the pass rate rises from 50% at 4K to 80% at 16K. Longer supervision therefore exposes the learner to later trajectory stages in which an initial

Table 9: **Recovery persistence across supervision horizons.** Successful trajectories over the 36 manually analyzed tasks, classified as proactive (pitfall anticipated) or reactive (error observed, then repaired). Pass rate is computed over the five tasks whose eight trials are all classified. **green**: best in row.

	4K	8K	12K	16K
Proactive successes	42	53	**56**	54
Reactive successes	27	38	41	**42**
Reactive share	39%	42%	42%	**44%**
Pass rate (5 tasks)	50%	70%	63%	**80%**

action has already failed, shifting behavior from Act → Observe Failure → Terminate toward Act → Observe Failure → Diagnose → Retry → Recover.

D.2 Beyond the Effective Horizon: Over-Persistence

The same tendency to continue becomes costly on tasks that shorter horizons already solve. On `simple-web-scraper` and `vim-terminal-task`, the 8K model is best, and longer horizons both lower success and lengthen runs. For example, 8K solves `vim-terminal-task` in $5/8$ attempts with 13.0 episodes per run, whereas 16K solves it in $3/8$ with 23.1 (Table 8).

Failed 16K trajectories on `simple-web-scraper` reveal the mechanism. The agent often produces a working script, notices a minor formatting issue, and then either enters repetitive correction or introduces a new error that breaks the solution; one failure repeats the same incorrect diagnosis over several episodes. In other runs, the agent verifies only coarse properties and terminates while semantic defects remain. The same tendency appears in aggregate: 89% of successful 8K runs finish within 20 episodes, compared with 81% at 16K, and mean episodes of successful runs rise from 10.8 at 12K to 12.7 at 16K (Figure 2).

Together, these results reconcile the two faces of 16K. It shows the strongest recovery persistence on structurally reactive tasks, but this persistence is not selective: the same drive to continue helps tasks that require recovery and hurts tasks that were already solved. Behavior therefore moves from *under-persistence* at 4K, through productive persistence around 12K, to *over-persistence* at 16K, consistent with Section 5.1.

E Additional Results for Selective Refinement

This section provides additional analyses supporting the selective long-horizon refinement results in Section 6 and Section 7 of the main paper. We further examine the effect of selection ratio, likelihood-based filtering, refinement horizon, and evaluation benchmark, complementing the main results and ablations on when and why selective refinement improves long-horizon agent training.

E.1 Task-Level Results

Selective refinement consistently improves aggregate success. The advantage of selective refinement is not confined to a single configuration. All eight filtered models in Table 6 achieve 112–126 successful attempts, exceeding the best one-stage result of 111. Uniformly supervising all long-horizon continuations is therefore unnecessary for obtaining the benefits of long-horizon training.

Selection trades coverage against repeatability. Successful attempts and task coverage need not move together. 16K-T75 obtains the most successful attempts (126) but solves 26 tasks, whereas 16K-B75 obtains fewer (114) while solving 30. Likelihood-based selection thus redistributes reliability: high-likelihood subsets favor repeated success on solvable tasks, whereas low-likelihood subsets expand the set of tasks solved.

E.2 Top- vs. Bottom-Likelihood Selection

Top-likelihood refinement repairs regressions and shortens runs. Top-likelihood selection recovers several tasks that regress under full 16K training, usually with shorter runs (16K columns of Table 6). `fibonacci-server` improves from $4/8$ under full 16K training to $7/8$ under both T50 and T75, while mean episodes per run nearly halve ($14.0 \rightarrow 7.9$ under T50). `incompatible-python-fasttext.base_with_hint` returns to $8/8$, matching its 12K peak, and `swe-bench-langcodes` rises from $6/8$ to $8/8$. `fix-pandas-version` isolates the efficiency effect: every 16K variant solves it $8/8$, but T50 needs 9.9 episodes per run versus 17.5 for full 16K training. In aggregate, T75 yields more completed runs than B75 (209 vs. 197), with shorter completed trajectories (10.7 vs. 12.2 mean episodes).

Selection is not uniformly beneficial, however. `vim-terminal-task`, already an over-persistence case, degrades further under Top selection ($3/8$ under full 16K training vs. $1/8$ under T75 and $0/8$ under T50). `modernize-fortran-build` is highly sensitive to which continuations are retained: it keeps $7/8$ under T75 but collapses to $1/8$ under B75, suggesting that the recovery segments it depends on can be filtered out entirely.

Bottom-likelihood refinement trades repeatability for breadth. B75 solves the most tasks of any 16K model ($\mathrm{Rel}_{\geq 1} = 30$), including `eval-mteb`, `eval-mteb.hard`, and `decommissioning-service-with-sensitive-data`, which no one-stage or Top-selected model solves. B50 similarly adds `crack-7z-hash.hard` and `path-tracing`. These new successes are mostly isolated (1–2/8): relative to T75, B75 has fewer consistently solved tasks ($\mathrm{Rel}_{\geq 6}$: 11 vs. 14) and more fragile ones (13 vs. 9; Table 5). Low-likelihood continuations thus appear to carry rare, diverse behaviors that expand coverage, whereas high-likelihood continuations consolidate repeatable solving, consistent with the capacity- and budget-dependent effects of likelihood-based selection reported by Cuong et al. (2026).

E.3 How do training schedule, data quantity, and likelihood selection interact?

To disentangle the effects of the two-stage schedule, data quantity, and likelihood-based selection, we add two controls. First, we refine the 8K warm-start model at 12K and 16K using all eligible trajectories, isolating the effect of the two-stage schedule from filtering. Second, for each retention ratio $\rho \in \{25\%, 50\%, 75\%\}$, we uniformly sample the same number of eligible trajectories as the corresponding Top/Bottom-ρ setting, so comparisons among Top, Random, and Bottom are made at a fixed data budget. Table 10 supports the bias–complexity trade-off in Theorem 5.2. First, two-stage training with all eligible data improves over one-stage training from 111 ± 2.3 to 120 ± 2.4 successes at 12K and from 110 ± 2.7 to 123 ± 2.4 at 16K, indicating that the warm-start schedule itself provides a substantial benefit. Second, the Random controls reveal a horizon-dependent data-quantity effect. At 12K, increasing retention from 25% to 75% changes success only from 116 ± 2.5 to 118 ± 2.6, which is within the estimated run-to-run variation. At 16K, however, the corresponding increase is much larger, from 111 ± 2.8 to 122 ± 2.4, consistent with the finite-sample term $O\left(\sqrt{\mathcal{C}(H)/N_{\mathrm{sel}}}\right)$: as $\mathcal{C}(H)$ grows with horizon, using too few trajectories becomes increasingly costly. Finally, fixed-budget comparisons isolate selection quality. At 12K-25%, Top reaches 125 ± 2.2 successes, compared with 116 ± 2.5 for Random and 108 ± 2.8 for Bottom, a gap larger than the reported variation. At 16K-75%, Top similarly reaches 126 ± 2.1, versus 122 ± 2.4 for Random and 114 ± 2.7 for Bottom. In contrast, some settings show little separation, for example, at 16K-50%, Top, Random, and Bottom obtain 122 ± 2.4, 121 ± 2.5, and 120 ± 2.5 successes, respectively. Overall, the controls support a joint quality–quantity view: the two-stage schedule improves initialization, likelihood selection can provide additional gains when the separation exceeds evaluation variation, and retaining sufficient data becomes increasingly important at longer horizons.

E.4 Results on Other Benchmarks

The emergence–saturation pattern generalizes across benchmarks. As shown in Table 11 and Table 12, the one-stage results reproduce the same qualitative trend observed in the main experiments. On Terminal-Bench v2.0, successful attempts increase from 44 ± 2.6 at 4K to 65 ± 2.3 at 12K, but

Table 10: **Disentangling two-stage training, data quantity, and likelihood selection.** All two-stage models warm-start from 8K and refine at the target horizon on a random, highest- (Top), or lowest-likelihood (Bottom) $\rho\%$ of continuations. One-stage and Top/Bottom results are measured. Shaded : improves over one-stage training at the same target; **green**: best in column within each target.

Selection	ρ	Succ. ↑	Reliability ↑			Fragile ↓
			$\mathrm{Rel}_{\geq 4}$	$\mathrm{Rel}_{\geq 6}$	$\mathrm{Rel}_{\geq 8}$	
12K target						
One-stage (full)	—	111±2.3	13±0.8	9±0.6	6±0.5	12±0.8
Two-stage All	100%	120±2.4	16±0.8	10±0.6	6±0.5	9±0.7
Random		116±2.5	15±0.8	8±0.6	4±0.5	10±0.7
Top	25%	125±2.2	19±0.7	10±0.6	5±0.5	6±0.6
Bottom		108±2.8	13±0.8	7±0.6	4±0.5	13±0.8
Random		117±2.5	15±0.8	9±0.6	5±0.5	10±0.7
Top	50%	121±2.3	15±0.7	11±0.6	6±0.5	9±0.7
Bottom		112±2.6	15±0.8	8±0.6	3±0.4	9±0.7
Random		118±2.6	15±0.8	9±0.7	5±0.5	10±0.8
Top	75%	119±2.5	16±0.8	9±0.6	6±0.5	10±0.7
Bottom		117±2.7	14±0.8	9±0.7	5±0.5	14±0.8
16K target						
One-stage (full)	—	110±2.7	14±0.8	9±0.7	4±0.5	9±0.7
Two-stage All	100%	123±2.4	16±0.8	12±0.7	6±0.5	8±0.7
Random		111±2.8	14±0.8	9±0.7	5±0.5	10±0.7
Top	25%	114±2.6	14±0.7	9±0.6	5±0.5	10±0.7
Bottom		108±2.9	13±0.8	9±0.7	4±0.5	9±0.7
Random		121±2.5	15±0.8	11±0.7	6±0.5	9±0.7
Top	50%	122±2.4	15±0.7	11±0.6	6±0.5	9±0.7
Bottom		120±2.5	16±0.8	12±0.7	6±0.5	8±0.6
Random		122±2.4	16±0.8	12±0.7	6±0.5	9±0.8
Top	75%	126±2.1	15±0.7	14±0.6	7±0.5	9±0.7
Bottom		114±2.7	15±0.8	11±0.7	4±0.5	13±0.8

remain essentially unchanged at 16K (64 ± 2.6); similarly, on OpenThoughts-TBLite they increase from 96 ± 2.9 to 139 ± 2.4 before saturating at 137 ± 2.7. The reliability profile also exhibits the same redistribution beyond the effective horizon: from 12K to 16K, task coverage decreases ($17 \rightarrow 15$ on TB v2.0; $36 \rightarrow 32$ on TBLite), while mean success among solved tasks increases ($3.82 \rightarrow 4.27$; $3.86 \rightarrow 4.28$), $\mathrm{Rel}_{\geq 4}$ increases, and the number of fragile tasks decreases. Thus, across benchmarks, longer supervision initially broadens task solving, whereas supervision beyond the intermediate horizon primarily makes already-solvable tasks more repeatable rather than increasing aggregate reliability.

Selective refinement consistently improves long-horizon reliability. Table 11 and Table 12 further show that the benefit of selective refinement transfers across benchmarks. On Terminal-Bench v2.0, 12K Top-50 improves successful attempts from 65 ± 2.3 to 71 ± 2.3, while 16K Top-75 improves from 64 ± 2.6 to 73 ± 2.1. The same pattern is stronger on OpenThoughts-TBLite, where the corresponding improvements are $139\pm2.4 \rightarrow 151\pm2.3$ and $137\pm2.7 \rightarrow 155\pm2.2$. These gains are larger than the estimated run-to-run variation, supporting the main-paper observation that uniformly supervising all long-horizon continuations is unnecessary: selecting a subset of compatible continuations can yield higher reliability than full long-horizon training.

The effect of selection depends on both likelihood and retained data. As shown in Table 11 and Table 12, the cross-benchmark results also preserve the interaction between continuation likelihood and selection ratio observed in the main experiments. At 12K and $\rho = 50\%$, Top-likelihood selection consistently exceeds Bottom-likelihood selection in aggregate success (71 vs. 66 on TB v2.0 and 151 vs. 141 on TBLite), while also improving higher-order repeatability. However, the advantage

Table 11: **Reliability of one-stage and selective two-stage training on Terminal-Bench v2.0.** Two-stage models warm-start from 8K and refine at 12K or 16K on the highest- (**Top**) or lowest-likelihood (Bot.) $\rho\%$ of continuations. Values are reported with estimated run-to-run variation. Shaded : improves over one-stage training at the same horizon; **green**: best in row.

	One-stage				12K refinement (from 8K)						16K refinement (from 8K)					
	training horizon				ρ=25%		ρ=50%		ρ=75%		ρ=25%		ρ=50%		ρ=75%	
Metric	4K	8K	12K	16K	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.
Aggregate reliability																
Succ. attempts ↑	44±2.6	55±2.5	65±2.3	64±2.6	72±2.2	63±2.7	71±2.3	66±2.5	70±2.4	69±2.6	66±2.5	63±2.8	71±2.3	70±2.5	73±2.1	66±2.6
$\mathrm{Rel}_{\geq 1}$ ↑	13±0.7	14±0.7	17±0.6	15±0.7	15±0.7	16±0.8	16±0.7	16±0.8	17±0.7	18±0.8	16±0.7	14±0.8	16±0.7	15±0.8	15±0.7	17±0.8
$\bar{s}_+$ ↑	3.38±0.1	3.93±0.1	3.82±0.1	4.27±0.1	4.80±0.1	3.94±0.1	4.44±0.1	4.13±0.1	4.12±0.1	3.83±0.1	4.13±0.1	4.50±0.1	4.44±0.1	4.67±0.1	4.87±0.1	3.88±0.1
Repeatability																
$\mathrm{Rel}_{\geq 4}$ ↑	5±0.5	6±0.5	8±0.6	9±0.6	11±0.6	8±0.6	9±0.6	9±0.6	9±0.6	8±0.6	8±0.6	8±0.6	9±0.6	9±0.6	9±0.6	9±0.6
$\mathrm{Rel}_{\geq 6}$ ↑	2±0.4	4±0.5	5±0.5	5±0.5	6±0.5	4±0.5	6±0.5	5±0.5	5±0.5	5±0.5	5±0.5	5±0.5	6±0.5	7±0.6	8±0.6	6±0.5
$\mathrm{Rel}_{\geq 8}$ ↑	1±0.3	2±0.4	3±0.4	2±0.4	3±0.4	2±0.4	4±0.5	2±0.4	4±0.5	3±0.4	3±0.4	2±0.4	3±0.4	3±0.4	4±0.5	2±0.4
Fragile ($1 \leq s_i \leq 2$) ↓	7±0.6	6±0.6	7±0.7	5±0.6	4±0.5	8±0.7	5±0.6	5±0.6	6±0.6	8±0.7	6±0.6	5±0.6	5±0.6	5±0.6	5±0.6	8±0.7

Table 12: **Reliability of one-stage and selective two-stage training on OpenThoughts-TBLite.** Two-stage models warm-start from 8K and refine at 12K or 16K on the highest- (**Top**) or lowest-likelihood (Bot.) $\rho\%$ of continuations. Values are reported with estimated run-to-run variation. Shaded : improves over one-stage training at the same horizon; **green**: best in row.

	One-stage				12K refinement (from 8K)						16K refinement (from 8K)					
	training horizon				ρ=25%		ρ=50%		ρ=75%		ρ=25%		ρ=50%		ρ=75%	
Metric	4K	8K	12K	16K	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.	**Top**	Bot.
Aggregate reliability																
Succ. attempts ↑	96±2.9	118±2.7	139±2.4	137±2.7	153±2.2	135±2.8	151±2.3	141±2.6	149±2.5	147±2.7	141±2.6	135±2.9	151±2.4	149±2.5	155±2.2	141±2.7
$\mathrm{Rel}_{\geq 1}$ ↑	28±0.8	30±0.8	36±0.7	32±0.8	32±0.7	34±0.8	34±0.7	34±0.8	36±0.7	38±0.8	34±0.7	30±0.8	34±0.7	32±0.8	32±0.7	36±0.8
$\bar{s}_+$ ↑	3.43±0.1	3.93±0.1	3.86±0.1	4.28±0.1	4.78±0.1	3.97±0.1	4.44±0.1	4.15±0.1	4.14±0.1	3.87±0.1	4.15±0.1	4.50±0.1	4.44±0.1	4.66±0.1	4.84±0.1	3.92±0.1
Repeatability																
$\mathrm{Rel}_{\geq 4}$ ↑	11±0.7	13±0.7	17±0.8	19±0.8	23±0.7	17±0.8	19±0.7	19±0.8	19±0.8	17±0.8	17±0.7	17±0.8	19±0.7	19±0.8	19±0.7	19±0.8
$\mathrm{Rel}_{\geq 6}$ ↑	4±0.5	9±0.6	11±0.6	11±0.7	13±0.6	8±0.6	13±0.6	11±0.6	11±0.6	11±0.7	11±0.6	11±0.7	13±0.6	15±0.7	17±0.6	13±0.7
$\mathrm{Rel}_{\geq 8}$ ↑	2±0.4	4±0.5	6±0.5	4±0.5	6±0.5	4±0.5	8±0.5	4±0.5	8±0.5	6±0.5	6±0.5	4±0.5	6±0.5	6±0.5	9±0.5	4±0.5
Fragile ($1 \leq s_i \leq 2$) ↓	15±0.8	13±0.7	15±0.8	11±0.7	9±0.6	17±0.8	11±0.7	11±0.7	13±0.7	17±0.8	13±0.7	11±0.7	11±0.7	11±0.7	11±0.7	17±0.8

is not uniform across every metric or selection ratio: Bottom-likelihood subsets can retain broader task coverage in some settings, and the strongest 16K result emerges only when sufficient data are retained, with Top-75 achieving 73±2.1 successes on TB v2.0 and 155±2.2 on TBLite. This is consistent with the main conclusion that likelihood is a useful compatibility signal, but effective refinement must jointly balance continuation learnability and data sufficiency.

F Missing Proofs

In this section, we provide the detailed derivation of Theorem F.11 and establish several consequences for choosing the supervision horizon. In particular, Corollary F.14 characterizes when shorter-horizon supervision yields a tighter bound than full-horizon training, Corollary F.16 and Corollary F.18 characterize when an intermediate horizon minimizes a surrogate bound, and Corollary F.19 describes how this horizon changes with the amount of expert data.

Our analysis proceeds in three steps. First, we formalize the benefit of longer supervision through *temporal supervision bias*, which captures the full-horizon performance gap induced by truncating teacher trajectories. Second, we show that extending the supervision horizon exposes the learner to a richer set of decision histories, increasing the effective policy complexity and hence the finite-sample estimation burden. Third, we connect prefix-level estimation error to full-horizon performance and combine these two effects to obtain the bias–complexity bound in Theorem F.11.

We then use this result to answer three questions about the bound: when shorter-horizon supervision yields a tighter guarantee than full-horizon training, under what conditions an intermediate horizon minimizes a surrogate of the bound, and how this horizon shifts as the amount of expert data grows. These results characterize the guarantee rather than the true error, consistent with the scope stated below. Finally, we connect the decision-level analysis to the token-level supervision horizon used in our experiments.

Setup. We build on the decision-level imitation-learning formulation in Section 3. Let Π denote the student policy class and π^{T} the teacher policy. Let H denote the number of teacher decisions retained for supervision. We assume that N teacher trajectories are sampled independently from π^{T} and have lengths $T_i \leq T$. For $t > T_i$, we set $\ell \equiv 0$, so that L_H and $\widehat{L}_H$ are well defined for every

$H \le T$. For any $H \le T$, supervision is restricted to the first H decisions, while the policy continues to condition on the complete history h_t at each step.

Scope of the analysis. The analysis concerns imitation risk under teacher-visited histories, for an empirical risk minimizer over a finite policy class with bounded loss. For log-loss, boundedness requires clipping probabilities below at some $\epsilon > 0$, giving $M = \ell_{\max} \log(1/\epsilon)$ for actions of at most $\ell_{\max}$ tokens. Neural networks trained for a finite number of optimizer steps on a token-averaged objective (Equation 8) only approximate this setting. The resulting bound does not by itself imply that the true error, or task success under the agent's own rollouts, is non-monotone in H, and it makes no prediction about trajectory length or persistence. We therefore use it to identify two competing quantities and treat the behavioral findings of Section 5.1 as empirical.

Definition F.1 (Horizon-H imitation risk)**.** Let $\ell(\pi(\cdot \mid h_t), a_t)$ denote an action-level imitation loss satisfying

$$0 \le \ell(\pi(\cdot \mid h_t), a_t) \le M. \tag{12}$$

The population and empirical imitation risks under supervision horizon H are

$$L_H(\pi) = \mathbb{E}_{\tau \sim \pi^{\mathrm{T}}} \left[\frac{1}{H} \sum_{t=1}^{H} \ell(\pi(\cdot \mid h_t), a_t) \right], \qquad \widehat{L}_H(\pi) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{H} \sum_{t=1}^{H} \ell\Big(\pi(\cdot \mid h_t^{(i)}), a_t^{(i)}\Big), \tag{13}$$

where $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^{N}$ denotes the N teacher trajectories. Behavior cloning with horizon H learns

$$\widehat{\pi}_H \in \arg\min_{\pi \in \Pi} \widehat{L}_H(\pi). \tag{14}$$

Definition F.2 (Horizon-induced policy complexity)**.** Let

$$\mathcal{H}_{\le H} = \left\{ h_t : 1 \le t \le H, \ \Pr_{\tau \sim \pi^{\mathrm{T}}}(h_t) > 0 \right\} \tag{15}$$

denote the set of interaction histories reachable under the teacher policy within the first H decisions. Two policies $\pi, \pi' \in \Pi$ are equivalent at horizon H, written $\pi \equiv_H \pi'$, if

$$\pi(\cdot \mid h) = \pi'(\cdot \mid h), \qquad \forall h \in \mathcal{H}_{\le H}. \tag{16}$$

The corresponding quotient class and horizon-induced policy complexity are

$$\Pi_H = \Pi / \equiv_H, \qquad \mathcal{C}(H) = \log |\Pi_H|. \tag{17}$$

For infinite policy classes, the union bound in Lemma F.7 must be replaced by a covering or symmetrization argument, with $\mathcal{C}(H)$ replaced by the corresponding complexity of the loss class restricted to $\mathcal{H}_{\le H}$. We do not pursue this extension here.

Lemma F.3 (Policy complexity increases with supervision horizon)**.** *For any $H_1 < H_2$,*

$$\mathcal{C}(H_1) \le \mathcal{C}(H_2). \tag{18}$$

Proof. Since

$$\mathcal{H}_{\le H_1} \subseteq \mathcal{H}_{\le H_2}, \tag{19}$$

any two policies distinguishable within horizon H_1 remain distinguishable within horizon H_2. Thus, increasing the supervision horizon may split equivalence classes but cannot merge them, implying

$$|\Pi_{H_1}| \le |\Pi_{H_2}|, \qquad \mathcal{C}(H_1) \le \mathcal{C}(H_2). \tag{20}$$

□

Lemma F.3 formalizes that longer supervision exposes the learner to a richer set of distinguishable behaviors. This increase in complexity reflects the diversity of observed decision histories, not an increase in the number of model parameters.

Definition F.4 (Prefix-optimal set and pessimistic reference risk)**.** Let

$$L_H^\star := \min_{\pi \in \Pi} L_H(\pi), \qquad \Pi_H^\star := \arg\min_{\pi \in \Pi} L_H(\pi), \qquad \pi_T^\star \in \arg\min_{\pi \in \Pi} L_T(\pi), \tag{21}$$

and define the pessimistic full-horizon risk of the prefix-optimal set as

$$R_H := \sup_{\pi \in \Pi_H^\star} L_T(\pi). \tag{22}$$

Since $0 \le \ell \le M$, $R_H \le M$ is finite.

Supervision over the first H decisions constrains a policy only on histories in $\mathcal{H}_{\leq H}$ (Definition F.2). Policies that agree on these histories have identical prefix risk but may behave differently afterward, so $\Pi_H^\star$ generally contains policies with different full-horizon risks. R_H accounts for all of them rather than for a single tie-broken minimizer.

Definition F.5 (Temporal supervision bias)**.** We define the temporal supervision bias at horizon H (restated from Definition 5.1) as

$$B(H) := R_H - L_T(\pi_T^\star) = \sup_{\pi \in \Pi_H^\star} L_T(\pi) - L_T(\pi_T^\star). \tag{23}$$

Since $\Pi_H^\star$ is non-empty and $\pi_T^\star$ minimizes L_T over $\Pi \supseteq \Pi_H^\star$, we have $B(H) \geq 0$. Moreover, every $\pi \in \Pi_T^\star$ (the case $H = T$ of $\Pi_H^\star$) is a minimizer of L_T and therefore attains $L_T(\pi) = L_T(\pi_T^\star)$, so $R_T = L_T(\pi_T^\star)$ and $B(T) = 0$. Thus, $B(H)$ measures the worst-case excess full-horizon risk left undetermined by supervising only the first H decisions.

Assumption F.6 (Monotone temporal supervision bias)**.** For any $H_1 < H_2 \leq T$,

$$R_{H_2} \leq R_{H_1}, \qquad \text{equivalently} \qquad B(H_2) \leq B(H_1). \tag{24}$$

Lemma F.7 (Finite-sample estimation error)**.** *Assume that the N teacher trajectories are independently sampled and that $0 \leq \ell \leq M$. Then, with probability at least $1 - \delta$,*

$$L_H(\widehat{\pi}_H) - L_H^\star \leq M\sqrt{\frac{2\left[\mathcal{C}(H) + \log\frac{2}{\delta}\right]}{N}}. \tag{25}$$

Proof. Define the trajectory-level prefix loss as

$$Z_H(\pi;\tau) = \frac{1}{H}\sum_{t=1}^{H} \ell(\pi(\cdot \mid h_t), a_t), \qquad 0 \leq Z_H(\pi;\tau) \leq M. \tag{26}$$

Although decisions within a trajectory are dependent, the variables $Z_H(\pi;\tau^{(1)}), \ldots, Z_H(\pi;\tau^{(N)})$ are independent across teacher trajectories.

For a fixed policy π, Hoeffding's inequality gives

$$\Pr\left(\left|L_H(\pi) - \widehat{L}_H(\pi)\right| > \epsilon\right) \leq 2\exp\left(-\frac{2N\epsilon^2}{M^2}\right). \tag{27}$$

Both $L_H(\pi)$ and $\widehat{L}_H(\pi)$ depend on π only through its behavior on $\mathcal{H}_{\leq H}$, and are therefore constant on each equivalence class of Π_H. A union bound over the $|\Pi_H|$ classes thus yields

$$\Pr\left(\sup_{\pi \in \Pi}\left|L_H(\pi) - \widehat{L}_H(\pi)\right| > \epsilon\right) \leq 2|\Pi_H|\exp\left(-\frac{2N\epsilon^2}{M^2}\right). \tag{28}$$

Setting the right-hand side to δ and using $\mathcal{C}(H) = \log|\Pi_H|$ gives

$$\sup_{\pi \in \Pi}\left|L_H(\pi) - \widehat{L}_H(\pi)\right| \leq M\sqrt{\frac{\mathcal{C}(H) + \log(2/\delta)}{2N}}. \tag{29}$$

Fix any $\pi' \in \Pi_H^\star$. Since $\widehat{\pi}_H$ minimizes the empirical risk, $\widehat{L}_H(\widehat{\pi}_H) \leq \widehat{L}_H(\pi')$, and hence

$$\begin{aligned} L_H(\widehat{\pi}_H) - L_H^\star &= \left[L_H(\widehat{\pi}_H) - \widehat{L}_H(\widehat{\pi}_H)\right] + \left[\widehat{L}_H(\widehat{\pi}_H) - \widehat{L}_H(\pi')\right] + \left[\widehat{L}_H(\pi') - L_H(\pi')\right] \\ &\leq 2\sup_{\pi \in \Pi}\left|L_H(\pi) - \widehat{L}_H(\pi)\right|. \end{aligned} \tag{30}$$

Substituting the uniform bound above proves the result. □

Training controls performance only over the supervised prefix, whereas our goal is performance over the full task horizon. Note that the full-horizon risk L_T, unlike L_H, is *not* constant on the equivalence classes of Π_H: prefix-equivalent policies may differ after step H. The transfer coefficient below is therefore defined relative to the pessimistic reference R_H, which covers every prefix-optimal policy.

Definition F.8 (Prefix-to-full-horizon transfer coefficient)**.** For supervision horizon H, define (restated from (5))

$$\Gamma_H := \sup_{\pi \in \Pi:\, L_H(\pi) > L_H^\star} \frac{[L_T(\pi) - R_H]_+}{L_H(\pi) - L_H^\star}, \qquad [x]_+ := \max\{x, 0\}, \tag{31}$$

with $\Gamma_H := 0$ if no policy has $L_H(\pi) > L_H^\star$.

The coefficient Γ_H measures how much prefix-level excess risk can amplify when evaluated over the full horizon. We assume there exists a constant Γ such that

$$\Gamma_H \le \Gamma, \qquad \forall H \le T. \tag{32}$$

For a finite policy class, each Γ_H is a maximum over finitely many ratios and is therefore finite, but it can be arbitrarily large, and Γ enters the bound multiplicatively through C_1. The assumption is therefore substantive as a quantitative condition. Because supervision constrains a policy only on $\mathcal{H}_{\le H}$, a policy with small prefix excess risk may still behave very differently after step H, which inflates Γ_H. A moderate Γ thus requires the policy class to couple late-stage behavior to early-stage behavior, for example through parameter sharing across histories. Defining Γ_H relative to R_H, rather than to a single minimizer, ensures that the transfer inequality also holds for policies that tie on the prefix (Lemma F.9).

Lemma F.9 (Prefix-to-full-horizon transfer)**.** *For every $\pi \in \Pi$,*

$$L_T(\pi) - R_H \le \Gamma_H \left[L_H(\pi) - L_H^\star\right]. \tag{33}$$

Consequently, under $\Gamma_H \le \Gamma$, with probability at least $1 - \delta$,

$$L_T(\widehat{\pi}_H) - R_H \le \Gamma M \sqrt{\frac{2\left[\mathcal{C}(H) + \log\frac{2}{\delta}\right]}{N}}. \tag{34}$$

Proof. Since $L_H(\pi) \ge L_H^\star$ for every $\pi \in \Pi$, two cases are exhaustive.

Case 1: $L_H(\pi) = L_H^\star$. Then $\pi \in \Pi_H^\star$, so $L_T(\pi) \le R_H$ by definition of R_H. The left-hand side of (33) is non-positive and the right-hand side is zero, so the inequality holds.

Case 2: $L_H(\pi) > L_H^\star$. By definition of Γ_H,

$$L_T(\pi) - R_H \le [L_T(\pi) - R_H]_+ \le \Gamma_H \left[L_H(\pi) - L_H^\star\right]. \tag{35}$$

Applying (33) to $\pi = \widehat{\pi}_H$, then $\Gamma_H \le \Gamma$ and Lemma F.7, proves the second claim. □

Lemma F.10 (Error decomposition)**.** *For any supervision horizon H,*

$$L_T(\widehat{\pi}_H) - L_T(\pi_T^\star) = B(H) + \mathcal{E}_N(H), \qquad \mathcal{E}_N(H) := L_T(\widehat{\pi}_H) - R_H. \tag{36}$$

Proof. Adding and subtracting R_H gives

$$L_T(\widehat{\pi}_H) - L_T(\pi_T^\star) = \left[L_T(\widehat{\pi}_H) - R_H\right] + \left[R_H - L_T(\pi_T^\star)\right]. \tag{37}$$

The two terms are $\mathcal{E}_N(H)$ and $B(H)$, respectively. Note that $\mathcal{E}_N(H)$ may be negative, e.g., when $\widehat{\pi}_H$ is a prefix-optimal policy with better late-stage behavior than the worst one in $\Pi_H^\star$. □

Theorem F.11 (Detailed Bias–Complexity Bound)**.** *Consider the history-conditioned terminal-agent imitation-learning problem above. Assume that:*

1. *the N teacher trajectories are sampled independently;*
2. *the imitation loss is bounded, $0 \le \ell \le M$;*
3. *the policy class Π is finite, so that minimizers exist and $\mathcal{C}(H) = \log|\Pi_H| < \infty$;*
4. *the transfer coefficient satisfies $\Gamma_H \le \Gamma < \infty$.*

Then, for every $1 \le H \le T$, with probability at least $1-\delta$,

$$L_T(\widehat{\pi}_H) - L_T(\pi_T^\star) \le B(H) + C_1\sqrt{\frac{\mathcal{C}(H)+\log(2/\delta)}{N}}, \qquad C_1 = \sqrt{2}\,\Gamma M. \tag{38}$$

Proof. By Lemma F.10 and Lemma F.9,

$$\begin{aligned} L_T(\widehat{\pi}_H) - L_T(\pi_T^\star) &= B(H) + \mathcal{E}_N(H) \\ &\le B(H) + \Gamma M\sqrt{\frac{2\left[\mathcal{C}(H)+\log(2/\delta)\right]}{N}}. \end{aligned} \tag{39}$$

Substituting $C_1 = \sqrt{2}\,\Gamma M$ proves the result. □

For the remainder of this section, we denote the estimation term of the bound by

$$\varepsilon_N(H) := C_1\sqrt{\frac{\mathcal{C}(H)+\log(2/\delta)}{N}}, \tag{40}$$

which upper-bounds the estimation error $\mathcal{E}_N(H)$ of Lemma F.10 with probability at least $1-\delta$. By Lemma F.3, $\varepsilon_N(H)$ is non-decreasing in H. Note that Theorem F.11 does not require Assumption F.6; monotonicity is used only to interpret the bound as a trade-off.

Interpretation. Theorem F.11 identifies two competing effects of increasing the supervision horizon. The first term, $B(H)$, is the *temporal supervision bias*, the worst-case full-horizon risk left undetermined by truncating supervision at H. Under Assumption F.6, it is non-increasing in H, reflecting the additional information provided by later-stage teacher behavior. The second term, $\varepsilon_N(H)$, is the *finite-sample estimation term*. By Lemma F.3, it is non-decreasing in H, since longer horizons expose the learner to a richer set of behaviorally distinguishable histories.

The bound therefore reflects a bias–complexity trade-off: additional supervision tightens the guarantee when the reduction in temporal supervision bias outweighs the increase in the estimation term. This is consistent with the empirical observation that performance improves with horizon initially but shows diminishing returns beyond an intermediate regime, although the bound alone does not establish that the true error follows this pattern.

Definition F.12 (Bound-optimal horizon)**.** The bound-optimal horizon is

$$H_N^\star \in \arg\min_{1\le H\le T}\left\{B(H)+\varepsilon_N(H)\right\}. \tag{41}$$

It is the horizon at which the guarantee of Theorem F.11 is tightest, and it need not coincide with the empirically best horizon.

Corollary F.13 (Full-horizon supervision)**.** *At $H = T$, the temporal supervision bias vanishes and the bound becomes*

$$B(T) = 0, \qquad L_T(\widehat{\pi}_T) - L_T(\pi_T^\star) \le C_1\sqrt{\frac{\mathcal{C}(T)+\log(2/\delta)}{N}}. \tag{42}$$

Moreover, by Lemma F.3,

$$\mathcal{C}(T) = \max_{H\le T}\mathcal{C}(H). \tag{43}$$

Thus, full-horizon supervision removes temporal supervision bias but incurs the largest horizon-dependent estimation term.

Corollary F.14 (When shorter supervision is preferable)**.** *For any $H < T$, the horizon-H bound is tighter than the full-horizon bound if and only if*

$$B(H) < \varepsilon_N(T) - \varepsilon_N(H), \tag{44}$$

where ε_N is the estimation term in (40). In particular, since $B(H) \ge 0$, this requires $\mathcal{C}(H) < \mathcal{C}(T)$.

Proof. By Theorem F.11 and Corollary F.13, the horizon-H and full-horizon bounds are

$$B(H)+\varepsilon_N(H), \qquad B(T)+\varepsilon_N(T) = \varepsilon_N(T), \tag{45}$$

respectively. Hence, horizon H gives a tighter bound if and only if

$$B(H)+\varepsilon_N(H) < \varepsilon_N(T), \tag{46}$$

which is equivalent to the stated condition. The second claim follows because ε_N is strictly increasing in $\mathcal{C}(H)$, so $\varepsilon_N(T) - \varepsilon_N(H) > 0$ requires $\mathcal{C}(H) < \mathcal{C}(T)$. □

To characterize diminishing returns, we impose an additional assumption on the temporal benefit contributed by later stages of a trajectory.

Assumption F.15 (Diminishing temporal novelty)**.** There exists a non-increasing function $\beta(H)$ such that

$$B(H) \le \beta(H), \qquad \Delta_\beta(H) := \beta(H) - \beta(H+1), \qquad \Delta_\beta(H+1) \le \Delta_\beta(H). \tag{47}$$

A useful special case is geometric decay, $\beta(H) = A\rho^H$ with $A > 0$ and $0 < \rho < 1$, for which $\Delta_\beta(H) = A(1-\rho)\rho^H$. Under Assumption F.15, the bound of Theorem F.11 is itself bounded by the surrogate

$$U_N(H) := \beta(H) + \varepsilon_N(H), \qquad \Delta_\varepsilon(H) := \varepsilon_N(H+1) - \varepsilon_N(H) \ge 0. \tag{48}$$

Corollary F.16 (Marginal characterization of the sweet spot)**.** *For every $H < T$,*

$$U_N(H+1) < U_N(H) \iff \Delta_\beta(H) > \Delta_\varepsilon(H). \tag{49}$$

Thus, extending the supervision horizon tightens the surrogate bound only while the marginal reduction in temporal supervision bias exceeds the marginal increase in the estimation term.

Proof. By definition, $U_N(H+1) - U_N(H) = -\Delta_\beta(H) + \Delta_\varepsilon(H)$. □

Assumption F.17 (Single crossing)**.** There exists $H_0 \in \{1, \dots, T\}$ such that $\Delta_\beta(H) > \Delta_\varepsilon(H)$ for all $H < H_0$, and $\Delta_\beta(H) \le \Delta_\varepsilon(H)$ for all $H_0 \le H < T$.

Corollary F.18 (Intermediate surrogate-optimal horizon)**.** *Under Assumption F.17, $H_0 \in \arg\min_{1 \le H \le T} U_N(H)$. Moreover, $H_0 < T$ if and only if $\Delta_\beta(T-1) \le \Delta_\varepsilon(T-1)$.*

Proof. By Corollary F.16, U_N is strictly decreasing on $\{1, \dots, H_0\}$ and non-decreasing on $\{H_0, \dots, T\}$, so H_0 is a minimizer. The second claim follows from the definition of H_0. □

Corollary F.19 (Effect of additional expert data)**.** *For a fixed horizon H, $\varepsilon_N(H) = O\big(\sqrt{\mathcal{C}(H)/N}\big) \to 0$ as $N \to \infty$. Moreover, let $H_0(N)$ denote the crossing point of Assumption F.17 at sample size N. If the assumption holds at both N and $N' > N$, then $H_0(N') \ge H_0(N)$.*

Proof. The first claim is immediate from (40). For the second, write $c_\delta = \log(2/\delta)$ and $\Delta_\varepsilon^{(N)}$ for Δ_ε at sample size N. Then

$$\Delta_\varepsilon^{(N)}(H) = \frac{C_1}{\sqrt{N}} \left[\sqrt{\mathcal{C}(H+1) + c_\delta} - \sqrt{\mathcal{C}(H) + c_\delta} \right], \tag{50}$$

where the bracket is non-negative by Lemma F.3 and independent of N. Hence $\Delta_\varepsilon^{(N)}(H)$ is non-increasing in N, while $\Delta_\beta(H)$ does not depend on N. It follows that for every $H < H_0(N)$,

$$\Delta_\beta(H) > \Delta_\varepsilon^{(N)}(H) \ge \Delta_\varepsilon^{(N')}(H). \tag{51}$$

Suppose, for contradiction, that $H_0(N') < H_0(N)$. Applying Assumption F.17 at N' with $H = H_0(N')$ gives $\Delta_\beta(H_0(N')) \le \Delta_\varepsilon^{(N')}(H_0(N'))$, which contradicts the display above since $H_0(N') < H_0(N)$. Therefore $H_0(N') \ge H_0(N)$. □

Thus, more expert data weakens the statistical penalty of longer horizons and moves the surrogate-optimal horizon weakly later.

Connection to token-level supervision. The analysis above is expressed in terms of a decision horizon H, while our implementation controls a token budget K. As defined in Section 3.3, K induces the trajectory-dependent decision horizon

$$H_K(\tau) = \max\{t : n_t(\tau) \le K\}, \qquad K_1 < K_2 \Longrightarrow H_{K_1}(\tau) \le H_{K_2}(\tau), \tag{52}$$

where $n_t(\tau)$ is the cumulative number of tokens through action a_t. Thus, increasing K monotonically extends the supervised portion of every trajectory. Two differences from the analysis remain. First, $H_K(\tau)$ varies across trajectories, whereas the analysis uses a common H. Second, an action that straddles the budget is partially supervised, which $H_K(\tau)$ ignores. The analysis should therefore be read as describing the qualitative effect of extending supervision, not as a quantitative model of token-level truncation.