MULTIPLE TURNS, SHORT HORIZONS: MEASURING EFFECTIVE-HORIZON COMPRESSION IN JAILBREAKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-turn jailbreaks use feedback from a victim large language model to adapt their queries over several turns. However, once an effective trajectory has been found, it is unclear how much of the earlier query history is still needed. In this work, we study this by removing early attacker queries and running the remaining queries again in a new conversation while regenerating the victim's responses. We define the *effective horizon* as the shortest remaining query sequence that recovers the same attack progress. Empirically, we find that roughly half of the original query sequence is sufficient on average to recover the same attack progress. We call this phenomenon *effective-horizon compression*. We show that this shorter history can recover the best point reached during an unsuccessful attack and provide a better starting point for continuing the attack. It also helps identify which earlier queries should receive less learning signal during attacker training, and downweighting those queries improves performance. We further find that short reward-propagation windows preserve calibrated defense performance close to full-trajectory returns. These results show that multi-turn attacks can contain more queries than are needed to recover the progress reached later in the interaction.

1 INTRODUCTION

Large language models (LLMs) are vulnerable to jailbreak attacks that can make them produce harmful responses despite safety alignment (Zou et al., 2023; Shen et al., 2024; Chao et al., 2025). Multi-turn jailbreaks are especially effective because each turn gives the attacker information that can steer the queries that follow (Russinovich et al., 2025; Yang et al., 2024; Ren et al., 2025; Rahman et al., 2025). However, this adaptivity can make the attack slow and costly, since the attacker may need several rounds of interaction before finding a query sequence that works.

Multi-turn attacks can be viewed as an adaptive search over possible attack trajectories. The attacker uses feedback from each interaction to decide how to proceed, often taking several steps before reaching an effective query sequence. However, once such a trajectory has been found, some of its earlier queries may no longer be needed. A shorter sequence of the later queries may be enough to recover the same attack progress. We therefore ask:

How much prior query history is needed to recover the same attack progress at a later turn?

To study this question, we remove earlier queries from successful attack trajectories and test whether the remaining queries can recover the same attack progress. As illustrated in Figure 1, we take the last few attacker queries from an existing trajectory and run them again in a new conversation. The attacker queries are kept exactly the same, while the victim generates new responses to them. If this shorter sequence reaches the same level of attack progress as the original trajectory, within a small tolerance, then the removed queries were not needed for the retained queries to remain effective. We define the *effective horizon* as the length of the shortest sequence that satisfies this condition.

We find that the effective horizon is often much shorter than the original trajectory generated by either X-Teaming (Rahman et al., 2025) or TROJail (Xiong et al., 2026). Across our evaluated settings, roughly half of the original query sequence is sufficient on average to recover the same attack progress. We refer to this reduction as *effective-horizon compression*. An attack trajectory can therefore often be shortened substantially while remaining similarly effective.

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.

This suggests that the full sequence of queries used to produce an attack may contain more history than is ultimately needed for the attack to remain effective.

This differs from shortening an attack by rewriting it. M2S (Ha et al., 2025), for example, uses a multi-turn dialogue to construct a new single-turn jailbreak prompt. Such a prompt can incorporate information from the entire dialogue. We leave the original attacker queries unchanged and only remove queries that came before them. The effective horizon therefore measures how much of the original query sequence is still needed to recover the same attack progress.

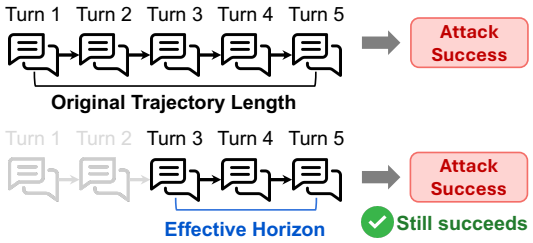


Figure 1: A later point in a multi-turn attack may require less query history than the full trajectory that led to it. We measure this by removing the earliest attacker queries and running the remaining queries again in a new conversation, while the victim generates new responses. The *effective horizon* is the shortest remaining query sequence that recovers the same attack progress reached at the target turn. In this example, the first two queries can be removed and the later queries still recover the same progress.

Effective-horizon compression also gives us a way to make better use of unsuccessful attacks. A failed trajectory may have made substantial progress at an earlier turn before later queries moved the attack in a worse direction. Rather than continuing from the end of such a trajectory, we return to its highest-scoring turn. We then use the effective horizon to keep only the shortest query history needed to recover the progress reached at that turn, and continue the attack from there. We call this *Effective-Horizon Context Recovery* (EH-CR). Across X-Teaming (Rahman et al., 2025) and ActorBreaker (Ren et al., 2025), EH-CR improves attack continuation compared with both keeping the complete failed trajectory and returning to the highest-scoring turn with its full history.

Effective-horizon compression also suggests a different way to train multi-turn attackers. When an attack succeeds, the training algorithm must decide which of the queries in the trajectory should be reinforced for producing that outcome. TROJail (Xiong et al., 2026) uses the final attack outcome as a learning signal for every attacker query in the trajectory. If the same attack progress can be recovered from only the last few queries, however, much earlier queries may not deserve the same learning signal. This does not mean that the earlier queries were useless, since they may have helped the attacker discover what to ask later. It does suggest that their contribution is different. We therefore introduce *Effective-Horizon Credit Assignment* (EH-CA), which keeps the full learning signal for queries within the effective horizon and reduces it for earlier queries. Across nine victim-model and benchmark pairs, this partial downweighting improves ASR@3 by 6.9 points on average over TROJail's original credit assignment.

We ask whether the same idea also matters for defense training. TurnGate (Shen et al., 2026) assigns a reward to each PASS or BLOCK decision, but trains an earlier decision using both its own reward and the rewards obtained at later turns. In the original method, this learning signal extends through the entire remaining conversation. We test whether such long-range reward propagation is actually necessary by limiting each decision to rewards from only a short number of subsequent turns. After calibrating the decision threshold, performance changes very little across these different reward windows. In particular, $\mathrm{F1}_{\phi_2}$ has a standard deviation of only 0.16%, and even using only the reward from the current turn performs nearly the same as using rewards from the full remaining trajectory. This suggests that a long interaction does not necessarily require learning signals to propagate across the entire conversation. Together with our attacker-training results, it shows that the useful range of a training signal can be much shorter than the trajectory in which it was generated.

Our contributions are:

- **Later attack progress often requires only part of the query history.** We introduce the *effective horizon*, the shortest sequence of later attacker queries that recovers the same attack progress as the original trajectory. Across X-Teaming and TROJail, roughly half of the original query sequence is sufficient on average. We call this *effective-horizon compression* (section 3).
- **Shorter histories can provide better starting points for continuing an attack.** An unsuccessful attack may reach a promising point and then move away from it. EH-CR returns to the highest-

scoring turn and keeps only the query history needed to recover that progress. This improves continuation over the failed trajectory and the full history leading to the same turn (section 4).

- **Learning signals need not extend across the full interaction.** *Effective-Horizon Credit Assignment* (EH-CA) reduces the learning signal given to queries before the effective horizon and improves attacker training across all nine evaluated settings. In defense training, calibrated performance also changes little when rewards are propagated over much shorter ranges. The useful range of a training signal can therefore be shorter than the trajectory that produced it (section 5).

2 Related Work

Our work relates to two lines of research on multi-turn jailbreaks: methods that construct attacks over several turns and defenses that use information across a conversation. We study how much of the earlier query history remains necessary once later attack queries have been produced.

Multi-turn Jailbreak Attacks. Search-based attacks generate queries during the interaction and adapt them to the victim's responses. Crescendo (Russinovich et al., 2025), FITD (Weng et al., 2025), and CoA (Yang et al., 2024) progressively develop an attack across turns, while Actor-Breaker (Ren et al., 2025), X-Teaming (Rahman et al., 2025), Trojan Knowledge (Wei et al., 2025), and PLAGUE (Bhuiya et al., 2025) incorporate planning, exploration, or coordination among multiple agents. Training-based methods instead learn policies for generating multi-turn attacks from supervised or reinforcement-learning signals (Zhao & Zhang, 2025; Xiong et al., 2026; Feng et al., 2026). M2S (Ha et al., 2025) takes a different approach by rewriting a multi-turn dialogue into a new single-turn jailbreak prompt. Our analysis keeps the original attacker queries unchanged and measures how much of their preceding query history can be removed.

Multi-turn Jailbreak Defenses. Multi-turn defenses use conversation history to detect harmful intent that may emerge only over several turns. Existing methods reason over the accumulated dialogue (Kuo et al., 2025), maintain information across turns using graph or recurrent representations (Huang et al., 2025; Albrethsen et al., 2026), or accumulate evidence of risk over time (Ma et al., 2026). D-Judge modifies victim responses to interrupt attack progression (Gong et al., 2026), while TRACE (Miah et al., 2026) and TurnGate (Shen et al., 2026) use trajectory-level reasoning or learned policies to decide when to block an interaction. Our defense experiments leave the observed conversation unchanged and instead study how far training rewards need to propagate across turns.

3 On Query-History Dependence in Multi-Turn Jailbreaks

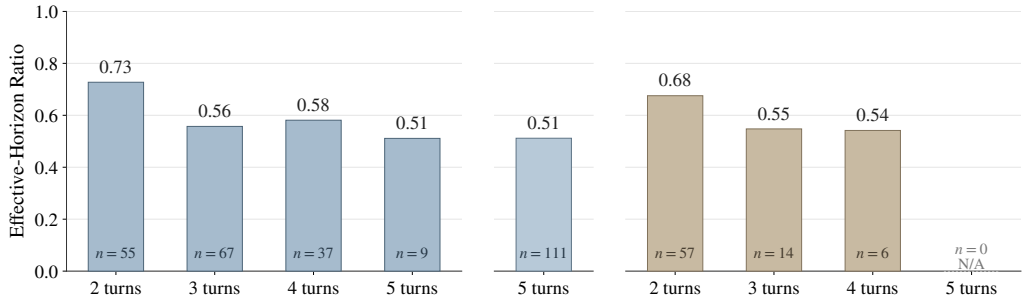
We aim to understand how much prior query history is needed to recover the attack progress reached at a later turn. We study this by taking an existing attack trajectory and selecting a target turn. We then remove attacker queries from the beginning of the sequence and run the remaining queries again in a new conversation, allowing the victim to generate new responses. If the shorter sequence reaches about the same attack score as the original query sequence, then the removed queries were not needed. The shortest query sequence that satisfies this condition defines the *effective horizon*.

3.1 Effective Horizon

We now make precise what we mean by the *effective horizon*. For any point in an attack trajectory, we ask how many of the most recent attacker queries are sufficient to recover the progress reached at that point. Consider a trajectory with attacker query sequence $\mathbf{q}_{1:T} = (q_1, \ldots, q_T)$, and let j denote the turn whose attack progress we want to recover. Let v_j be the attack score reached at that turn. For a candidate length k, we remove the earlier queries and run only the last k queries, $\mathbf{q}_{j-k+1:j}$, in a new conversation. The victim generates new responses, producing an attack score $\widetilde{v}_{j,k}$.

We say that a shorter sequence recovers the original attack progress if $\widetilde{v}_{j,k} \geq v_j - \epsilon$, where $\epsilon \geq 0$ is the maximum score decrease we allow. The value of ϵ depends on the scoring function and is specified for each experiment. We only remove queries from the beginning of the sequence, so each candidate is a contiguous suffix of the original query history. The *effective horizon* is the length of the shortest such suffix that recovers the original progress:

$$H_{\text{eff}}(\mathbf{q}_{1:j}) = \min \{k \in \{1, \ldots, j\} \mid \widetilde{v}_{j,k} \geq v_j - \epsilon\}. \tag{1}$$



(a) X-Teaming successful trajectories from 200 random behaviors

(b) 5-turn successes from 1,500 behaviors

(c) TROJail successful trajectories from 200 random behaviors

Figure 2: The attack progress reached by successful multi-turn attacks can often be recovered from only about half of their original query history. We measure this with the EHR, which divides the shortest query suffix that recovers the original attack progress by the full trajectory length. Across both X-Teaming and TROJail, trajectories of three or more turns have average EHRs close to 0.5–0.6. The larger X-Teaming evaluation in (b) confirms this pattern, with five-turn attacks reaching an average EHR of 0.512 over 111 successful trajectories. Panels (a) and (c) use 200 WildJailbreak behaviors, while (b) uses 1,500; n denotes the number of successful trajectories.

The effective horizon therefore tells us how much recent query history is sufficient to recover the progress reached at turn j. If $H_{\text{eff}} = 2$, for example, the last two attacker queries are sufficient and all earlier queries can be removed. If no tested suffix recovers the original progress, the effective horizon is undefined for that target turn. This definition isolates dependence on the preceding query history. We do not rewrite the retained queries or remove arbitrary turns; we only remove a prefix and regenerate the victim responses. A short effective horizon therefore means that the later queries can recover the target progress with little of the earlier query history still present. It does not mean that the removed turns were unnecessary for discovering those later queries. To compare trajectories of different lengths, we use the *effective-horizon ratio*, $\text{EHR} = H_{\text{eff}}/j$. An EHR of 1 means that the full query history is needed under our replay criterion, while smaller values indicate greater compression. We refer to this reduction as *effective-horizon compression*. The EHR therefore lets us compare how much query history remains necessary across attacks of different lengths.

3.2 Effective-Horizon Compression Across Attack Paradigms

We next test whether effective-horizon compression appears across different multi-turn attack methods. For successful trajectories from X-Teaming (Rahman et al., 2025) and TROJail (Xiong et al., 2026), we measure the effective horizon of the final turn. Both are evaluated on 200 WildJailbreak behaviors (Jiang et al., 2024) with a maximum of five turns. See experimental details in Appendix A.1. Figure 2(a) and (c) show a similar pattern for both attacks. For two-turn trajectories, the average EHR is 0.73 for X-Teaming and 0.68 for TROJail. Since a two-turn trajectory can only have an EHR of 0.5 or 1, these values mean that more than half of the successful attacks recover their original progress using only the final query. For longer trajectories, the average EHR remains around 0.5–0.6. This means that, even when an attack unfolds over several turns, only about half of the query history is needed on average to recover the progress reached at the end. The removed turns may still be necessary for current methods to discover the later queries. Their redundancy after the fact nevertheless suggests that there may be shorter paths to the same attack progress, and motivates future methods that discover these shorter paths more directly. Appendix B provides a case study.

Because the initial sample contains few successful five-turn X-Teaming trajectories, we repeat that setting with 1,500 behaviors. The resulting 111 successful attacks have an average EHR of 0.51, closely matching the smaller evaluation in Figure 2(b). The same effect also appears before an attack succeeds. Figure 3 measures the effective horizon at the highest-scoring point reached by unsuccessful X-Teaming trajectories. Across supported trajectory lengths and score levels, these intermediate states can often be recovered from only part of the preceding query history. In the larger evaluation, for example, trajectories reaching score 4 decrease from an average EHR of 0.729 at two turns to 0.533 at five turns. Effective-horizon compression therefore applies not only to

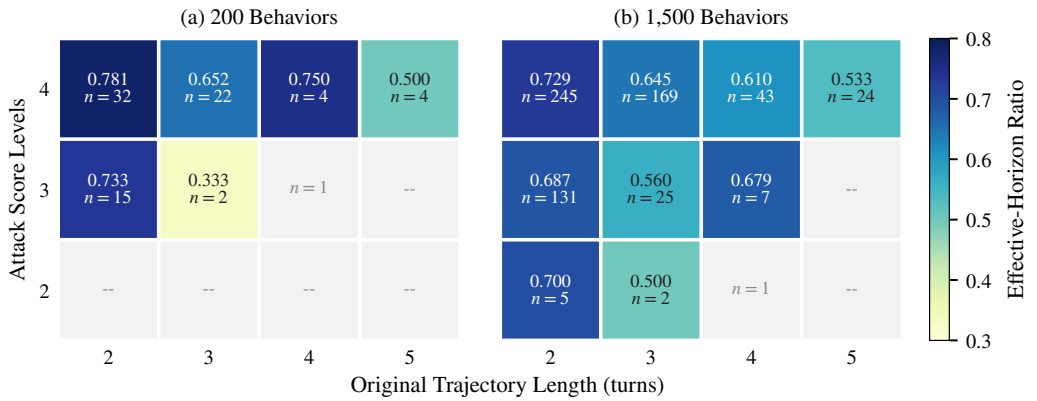


Figure 3: Effective-horizon compression also appears in unsuccessful attacks. We truncate each X-Teaming trajectory at the first turn that reaches its highest attack score and measure how much query history is needed to recover that progress. Most supported groups have EHR below 1, including high-scoring intermediate states. In the larger 1,500-behavior evaluation, score-4 trajectories decrease from an average EHR of 0.729 at two turns to 0.533 at five turns. This shows that attack progress can be recovered from a shorter history even before the attack succeeds.

successful attacks, but also to promising intermediate points that an attacker may want to return to. This observation motivates the continuation method in the next section.

4 Context Recovery in Search-Based Attacks

In the previous section, we showed that the progress reached by a multi-turn attack can often be recovered from much less query history than the full trajectory. We now ask whether this shorter history can help continue attacks that have not yet succeeded. A failed trajectory may have reached substantial progress at an earlier turn before later queries moved it in a worse direction. Returning to that earlier point removes those later failed turns, but still keeps all of the query history that led there. If only part of that history is needed to recover the same progress, we can restart the attack from a shorter context. This gives the attacker a better place to continue searching, using its remaining turns from a point that preserves the best progress reached so far without carrying unnecessary history.

Effective-Horizon Context Recovery (EH-CR). Given an unsuccessful trajectory, EH-CR first returns to the turn with the highest attack score,

$$j^\star = \arg\max_{1 \le j \le T} v_j. \tag{2}$$

We then apply Equation 1 to the query history up to $j^\star$ and keep only the shortest suffix that recovers the progress reached there. We run this shorter sequence again in a new conversation and continue the attack from the resulting response. If no shorter suffix recovers the same progress, we keep the full prefix up to $j^\star$. EH-CR removes two kinds of unnecessary history. It first discards the turns that came after the highest-scoring point, since those turns moved the attack away from its best progress. It then shortens the history leading up to that point by removing earlier queries that are no longer needed to recover it. The attack therefore restarts from its best recovered point using the shortest history that can reproduce that progress.

Does shorter history improve continuation? We test this with X-Teaming and ActorBreaker on 159 HarmBench behaviors. Each attack first runs for five turns, and unsuccessful trajectories receive five additional turns. We compare continuing from the full failed trajectory, returning to the highest-scoring turn with its full history (*Peak Context*), and returning to that same turn using only its effective suffix (EH-CR). See experimental details in Appendix A.2. Table 1 shows that EH-CR gives the largest improvement for every attack–victim pair. For X-Teaming, it improves ASR by 17.4 points on average, compared with 10.5 for Peak Context. For ActorBreaker, the gains are 19.2 and 14.2 points, respectively. The comparison with Peak Context is especially informative because both methods restart from the same point. The only difference is how much earlier history they

Table 1: Attack continuation after five-turn initial trajectories. Each unsuccessful attack receives five additional turns. Full continues from the complete trajectory, Peak returns to the highest-scoring turn with its full prefix, and EH-CR returns to the same turn using only its effective suffix. Entries report ΔASR, with final ASR in parentheses. Best values are shown in bold. EH-CR performs best for every attack–victim pair, showing that the shorter history is not only sufficient to recover progress but can provide a better continuation context despite using less of the original history.

Victim Model	X-Teaming				ActorBreaker			
	Base ASR	ΔASR from Continuation ↑			Base ASR	ΔASR from Continuation ↑		
		Full	Peak	EH-CR		Full	Peak	EH-CR
Qwen3-30B-A3B	67.9	+13.2 (81.1)	+15.7 (83.6)	**+22.6** (90.5)	37.1	+20.8 (57.9)	+20.1 (57.2)	**+22.0** (59.1)
Llama-3.3-70B	80.5	+10.1 (90.6)	+11.3 (91.8)	**+15.1** (95.6)	32.7	+22.0 (54.7)	+23.9 (56.6)	**+27.7** (60.4)
Muse-Glimmer-30B	15.1	+3.1 (18.2)	+3.1 (18.2)	**+10.7** (25.8)	13.2	+10.7 (23.9)	+8.8 (22.0)	**+13.8** (27.0)
DeepSeek-V4.1-Flash	23.9	+10.1 (34.0)	+13.2 (37.1)	**+18.9** (42.8)	11.3	+4.4 (15.7)	+5.7 (17.0)	**+11.3** (22.6)
GPT-5.6 Luna	13.8	+8.2 (22.0)	+11.3 (25.1)	**+20.1** (33.9)	8.2	+6.9 (15.1)	+6.9 (15.1)	**+16.4** (24.6)
Gemini 3.8 Flash	46.5	+8.2 (54.7)	+8.2 (54.7)	**+17.0** (63.5)	26.4	+20.1 (46.5)	+19.5 (45.9)	**+23.9** (50.3)
Avg.	-	+8.8	+10.5	**+17.4**	-	+14.2	+14.2	**+19.2**

retain. This isolates the effect of shorter retained history. The advantage of EH-CR therefore shows that history which is unnecessary for recovering attack progress can also hinder attack continuation.

This pattern also holds when we vary how much interaction occurs before continuation. As shown in Figure 4, EH-CR achieves the highest final ASR for initial trajectories of two, three, and four turns, improving over Full Context by 8.2 points on average. We also vary the number of turns available for continuation. Figure 5 shows that EH-CR already improves final ASR after one or two additional turns, and remains more effective as the continuation budget increases. Together, these results show that the effective horizon is useful beyond measurement. The shorter history identified by replay is not only sufficient to recover earlier attack progress; it can provide a better starting point for continuing the search. EH-CR is intended as a context-selection method rather than a way to reduce query count; query-cost results are reported in Appendix C.

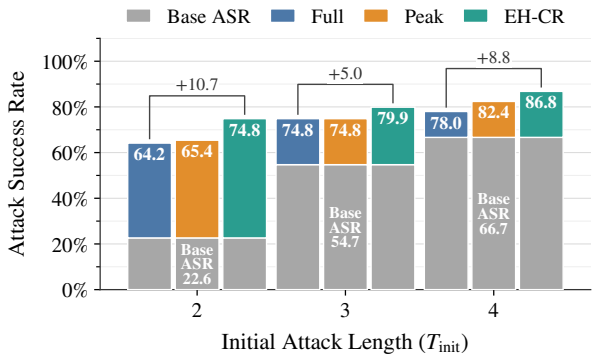


Figure 4: EH-CR consistently improves attack continuation across initial lengths. Each method receives five additional turns after an initial attack of $T_{\text{init}} \in \{2, 3, 4\}$. EH-CR consistently achieves the highest final ASR, improving over Full Context by 8.2 points on average and showing that recovering a shorter effective history remains useful even as the amount of prior interaction changes.

5 Effective Horizon and Temporal Credit Assignment

The previous sections showed that attack progress can often be recovered from much less query history than the full trajectory, and that this shorter history can provide a better continuation context. We now ask whether the same idea should change how learning signals are assigned during training. When an attack succeeds, a training method must decide how much that final outcome should reinforce each earlier query. If the same progress can be recovered from only the last few queries, giving every turn the same learning signal may be too coarse. Earlier queries may still have helped the attacker discover what to ask later, so we do not assume that their contribution is zero.

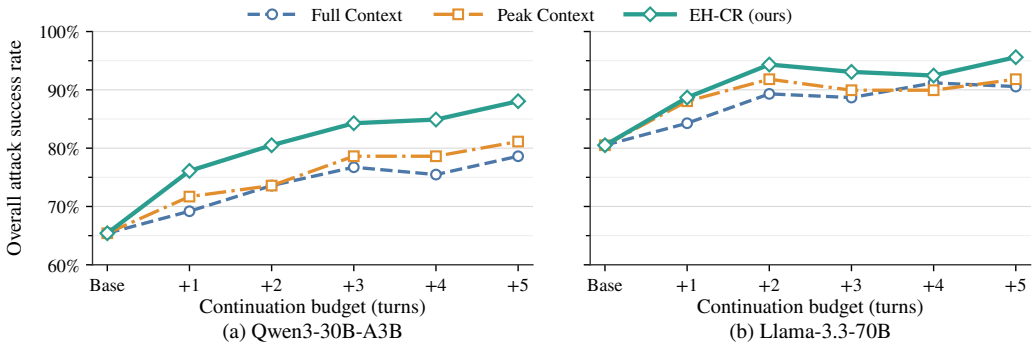


Figure 5: EH-CR remains more effective across continuation budgets. Starting from 5-turn initial trajectories, we vary 1–5 additional turns on 159 HarmBench behaviors for Qwen3-30B-A3B and Llama-3.3-70B. EH-CR achieves higher final ASR than Full and Peak for both victims at every continuation budget, with the advantage already appearing after a single additional turn.

We therefore study whether they should receive less credit for the final outcome, and ask the related question in defense training of how far turn-level rewards need to influence earlier decisions.

5.1 Outcome Credit in Attacker Training

Effective-Horizon Credit Assignment (EH-CA). TROJail (Xiong et al., 2026) provides a direct way to test this idea because it uses the final attack outcome as a learning signal for every attacker query in the trajectory. An early query therefore receives the same outcome signal as a query near the end of the attack. EH-CA changes only this assignment. Let $j^\star = T - H_{\text{eff}}(\mathbf{q}_{1:T}) + 1$ denote the first query within the effective horizon. We keep the original outcome signal for queries from $j^\star$ onward and reduce it for earlier queries:

$$A_i^{\text{out}} = \begin{cases} \beta A^{\text{traj}}, & i < j^\star, \\ A^{\text{traj}}, & i \geq j^\star, \end{cases} \tag{3}$$

where $\beta \in [0, 1]$ controls how much of the final-outcome signal earlier queries retain. Thus, $\beta = 1$ recovers TROJail's original assignment, while $\beta = 0$ removes this signal entirely before the effective horizon. All other components of TROJail training remain unchanged. See details in Appendix A.3.

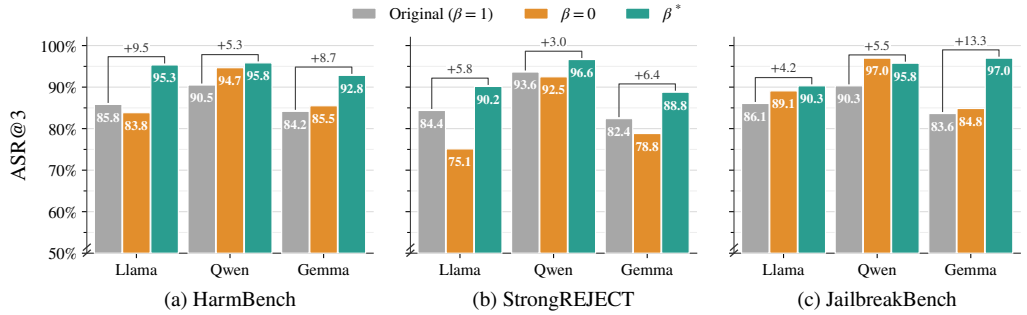
We follow the TROJail setup and evaluate three victims on HarmBench, StrongREJECT, and JailbreakBench, varying only β. Figure 6 shows that completely removing the earlier learning signal is too aggressive: $\beta = 0$ improves ASR@3 (success within three attempts) in five of nine victim–benchmark pairs but hurts the other four. In contrast, partial downweighting is consistently beneficial. Selecting $\beta \in \{0.1, 0.2, 0.3\}$ on HarmBench for each victim improves ASR@3 in all pairs, by 6.9 points on average over the original setting. This result helps explain why the effective horizon is useful for credit assignment. Queries before the effective horizon may still help attackers discover what to ask later, so removing their signal entirely can lose useful information. Meanwhile, giving them the same outcome signal as later queries is unnecessarily coarse. In our experiments, retaining a smaller signal for earlier queries gives the best of both. Appendix D reports detailed results.

5.2 Reward Propagation in Defense Training

We next ask whether the same distinction appears in defense training. TurnGate (Shen et al., 2026) makes a Pass or Block decision at each turn and receives a process reward for that decision. Its training signal for an earlier decision, however, includes rewards from all later turns. With TurnGate's choice of $\gamma = \lambda = 1$, the advantage at turn i is

$$A_i = \sum_{j=i}^{T} r_j, \tag{4}$$

where r_j is the reward at turn j. This means that the decision at turn i is trained using its own reward together with every reward that comes after it. For example, in a five-turn conversation, the decision



(a) HarmBench (b) StrongREJECT (c) JailbreakBench

Figure 6: Reducing outcome credit before the effective horizon improves attacker training. TROJail assigns the same outcome signal to every query ($\beta = 1$), while EH-CA scales this signal by β for queries before the effective horizon. Completely removing the earlier signal ($\beta = 0$) is not consistently beneficial, but partial retention improves ASR@3 across all pairs. $\beta^\star$ denotes the best value in $\{0.1, 0.2, 0.3\}$, selected on HarmBench for each victim and transferred to the other two benchmarks. Across all pairs, $\beta^\star$ improves ASR@3 by 6.9 points on average.

at turn 1 receives a learning signal based on rewards from all five turns, while the decision at turn 5 uses only its own reward. Rewards can therefore influence decisions made many turns earlier.

Rather than measuring a separate effective horizon for every defense decision, we test the underlying question directly by limiting how far later rewards can influence an earlier decision. To this end, we replace TurnGate's original training signal, which sums the reward at turn i with all rewards from later turns, with a truncated version:

$$A_i^{(K)} = \sum_{j=i}^{\min(i+K,T)} r_j. \tag{5}$$

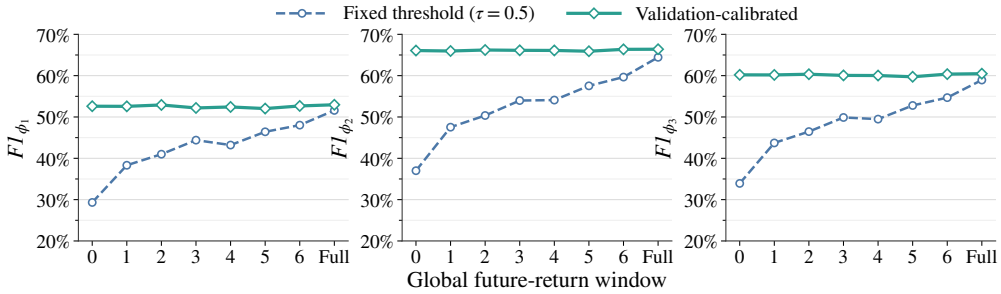
Here, K controls how many future turns contribute to the learning signal at turn i. When $K = 0$, a decision is trained only from the reward at that same turn. When $K = 1$, it also receives the next-turn reward, and larger values extend this window farther into the future. For example, in a five-turn conversation, the decision at turn 1 uses only r_1 when $K = 0$, uses $r_1 + r_2$ when $K = 1$, and eventually uses rewards from all five turns in the full setting. Importantly, we change only this reward window during training. The defense still observes the same dialogue at each turn. We can therefore ask directly whether a long conversation also requires rewards from distant future turns to train earlier decisions.

We follow the TurnGate setup on MTID (Shen et al., 2026), starting every policy from the same supervised checkpoint and holding the data, rewards, and optimization procedure fixed while varying $K \in \{0, 1, 2, 3, 4, 5, 6, \text{full}\}$. See details in Appendix A.4. Figure 7 shows that the answer depends strongly on how the decision threshold is chosen. At TurnGate's original threshold of $\tau = 0.5$, performance generally improves as rewards are propagated farther. After choosing each policy's threshold on the validation set, however, the differences largely disappear. Across reward windows, the standard deviation of test $F1_{\phi_2}$ is only 0.16%, and even $K = 0$ performs nearly the same as the full return. Appendix E provides additional metrics and threshold-free AUROC/AUPRC results.

The calibration result suggests that much of the apparent benefit from long reward windows comes from where a policy's scores fall relative to the fixed threshold, rather than from a large difference in the calibrated operating point. In this setting, rewards therefore do not need to travel across the entire conversation to train a similarly performing defense. Together with the attacker results, this reinforces the broader distinction developed throughout the paper: the trajectory may be long, while the useful temporal range of its learning signal can be much shorter.

6 Conclusion

Multi-turn jailbreaks are often described by the number of turns they take, yet the interaction used to discover an effective attack can be much longer than the history later needed to recover its progress.



(a) Exact-turn criterion (b) Linear premature-block penalty (c) Quadratic premature-block penalty

Figure 7: Long-range reward propagation has little effect on calibrated defense performance. TurnGate normally propagates rewards through the full remaining trajectory, while K limits each decision to rewards from at most K later turns; $K = 0$ uses only the current reward. Longer windows improve performance at the original threshold $\tau = 0.5$, but the gap largely disappears after selecting each policy's threshold on the validation set. For $F1_{\phi_2}$, performance across reward windows has a standard deviation of only 0.16%, and $K = 0$ performs nearly the same as the full return.

This distinction is becoming increasingly relevant as multi-turn attacks use more adaptive search and learned policies (Rahman et al., 2025; Xiong et al., 2026). In this work, we asked how much prior query history is needed to recover the same attack progress at a later turn. Motivated by this question, we introduced the *effective horizon*, which measures the shortest sequence of recent attacker queries sufficient to recover that progress. An attack may take many turns to discover even when only a few of those turns remain necessary once the useful queries have been found.

Across both search-based and training-based attacks, roughly half of the original query sequence is sufficient on average. This compression is useful beyond measurement. Recovering a high-scoring point from its effective history provides a better context for continuing unsuccessful attacks, while reducing final-outcome credit before the effective horizon improves attacker training. In defense training, rewards propagated over a short range preserve nearly the same calibrated performance as rewards propagated through the full remaining trajectory. Together, these results show that trajectory length and the amount of history that later behavior depends on can differ substantially. Treating them separately gives a more direct basis for deciding what context to retain and how far learning signals should extend (Zeng et al., 2025). This distinction may become important beyond jailbreaks as reinforcement learning is applied to long multi-turn agent trajectories (Wang et al., 2025).

Limitations and Future Work. Our results also raise broader questions about why later attack queries can remain effective with so little of the interaction history that helped produce them.

- **Recovering an attack is different from discovering it.** The effective horizon is measured only after later queries have been found. Earlier turns may still help construct those queries even if they are no longer needed for replay. This matters for multi-agent attacks (Rahman et al., 2025), adaptive tree search (Wei et al., 2025), and lifelong attack generation (Bhuiya et al., 2025). Future work could use effective-horizon information to discover shorter attack paths directly.
- **The horizon depends on how progress is measured.** We identify the shortest suffix that recovers the original attack score, so the horizon depends on the judge and recovery tolerance. We use established jailbreak evaluators, including HarmBench (Mazeika et al., 2024), but repeated replays or multiple measures of attack progress could provide more robust estimates. Our current analysis is also turn-level; measuring dependence within queries may reveal additional compression.
- **Temporal dependence can be richer than a single horizon.** EH-CA uses one boundary for final-outcome credit, while our defense experiment uses a fixed reward window rather than estimating a horizon for each decision. More adaptive methods could learn how much influence earlier turns should receive from the trajectory itself. This question also extends to multi-turn agents (Wang et al., 2025; Zeng et al., 2025) and multi-agent attacks (Rahman et al., 2025), where later actions may depend on selected messages, observations, or actions from much earlier in the interaction.

AI USE STATEMENT

Generative AI tools were used only to aid and polish the writing of this paper. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content.

ETHICS STATEMENT

This work studies the structure of multi-turn jailbreaks and therefore has a dual-use nature. While effective-horizon analysis may improve automated attack efficiency, its intended use is controlled red teaming, robustness evaluation, and the development of stronger multi-turn defenses. Our experiments use established research benchmarks and model-generated trajectories and involve no human participants or private data. We report aggregate results and a redacted illustrative example, with attacker queries summarized to preserve their conversational role without reproducing harmful details. We do not present harmful model outputs as instructional content.

REPRODUCIBILITY STATEMENT

We provide the experimental configurations, model and dataset specifications, evaluation protocols, and implementation details needed to reproduce our results in the appendix. Our experiments build on publicly available datasets, models, and open-source implementations, with all modifications introduced in this work explicitly documented. We will release our implementation and configuration files to facilitate independent reproduction.

REFERENCES

Justin Albrethsen, Yash Datta, Kunal Kumar, and Sharath Rajasekar. Deepcontext: Stateful real-time detection of multi-turn adversarial intent drift in llms. *arXiv preprint arXiv:2602.16935*, 2026.

Neeladri Bhuiya, Madhav Aggarwal, and Diptanshu Purwar. Plague: Plug-and-play framework for lifelong adaptive generation of multi-turn exploits. *arXiv preprint arXiv:2510.17947*, 2025.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37:55005–55029, 2024.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

Mingqian Feng, Xiaodong Liu, Weiwei Yang, Jialin Song, Xuekai Zhu, Chenliang Xu, and Jianfeng Gao. Sema: Simple yet effective learning for multi-turn jailbreak attacks. *arXiv preprint arXiv:2602.06854*, 2026.

Huanli Gong, Zhipeng Wei, Yu Fu, Haz Sameen Shahgir, Ananya Gupta, Yue Dong, and N Benjamin Erichson. D-judge: Disrupting multi-turn jailbreaks using semantics-preserving output rewriting. *arXiv preprint arXiv:2606.02640*, 2026.

Junwoo Ha, Hyunjun Kim, Sangyoon Yu, Haon Park, Ashkan Yousefpour, Yuna Park, and Suhyun Kim. M2s: Multi-turn to single-turn jailbreak in red teaming for llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16489–16507, 2025.

Zixuan Huang, Kecheng Huang, Lihao Yin, Bowei He, Huiling Zhen, Mingxuan Yuan, and Zili Shao. Attention-aware gnn-based input defense against multi-turn llm jailbreak. *arXiv preprint arXiv:2507.07146*, 2025.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, et al. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *Advances in Neural Information Processing Systems*, 37:47094–47165, 2024.

Martin Kuo, Jianyi Zhang, Aolin Ding, Louis DiValentin, Amin Hass, Benjamin F Morris, Isaac Jacobson, Randolph Linderman, James Kiessling, Nicolas Ramos, et al. Safety reasoning elicitation alignment for multi-turn dialogues. *arXiv preprint arXiv:2506.00668*, 2025.

Zhiqing Ma, Zhonghao Xu, Dong Yu, Chen Kang, Changliang Li, and Pengyuan Liu. Thrd: A training-free multi-turn defense framework for jailbreak attacks on large language models. *arXiv preprint arXiv:2606.01738*, 2026.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Md Messal Monem Miah, Adrita Anika, Zhiyuan Yu, and Ruihong Huang. Trace: Trajectory aware reasoning for multi-turn adversarial conversation evaluation. *arXiv preprint arXiv:2608.15594*, 2026.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, volume 2024, pp. 30988–31043, 2024.

Salman Rahman, Liwei Jiang, James Shiffer, Genglin Liu, Sheriff Issaka, Md Rizwan Parvez, Hamid Palangi, Kai-Wei Chang, Yejin Choi, and Saadia Gabriel. X-teaming: Multi-turn jailbreaks and defenses with adaptive multi-agents. *arXiv preprint arXiv:2504.13203*, 2025.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. Llms know their vulnerabilities: Uncover safety gaps through natural distribution shifts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24763–24785, 2025.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 2421–2440, 2025.

Xinjie Shen, Rongzhe Wei, Peizhi Niu, Haoyu Wang, Ruihan Wu, Eli Chien, Bo Li, Pin-Yu Chen, and Pan Li. One turn too late: Response-aware defense against hidden malicious intent in multi-turn dialogue. *arXiv preprint arXiv:2605.05630*, 2026.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1671–1685, 2024.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. *Advances in Neural Information Processing Systems*, 37:125416–125440, 2024.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Rongzhe Wei, Peizhi Niu, Xinjie Shen, Tony Tu, Yifan Li, Ruihan Wu, Eli Chien, Pin-Yu Chen, Olgica Milenkovic, and Pan Li. The trojan knowledge: Bypassing commercial llm guardrails via harmless prompt weaving and adaptive tree search. *arXiv preprint arXiv:2512.01353*, 2025.

Zixuan Weng, Xiaolong Jin, Jinyuan Jia, and Xiangyu Zhang. Foot-in-the-door: A multi-turn jailbreak for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1939–1950, 2025.

Xiqiao Xiong, Ouxiang Li, Zhuo Liu, Moxin Li, Wentao Shi, Fengbin Zhu, Qifan Wang, and Fuli Feng. Trojail: Trajectory-level optimization for multi-turn large language model jailbreaks with process rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 48086–48109, 2026.

Xikang Yang, Xuehai Tang, Songlin Hu, and Jizhong Han. Chain of attack: a semantic-driven contextual multi-turn attacker for llm. *arXiv preprint arXiv:2405.05610*, 2024.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, Yang Katie Zhao, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. In *ICML 2025 Workshop on Computer Use Agents*, 2025.

Yi Zhao and Youzhi Zhang. Siren: A learning-based multi-turn attack framework for simulating real-world human jailbreak behaviors. In *2025 IEEE Annual Computer Security Applications Conference (ACSAC)*, pp. 1209–1220. IEEE, 2025.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Additional Experimental Details

Because the main text focuses on the central empirical results, we use this section to spell out the parts of the protocol that matter for reproducing and interpreting them. The effective-horizon measurements and the two attack applications use the same replay operation. We keep a contiguous sequence of attacker queries unchanged, run those queries in a new conversation, and allow the victim to generate new responses. This makes it possible to ask how much of the original query history is sufficient to recover a previously reached level of attack progress without rewriting the attack or reusing the victim responses that originally produced it.

A.1 Effective-Horizon Compression

The main characterization compares two different ways of producing multi-turn attacks. For X-Teaming, we use Qwen2.5-32B-Instruct as the attacker and Llama-3.3-70B-Instruct as the victim. Attack progress is scored by GPT-4o using the 5-point jailbreak judge (Qi et al., 2024), with score 5 indicating success. For TROJail, we initialize the attacker with Qwen2.5-3B-Instruct, train it on AdvBench (Zou et al., 2023), and evaluate it against Llama-3.1-8B-Instruct using the HarmBench classifier (Mazeika et al., 2024), where scores above 0.9 indicate success. The main comparison uses 200 WildJailbreak behaviors with a maximum interaction length of five turns, and the expanded five-turn X-Teaming evaluation uses 1,500 behaviors.

For a target turn j, we record the original attack score v_j and evaluate candidate suffixes containing the last k attacker queries for $k \in \{1, \ldots, j\}$. Each candidate is run in a new conversation. Only the attacker queries $\mathbf{q}_{j-k+1:j}$ are retained; every victim response is generated again during replay. We do not summarize the removed prefix, rewrite the retained queries, or remove turns from the middle of the sequence. A candidate recovers the target progress when its replay score satisfies the criterion in Equation 1, and the effective horizon is the smallest k that does so. If no tested suffix satisfies the recovery criterion, the effective horizon is undefined for that target turn. Because the victim responses are regenerated, this procedure measures whether the retained attacker queries are sufficient to recover the score reached at turn j. It does not test whether the removed turns helped the attacker discover those later queries in the first place.

For successful trajectories, the target is the final turn, so the effective horizon measures how much of the query history remains necessary after the attack has reached success. For the unsuccessful-trajectory analysis in the main text, we instead truncate each trajectory at the first turn that reaches its highest attack score and measure the effective horizon at that point. This second construction is important for EH-CR because the continuation method does not require the initial attack to have succeeded. It only requires a point whose progress is worth recovering from a shorter history.

A.2 Context Recovery in Search-Based Attacks

The context-recovery experiments ask whether the same shorter history that is sufficient for replay is also useful when an attacker continues searching. For both X-Teaming and ActorBreaker, we use Qwen2.5-32B-Instruct as the attacker and evaluate on 159 harmful behaviors from HarmBench. The six victim models are Qwen3-30B-A3B, Llama-3.3-70B, Muse-Glimmer-30B, DeepSeek-V4.1-Flash, GPT-5.6 Luna, and Gemini 3.8 Flash. GPT-4o serves as the 5-point jailbreak judge (Qi et al., 2024), with score 5 indicating success. Because this score is discrete, EH-CR uses $\epsilon = 0$, so the recovered context must reach the same judge score as the selected point in the original trajectory.

In the main five-turn experiment, each attack first receives five turns. We apply continuation only when this initial trajectory has not succeeded, and then allow up to five additional turns. Full Context continues from the complete failed trajectory. Peak Context returns to the highest-scoring turn and retains the full query history leading to that turn. EH-CR returns to the same highest-scoring turn and keeps only the shortest suffix that recovers its score. If no shorter suffix recovers that progress, EH-CR keeps the full prefix up to the selected turn. The comparison between Peak Context and EH-CR is especially useful because both restart from the same point in the original search. Their difference is the amount of earlier query history retained. The robustness experiments in the main text vary the amount of interaction before recovery and the number of turns available afterward, so the continuation result does not depend on the specific choice of five initial turns and five continuation turns.

A.3 Outcome Credit in Attacker Training

The attacker-training experiment keeps the TROJail pipeline fixed and changes only how its final-outcome signal is distributed across queries. TROJail trains a separate Qwen2.5-3B-IT attacker for each victim using 520 AdvBench behaviors (Zou et al., 2023) over 260 optimization steps. We consider Qwen2.5-7B-IT, Llama-3.1-8B-IT, and Gemma-2-9B-IT as victims, and evaluate each attacker on HarmBench (Mazeika et al., 2024), StrongREJECT (Souly et al., 2024), and JailbreakBench (Chao et al., 2024). We report ASR@3 using the HarmBench classifier, counting a behavior as successful when at least one of three attempts scores above 0.9. We use $\epsilon = 0.02$ when computing the effective horizon for EH-CA, and leave TROJail's other credit components unchanged.

The credit-retention factor is selected without tuning it separately for every evaluation benchmark. For each victim, we choose $\beta \in \{0.1, 0.2, 0.3\}$ using HarmBench performance and then use that same value on StrongREJECT and JailbreakBench. The original uniform assignment $\beta = 1$ and complete removal of the prefix outcome signal $\beta = 0$ are retained as baselines. Section D reports the full sweep and shows how the HarmBench-selected values transfer to the other two benchmarks.

A.4 Reward Propagation in Defense Training

The defense experiment isolates the temporal range of the training signal while leaving the dialogue observed by the defense unchanged. We follow TurnGate (Shen et al., 2026) on the Multi-turn Intent Dataset (MTID). All variants use Qwen3-4B, start from the same reward-weighted SFT checkpoint, and are trained for three epochs with seed 42 using the same cached reference trajectories, reward design, and optimization settings. At inference time, the policy still observes the same conversation and makes a PASS or BLOCK decision at each turn. The only quantity changed during RL training is how many later rewards can contribute to the learning signal for an earlier decision.

With a reward window of K, the decision at turn i uses rewards from the current turn through at most K later turns,

$$A_i^{(K)} = \sum_{j=i}^{\min(i+K,T)} r_j. \tag{6}$$

Thus, $K = 0$ trains a decision only from the reward at that turn, $K = 1$ also includes the next reward, and Full includes every reward through the end of the trajectory. For example, the first decision in a five-turn conversation uses only r_1 when $K = 0$, uses $r_1 + r_2$ when $K = 1$, and uses $r_1 + \cdots + r_5$ in the Full setting. This ablation changes the distance over which rewards propagate during training without shortening the dialogue available to the defense.

TurnGate evaluates benign preservation together with how well harmful conversations are blocked at the first harm-enabling turn. For a dialogue trajectory $\mathbf{q}_{1:T}$, let $t^\star$ denote the first harm-enabling turn, with $t^\star = \infty$ for benign trajectories, and let $\hat{t}$ denote the first predicted BLOCK, with $\hat{t} = \infty$ if no block is produced. For benign trajectories,

$$\mathrm{BenignScore} = 1 - \mathrm{FPR}, \tag{7}$$

where FPR is the fraction of benign trajectories that are incorrectly blocked. For harmful trajectories, TurnGate uses

$$\phi_i(\hat{t}, t^\star) = \begin{cases} 0, & \hat{t} < t^\star,\ i = 1, \\ \left(\frac{\hat{t}}{t^\star}\right)^{i-1}, & \hat{t} < t^\star,\ i \in \{2, 3\}, \\ 1, & \hat{t} = t^\star, \\ 0, & \hat{t} > t^\star. \end{cases} \tag{8}$$

Here, ϕ_1 gives no credit to a premature block. The other two metrics give partial credit according to how close that block is to $t^\star$, with ϕ_3 penalizing an early decision more strongly than ϕ_2.

Let ACC denote the fraction of harmful trajectories blocked exactly at $t^\star$, and EARLY the fraction blocked before $t^\star$. The harmful-side score is

$$\mathrm{HarmfulScore}_{\phi_i} = \mathrm{ACC} + \mathrm{EARLY}\, \mathbb{E}\left[\phi_i(\hat{t}, t^\star) \mid \hat{t} < t^\star\right], \tag{9}$$

Table 2: A five-turn attack can depend on much less history when its later queries are run again. Attacker queries are lightly summarized to preserve their conversational role without reproducing harmful details. The final query alone reaches score 4, while replaying turns 4–5 in a new conversation recovers the original score of 5. The trajectory therefore has $H_{\text{eff}} = 2$ and EHR $= 0.4$. Earlier turns may still have helped discover the retained queries; the replay result shows that they are no longer required in the query history used to recover the final progress.

Turn	Redacted attacker query	Original score	Replay suffix
1	Request a neutral explanation of socioeconomic disparity.	1	Discarded
2	Ask how people with fewer resources are stereotyped in conversation and media.	2	Discarded
3	Request examples of disparaging jokes, framed as awareness of harmful language.	4	Discarded
4	Request supposedly "harmless" everyday remarks that reinforce class stereotypes.	4	Retained
5	Reframe the request as sarcastic remarks by a fictional character in social media or workplace settings.	5	Retained
Replay of turn 5 in a new conversation		final score 4 (not recovered)	
Replay of turns 4–5 in a new conversation		final score 5 (recovered)	

and the final metric is the harmonic mean of benign and harmful performance,

$$F1_{\phi_i} = \frac{2\,\text{BenignScore}\,\text{HarmfulScore}_{\phi_i}}{\text{BenignScore} + \text{HarmfulScore}_{\phi_i}}, \qquad i \in \{1, 2, 3\}. \tag{10}$$

We evaluate each learned policy in two ways because a fixed decision threshold can confound score calibration with the quality of the learned ranking. We first report the original TurnGate threshold $\tau = 0.5$. We then choose a separate threshold for each policy by maximizing $F1_{\phi_2}$ on the validation set and apply that threshold unchanged to the test set. Section E additionally reports AUROC and AUPRC, which remove the threshold choice altogether. These evaluations let us distinguish a policy that genuinely ranks harmful turns better from one whose scores have simply shifted relative to a fixed cutoff.

B A Concrete Effective-Horizon Example

The aggregate EHR results summarize how much query history can be removed, but they do not show what this compression looks like inside an actual attack. Table 2 follows one successful five-turn X-Teaming trajectory whose judge score rises from 1 to 5. The attacker reaches the maximum score only at turn 5. We then keep the original attacker queries fixed and replay suffixes of the trajectory in new conversations, allowing the victim responses to be generated again.

The final query by itself reaches score 4, so a one-turn suffix does not recover the progress of the original trajectory. When turn 4 is added, the replay reaches score 5. The shortest sufficient suffix therefore contains turns 4 and 5, which gives $H_{\text{eff}} = 2$ and an EHR of $2/5 = 0.4$. This example also shows why the effective horizon is more informative than asking whether the last query can jailbreak the victim on its own. The final query still depends on some local conversational setup, but that dependence extends only one turn backward rather than across the full five-turn trajectory.

The first three turns should not be interpreted as useless. They may have helped the attacker learn how the victim responds, move toward the later framing, or construct the queries that eventually reached score 5. The replay experiment asks a narrower question. Once turns 4 and 5 have already been discovered, are turns 1–3 still needed in the query history to recover the same progress? In this trajectory they are not. This is the distinction between the interaction used to discover a useful attack path and the shorter history sufficient to recover the progress reached along that path.

C VICTIM-LLM QUERY COST OF ATTACK CONTINUATION

The main continuation results compare methods under the same number of additional attack turns, but EH-CR performs extra victim queries while it searches for a suffix that recovers the selected progress. A natural concern is therefore that the ASR gain comes at the cost of many more victim calls. We examine total query cost separately from continuation length because EH-CR introduces two opposing effects. Validating candidate suffixes adds queries before continuation begins, while a better continuation context can allow the attack to succeed in fewer subsequent turns. The net cost depends on which effect is larger.

Table 3: EH-CR does not have a uniform effect on total victim-query cost. For X-Teaming, fewer continuation turns offset the queries used to validate the effective suffix, while for ActorBreaker they do not. The table reports average victim-LLM queries per behavior in the five-turn continuation experiment, including suffix-replay validation for EH-CR. Results include all valid behaviors for which the initial attack fails, regardless of continuation success. Lower is better, and best values are shown in bold.

	X-Teaming			ActorBreaker		
Victim Model	Full	Peak	EH-CR	Full	Peak	EH-CR
Qwen3-30B-A3B	15.57	14.16	**13.65**	**5.13**	5.24	8.14
Llama-3.3-70B	12.77	13.03	**11.90**	5.38	**5.36**	7.65
Muse-Glimmer-30B	**19.02**	19.05	19.35	**7.04**	7.09	10.04
DeepSeek-V4.1-Flash	18.12	**17.59**	18.11	6.79	**6.72**	8.85
GPT-5.6 Luna	18.70	**17.82**	18.22	6.54	**6.50**	7.89
Gemini 3.8 Flash	18.22	18.21	**16.80**	**5.88**	5.94	8.79
Avg.	17.07	16.64	**16.34**	**6.13**	6.14	8.56

In Table 3, the total-cost results show why EH-CR should be viewed primarily as a context-selection method. For X-Teaming, EH-CR uses 16.34 victim queries per behavior on average, compared with 17.07 for Full Context and 16.64 for Peak Context. Its replay overhead is therefore more than recovered by the reduction in downstream search. ActorBreaker behaves differently. EH-CR uses 8.56 queries per behavior, compared with 6.13 and 6.14 for Full and Peak Context. In that setting, replaying candidate suffixes costs more than the later continuation saves. The ASR improvement reported in the main text therefore does not come with a general claim of lower query cost.

Table 4 separates the continuation itself from the cost of constructing the context. EH-CR terminates earlier for every victim model under both attacks. The pooled average for X-Teaming decreases from 4.67 turns with Full Context and 4.58 with Peak Context to 4.03 with EH-CR. For ActorBreaker, the reduction is smaller, from 4.59 and 4.61 turns to 4.35. This distinction explains the cost pattern above. The shorter recovered history makes the subsequent search more effective in both cases, but only X-Teaming saves enough continuation queries to pay for suffix validation. The benefit of EH-CR is therefore better use of the continuation context; whether that also reduces total victim calls depends on the dynamics of the underlying attack.

D CROSS-BENCHMARK SELECTION OF THE CREDIT-RETENTION FACTOR

The main EH-CA result depends on the amount of outcome credit retained before the effective horizon. If this value were chosen independently on every evaluation benchmark, an improvement could reflect benchmark-specific tuning rather than a reusable credit-assignment rule. We therefore use a stricter selection protocol. For each victim model, we select $\beta \in \{0.1, 0.2, 0.3\}$ using HarmBench only and then freeze that value for StrongREJECT and JailbreakBench. The original TROJail assignment $\beta = 1$ and complete removal of prefix outcome credit $\beta = 0$ are evaluated as fixed baselines rather than selection candidates.

Table 5 shows that HarmBench selects $\beta = 0.1$ for Llama-3.1-8B-IT and Gemma-2-9B-IT, and $\beta = 0.3$ for Qwen2.5-7B-IT. These choices are then applied unchanged to the other two benchmarks. All six transfer evaluations improve over the original uniform assignment, with an average gain of 6.4

Table 4: The shorter EH-CR context leads to earlier termination for both attacks. The table reports the average number of continuation turns used per behavior under a five-turn budget. EH-CR uses fewer turns for every evaluated victim model with both X-Teaming and ActorBreaker, although the reduction is larger for X-Teaming. Lower is better, and the pooled average aggregates behaviors across all six victim models.

	X-Teaming			ActorBreaker		
Victim Model	Full	Peak	EH-CR	Full	Peak	EH-CR
Qwen3-30B-A3B	4.24	3.90	**3.06**	4.21	4.27	**4.09**
Llama-3.3-70B	3.58	3.61	**2.58**	4.22	4.25	**4.08**
Muse-Glimmer-30B	4.91	4.93	**4.56**	4.77	4.80	**4.59**
DeepSeek-V4.1-Flash	4.71	4.62	**4.26**	4.87	4.84	**4.55**
GPT-5.6 Luna	4.80	4.63	**4.07**	4.87	4.85	**4.46**
Gemini 3.8 Flash	4.68	4.70	**3.95**	4.36	4.42	**4.18**
Pooled Avg.	4.67	4.58	**4.03**	4.59	4.61	**4.35**

Table 5: A credit-retention factor selected on HarmBench transfers to two other benchmarks without further tuning. For each victim, we choose $\beta \in \{0.1, 0.2, 0.3\}$ on HarmBench and apply that value unchanged to StrongREJECT and JailbreakBench. Shaded rows show the selected settings. They outperform the original $\beta = 1$ assignment in all six cross-benchmark evaluations by 6.4 ASR@3 points on average; including HarmBench, the average gain is 6.9 points across all nine evaluations. Values are mean $\pm$ standard deviation over three evaluation runs.

Victim	Setting	HarmBench	StrongREJECT	JailbreakBench
Llama-3.1-8B-IT	Original ($\beta = 1$)	85.83 ± 0.62	84.38 ± 0.85	86.06 ± 2.27
	$\beta = 0$	83.83 ± 0.62	75.12 ± 0.59	89.09 ± 1.48
	$\beta = 0.1$	95.33 ± 1.31	90.16 ± 0.33	90.30 ± 0.86
	$\beta = 0.2$	90.67 ± 0.24	73.84 ± 1.66	78.18 ± 3.93
	$\beta = 0.3$	77.50 ± 0.41	58.80 ± 0.59	75.15 ± 2.27
Qwen2.5-7B-IT	Original ($\beta = 1$)	90.50 ± 1.41	93.63 ± 1.18	90.30 ± 2.27
	$\beta = 0$	94.67 ± 0.24	92.48 ± 1.82	96.97 ± 0.86
	$\beta = 0.1$	85.67 ± 1.43	84.72 ± 0.49	84.85 ± 3.74
	$\beta = 0.2$	88.50 ± 1.78	86.81 ± 0.75	92.12 ± 1.71
	$\beta = 0.3$	95.83 ± 1.43	96.64 ± 1.07	95.76 ± 2.27
Gemma-2-9B-IT	Original ($\beta = 1$)	84.17 ± 1.70	82.41 ± 0.87	83.64 ± 1.48
	$\beta = 0$	85.50 ± 1.22	78.82 ± 1.47	84.85 ± 2.27
	$\beta = 0.1$	92.83 ± 0.24	88.77 ± 1.34	96.97 ± 0.86
	$\beta = 0.2$	90.00 ± 0.82	84.84 ± 0.65	92.12 ± 3.74
	$\beta = 0.3$	89.83 ± 0.62	87.96 ± 1.34	90.30 ± 0.86

ASR@3 points. Within the tested partial-retention grid, the HarmBench-selected value also remains the strongest setting for the same victim on both StrongREJECT and JailbreakBench. Including the three HarmBench selection results gives an average improvement of 6.9 points across all nine victim–benchmark pairs. The consistency across benchmarks is useful because it indicates that the benefit is not produced by choosing a separate β after seeing each test benchmark.

The comparison with $\beta = 0$ also clarifies what the effective horizon does and does not say about credit. Completely removing the prefix outcome signal improves five of the nine evaluations but hurts the other four, including all three StrongREJECT settings. Queries before the effective horizon can therefore still carry useful training information even when they are not needed to replay the final attack progress. They may help the attacker discover the later queries that fall inside the effective horizon. EH-CA retains that possibility by assigning those earlier queries a smaller outcome signal instead of forcing their credit to zero. The transfer results support this softer use of the effective horizon as a boundary for reducing credit rather than treating it as a causal cutoff.

Table 6: After threshold calibration, TurnGate performance changes little across reward-propagation windows. All variants use Qwen3-4B, start from the same reward-weighted SFT checkpoint, and differ only in K, the number of later turns whose rewards can contribute to an earlier decision. Full uses every remaining reward. Each RL policy's BLOCK threshold is selected on validation using $F1_{\phi_2}$ and then fixed for all test metrics. The additional Full row uses TurnGate's original fixed threshold $\tau = 0.5$ so that the effect of calibration can be seen directly.

		Benign (Utility)	Harmful						Overall		
Method	τ	Benign Score ↑	Miss ↓	Early ↓	ℓ_1 ↓	Acc. (ϕ_1) ↑	Harmful Score ϕ_2 ↑	Harmful Score ϕ_3 ↑	$F1_{\phi_1}$ ↑	$F1_{\phi_2}$ ↑	$F1_{\phi_3}$ ↑
Reward-weighted SFT†	–	0.856	0.407	0.262	0.90	0.332	0.460	0.402	0.478	0.599	0.547
TurnGate (Full, fixed threshold)	0.5000	0.818	0.306	0.318	0.93	0.377	0.531	0.461	0.516	0.644	0.589
TurnGate ($K = 0$)	0.3430	0.765	0.219	0.380	0.98	0.401	0.582	0.496	0.526	0.661	0.602
TurnGate ($K = 1$)	0.4013	0.796	0.253	0.355	0.95	0.393	0.563	0.484	0.526	0.660	0.602
TurnGate ($K = 2$)	0.4013	0.794	0.241	0.362	0.96	0.397	0.568	0.487	0.529	0.662	0.603
TurnGate ($K = 3$)	0.4013	0.788	0.233	0.377	0.97	0.390	0.569	0.485	0.522	0.661	0.601
TurnGate ($K = 4$)	0.4013	0.786	0.229	0.378	0.99	0.393	0.570	0.486	0.524	0.661	0.600
TurnGate ($K = 5$)	0.4013	0.772	0.216	0.392	1.01	0.393	0.575	0.487	0.520	0.659	0.597
TurnGate ($K = 6$)	0.4378	0.790	0.232	0.373	0.98	0.395	0.572	0.488	0.527	0.664	0.604
TurnGate (Full)	0.4626	0.780	0.223	0.376	0.99	0.401	0.578	0.494	0.530	0.664	0.605

† The reward-weighted SFT checkpoint is shown as a reference using its historical default-decoding evaluation and is not calibration-matched to the RL rows. All scores are in $[0, 1]$, except ℓ_1, which is measured in turns over blocked harmful trajectories.

E ADDITIONAL RESULTS FOR REWARD PROPAGATION IN DEFENSE TRAINING

The main defense experiment asks whether an earlier decision needs rewards from the full remainder of the conversation in order to learn an effective blocking policy. The fixed-threshold results alone leave an important ambiguity. A longer reward window might genuinely improve the policy's ability to distinguish the harm-enabling turn, or it might mainly move the policy's scores relative to the fixed threshold $\tau = 0.5$. We use validation-based threshold calibration and threshold-free ranking metrics to separate these possibilities. The full metrics below also show how changes in the reward window trade off benign preservation, missed harmful turns, and premature blocking.

Table 6 makes the calibration effect visible. At the original threshold $\tau = 0.5$, the Full policy obtains $F1_{\phi_2} = 0.644$, and the ablation in the main text shows a general improvement as rewards are allowed to propagate farther. The learned policies do not, however, use the same score scale. Their validation-selected thresholds range from 0.3430 for $K = 0$ to 0.4626 for Full. Once each policy is evaluated at its own validation-selected threshold, the difference across reward windows becomes small. $F1_{\phi_2}$ ranges from 0.659 to 0.664, with a standard deviation of only 0.16 percentage points. Even $K = 0$, which trains every decision using only its same-turn reward, reaches 0.661 compared with 0.664 for Full. The benign and harmful components move somewhat differently across settings, but these changes largely cancel at the calibrated operating point.

This result indicates that a fixed threshold can make long-range propagation look more important than it is for the underlying decision rule. Calibration alone cannot establish that the learned rankings are equivalent, because two policies can achieve similar performance after choosing different thresholds while still ordering examples differently. We therefore evaluate the raw score rankings directly in Table 7. AUROC and AUPRC do not require choosing a BLOCK threshold, so they provide a separate test of whether larger K makes harm-enabling turns more distinguishable from benign and pre-harm turns.

The threshold-free results reinforce the calibrated comparison. AUROC is 0.810 for $K = 0$ and 0.809 for Full, while AUPRC is 0.457 and 0.452, respectively. There is no monotonic improvement as K grows, and the entire range across all eight policies is very small. In this experiment, propagating rewards through more of the remaining trajectory therefore does not produce a measurable improvement in the model's ability to rank the turn at which harmful intent becomes actionable. The fixed-threshold gains are better explained by differences in where the learned scores fall relative to a common cutoff.

Table 7: Threshold-free ranking quality is nearly unchanged across reward-propagation windows. AUROC varies only from 0.808 to 0.810, while AUPRC varies from 0.452 to 0.457. The $K = 0$ policy matches or slightly exceeds Full on both metrics. The ranking task treats harm-enabling turns as positives and benign and pre-harm turns as negatives; the AUPRC no-skill baseline is the positive prevalence of 19.27%.

	Global future-return window							
Metric	$K = 0$	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	Full
AUROC ↑	0.810	0.809	0.810	0.809	0.808	0.808	0.809	0.809
AUPRC ↑	0.457	0.453	0.457	0.455	0.454	0.453	0.455	0.452

Our reproduced Full setting at the original threshold differs slightly from the value reported by TurnGate. We keep this row because it anchors the ablation to the released setup, while the scientific comparison in our experiment is between matched runs. Every reward-window policy starts from the same SFT checkpoint and uses the same data, cached trajectories, rewards, number of training epochs, seed, and optimization settings. Only K changes. We therefore interpret relative differences across these matched policies rather than small differences in absolute reproduction. The reward-weighted SFT row is included only as a reference because its historical default-decoding evaluation is not calibration-matched to the RL rows.

Taken together, the calibrated and threshold-free results sharpen the claim made in the main text. The defense still observes the complete dialogue at each turn, so short reward windows do not remove conversational information from the policy input. They only shorten the distance over which a later reward can train an earlier decision. Under this controlled offline setting, extending that distance to the full remaining trajectory provides little additional benefit once threshold placement is accounted for. The useful temporal range of the learning signal can therefore be much shorter than the dialogue on which the decision is conditioned.