PARTIAL POLICY GRADIENTS FOR CONVERSATIONAL RL IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning to act sequentially in an unknown environment can be formulated using reinforcement learning and solved by policy gradients. We propose a natural approach for trading off the bias and variance in policy gradients. The key idea is to optimize for a subset of future rewards: smaller subsets represent simpler policies, which can be learned more efficiently because their empirical gradient estimates have lower variance. Our approach encompasses various policy classes, such as full planning, greedy, and K-step lookahead policies. We evaluate the policies empirically in the offline setting on conversational tasks of persona alignment and agentic tool-calling, and directly measure gradient variance across lookahead horizons. Different policies excel in different problems, reflecting their different characteristics and highlighting the importance of our studied trade-off.

1 INTRODUCTION

Reinforcement learning (RL) (Sutton & Barto, 1998) is a framework for learning to act sequentially in an unknown environment with the goal of maximizing future rewards. Due to its generality and broad applicability, RL has been studied extensively and many RL algorithms have been proposed, including temporal-difference learning (Sutton, 1988), Q-learning (Watkins & Dayan, 1992), policy gradients (Williams, 1992), and actor-critic methods (Sutton et al., 2000). RL is more data demanding than supervised learning because it plans for the future (Sutton & Barto, 1998). Therefore, many early works on planning and RL tried to increase the statistical efficiency of learning algorithms by assuming additional structure, such as a known model of the environment (Bellman, 1957; Boutilier et al., 1995; Koller & Parr, 1999), linear value or Q functions (Tsitsiklis & Van Roy, 1997; de Farias & Van Roy, 2003; Hauskrecht & Kveton, 2004), and baseline subtraction in policy gradients (Sutton et al., 2000; Baxter & Bartlett, 2001; Munos, 2006).

Proximal policy optimization (PPO) (Schulman et al., 2017) and *group-relative policy optimization (GRPO)* (Shao et al., 2024) are arguably the most popular RL algorithms for *large language models (LLMs)* (Ouyang et al., 2022). PPO is a policy gradient algorithm where the advantage is estimated using a critic's value estimate. The challenge with applying PPO in practice is that the critic has to be learned. GRPO is a policy gradient algorithm where the advantage is estimated by simulation, which eliminates the need for learning the critic. The challenge with applying GRPO is that the advantage is attributed equally to all tokens, potentially reducing statistical efficiency.

We propose a natural approach for trading off the bias and variance in policy gradients. The first step is decomposing the total reward additively over time. This decomposition may be a part of problem definition, for instance when the problem is a *Markov decision process (MDP)* with per-step rewards (Bellman, 1957; Puterman, 1994), or obtained by a simple algebraic transformation. The second step is optimization for subsets of future rewards. The subsets lead to simpler policies than full planning, which can be learned more efficiently because their empirical gradient estimates have lower variance. We study both online and offline settings, and analyze them. Full planning, greedy, and K-step lookahead policies are all instances of our framework.

We evaluate our policies on two problems. The first problem is simulation of human personas (Tseng et al., 2024; Jandaghi et al., 2024). Off-the-shelf LLMs frequently drift from their assigned personas, contradict to their earlier statements, or abandon role-appropriate behavior (Ji et al., 2025; Abdulhai et al., 2025). We apply our policies to LLMs to maintain consistent and faithful human personas in role-playing dialogues. This problem is challenging because many dialogues have long horizons, on

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.

average 20 steps and up to 60. Our learned K-step lookahead policies lead to consistent personas in extended dialogues, mitigating the persona drift that plagues base models. The second problem is agentic tool-calling in multi-turn conversations (Blakeman et al., 2025), where the agent must follow an operational policy guidelines without skipping required preconditions or violating procedural constraints. The two problems are analogous in their conversational nature: protocol violations in tool-calling mirror failure to maintain persona consistency in dialogue. Our **main contributions** are:

1. We propose a natural approach for trading off the bias and variance in policy gradients. The key idea is to optimize for a subset of future rewards, which can be either observed or obtained by a simple algebraic transformation. Our formulation is more general than those in the closest related works, which focused only on specific policies (Kveton et al., 2025) and sometimes not even with observations (Guo et al., 2025).
2. Our framework trades off the complexity of learned policies for the statistical efficiency of learning them: optimization for smaller subsets of future rewards leads to learning simpler policies, which can be learned more efficiently because their empirical gradient estimates have lower variance. We propose both online and offline variants of our approach.
3. We instantiate the framework: both the classic policy gradient (Williams, 1992) and that for greedy policies (Kveton et al., 2025) are instances of our framework. This is the first work that proposes and empirically evaluates K-step lookahead policies in LLMs.
4. We empirically evaluate our policies on conversational persona-alignment and tool-calling problems. We experimentally demonstrate that that K-step lookahead policies lead to $10 - 19\%$ improvements (Section 5.2) by preventing the drift across multiple LLMs. The optimal value of K depends on the complexity of the task and training data size.

The paper is organized as follows. We present our setting in Section 2. In Section 3, we present our framework for optimizing a subset of future rewards, interpret it, and analyze it. We instantiate the framework in Section 4.3 and evaluate it empirically in Section 5. Related works are reviewed in Section 6, and we conclude in Section 7.

2 Setting

We start with introducing our notation. The marginal and conditional probabilities under the probability measure p are $p(X = x)$ and $p(X = x \mid Y = y)$, respectively; and we write $p(x)$ and $p(x \mid y)$ when the random variables are clear from context. The indicator function is $\mathbb{1}\{\cdot\}$. For a positive integer n, we define $[n] = \{1, \ldots, n\}$. We denote the i-th entry of vector v by v_i. If the vector is already indexed, such as v_j, we write $v_{j,i}$.

We consider a general reinforcement learning problem (Sutton & Barto, 1998) where an *agent* interacts with an *environment* in an *episode* consisting of multiple steps. The agent takes actions conditioned on the interaction history and the environment responds. When the episode ends, it is assigned a reward. The reward measures the quality of the interaction and the agent maximizes it.

We formalize the problem as follows. The agent first observes *context* $x \in \mathcal{S}$, where $\mathcal{S}$ is the space of all token sequences. The context defines the task. The interaction between the agent and environment consists of steps indexed by $t \in \mathbb{N}$, where $\mathbb{N}$ is a set of positive integers. At step t, the agent takes an *action* $a_t \in \mathcal{S}$ and the environment responds with an *observation* $y_t \in \mathcal{S}$. The interaction history up to step n is a *trajectory* $\tau_n = (a_1, y_1, \ldots, a_n, y_n)$ of n actions and observations. We assume that the number of steps n is fixed. The *reward* is a non-negative function of x and τ_n, denoted by $r(x, \tau_n) \geq 0$, and measures the quality of the interaction. We do not make any additional assumptions on the reward. This is to maintain generality and because our algorithms (Section 3) do not require it.

The agent follows a policy conditioned on its history. More specifically, the probability that action a is taken in context x and history τ_{t-1} is $\pi(a \mid x, \tau_{t-1}; \theta)$, and is parameterized by $\theta \in \Theta$. The probability of observing y_t conditioned on history τ_{t-1} and action a_t is denoted by $p(y_t \mid x, \tau_{t-1}, a_t)$. We denote by $\pi(\tau_n \mid x; \theta) = \prod_{t=1}^{n} p(y_t \mid x, \tau_{t-1}, a_t)\, \pi(a_t \mid x, \tau_{t-1}; \theta)$ the probability of trajectory τ_n in context x. The factorization follows from the chain rule of probability. The expected value of the policy, where q is a distribution over contexts x, is defined as

$$V(\theta) = \mathbb{E}_{x \sim q,\, \tau_n \sim \pi(\cdot \mid x; \theta)} \left[r(x, \tau_n) \right] . \tag{1}$$

Algorithm 1 Policy gradient algorithm.

1: **Input:** Learning rate schedule $(\alpha_j)_{j\in\mathbb{N}}$, batch size B, gradient estimator g
2: Initialize θ and $j \leftarrow 1$
3: **while** (not convergence) **do**
4: **for** $i = 1, \ldots, B$ **do**
5: $x_i \sim q,\ \tau_{i,n} \sim \pi(\cdot \mid x_i; \theta),\ r_i \leftarrow r(x_i, \tau_{i,n})$ ▷ Online policy gradient
6: $x_i \sim q,\ \tau_{i,n} \sim \pi_0(\cdot \mid x_i; \theta),\ r_i \leftarrow r(x_i, \tau_{i,n})$ ▷ Offline policy gradient
7: Batch $\mathcal{B} \leftarrow \{(x_i, \tau_{i,n}, r_i)\}_{i=1}^{B}$
8: $\theta \leftarrow \theta + \alpha_j g(\mathcal{B}),\ j \leftarrow j + 1$
9: **Output:** Learned policy parameter θ

Our goal is to learn a policy $\theta_* = \arg\max_{\theta\in\Theta} V(\theta)$ with the highest expected value. Our formulation is more general than those in the closest related works, which focused only on specific policies and did not consider observations (Kveton et al., 2025; Guo et al., 2025).

Our framework can model many problems. For instance, suppose that we want to maximize the pedagogical value of a conversation with a student over n steps (Scarlatos et al., 2025). Then τ_n is an n-step conversation with the student and $r(x, \tau_n)$ is its pedagogical value. Similarly, suppose that we want to align responses of an agent to its persona in a dialogue (Abdulhai et al., 2025). Then τ_n is the same as before and $r(x, \tau_n)$ is the number of steps where the responses are aligned. Finally, suppose that we want a tool calling agent follow its operational policy, without skipping required preconditions or violating procedural constraints. Then τ_n is the trajectory of tools calls and $r(x, \tau_n)$ is the number of calls where the operational policy is not violated. We experiment with the last two problems in Section 5.

3 Algorithms

This section has three main parts. In Section 3.1, we introduce policy gradients. In Section 3.2, we present our credit attribution strategy. In Section 3.3, we introduce partial policy gradients.

3.1 Policy Gradient

We want to maximize the expected policy value $V(\theta)$ in (1). The most straightforward approach for complex policies, like those represented by LLMs, are policy gradients (Williams, 1992; Sutton & Barto, 1998). We consider both online and offline policy gradients.

The key idea in an *online policy gradient* is to update the policy iteratively by gradient ascent. The gradient of $V(\theta)$ at θ is

$$\nabla V(\theta) = \mathbb{E}\left[r(x, \tau_n) \nabla \log \pi(\tau_n \mid x; \theta)\right] = \mathbb{E}\left[r(x, \tau_n) \sum_{t=1}^{n} \nabla \log \pi(a_t \mid x, \tau_{t-1}; \theta)\right], \quad (2)$$

where $\mathbb{E}[\cdot] = \mathbb{E}_{x\sim q,\, \tau_n \sim \pi(\cdot|x;\theta)}[\cdot]$ is an expectation over the same random variables as in (1). This identity can be derived by a direct application of the score identity (Aleksandrov et al., 1968). The gradient is estimated empirically as

$$g(\mathcal{B}) = \frac{1}{B} \sum_{i=1}^{B} r_i \sum_{t=1}^{n} \nabla \log \pi(a_{i,t} \mid x_i, \tau_{i,t-1}; \theta)\,. \quad (3)$$

where $\mathcal{B}$ is the batch, B is its size, and $\tau_{i,n}$ is the i-th trajectory in $\mathcal{B}$; with context x_i and reward r_i. We present the complete policy gradient algorithm in Algorithm 1 and call it PG. The online policy gradient corresponds to Line 5.

Offline RL (Jaques et al., 2020; Levine et al., 2020) is a class of machine learning algorithms that can be applied to datasets of logged trajectories with rewards, similarly to supervised fine-tuning and direct preference optimization in LLMs (Rafailov et al., 2023). We consider the *offline policy gradient* in Mukherjee et al. (2025) because of its algebraic proximity to (2). The key idea in their work is to replace trajectory sampling distribution in (2) with that from a logged dataset, or equivalently another fixed policy π_0. Mathematically, the only difference from (5) is that the expectation is replaced with $\mathbb{E}_0[\cdot] = \mathbb{E}_{x\sim q,\, \tau_n \sim \pi_0(\cdot|x)}[\cdot]$. This approach does not require propensity scores and guarantees a local improvement in (1). The offline policy gradient corresponds to Line 6.

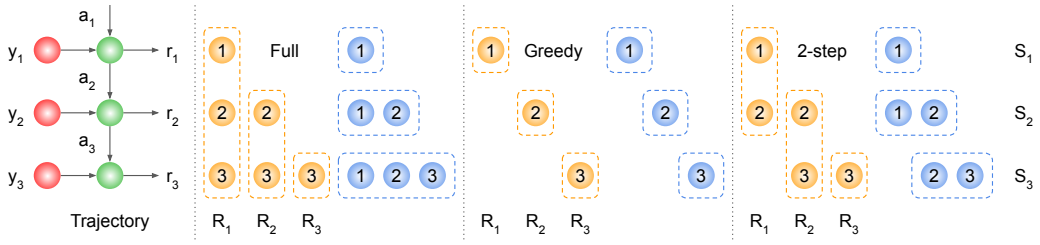


Figure 1: Examples of reward indices $\mathcal{R}_t$ affected by a_t (yellow circles in columns) and action indices $\mathcal{S}_t$ affecting r_t (blue circles in rows) in full, greedy, and 2-step lookahead policies.

3.2 Partial Credit Assignment

The reward in (2) is attributed equally to each step, since all $\nabla \log \pi(a_t \mid x, \tau_{t-1}; \theta)$ are multiplied by the same reward $r(x, \tau_n)$. If the attribution improved, we could reduce the variance of estimator (3) and improve statistical efficiency. We propose a general attribution framework that allows for such tradeoffs. Our key assumption is that the reward factors as

$$r(x, \tau_n) = \textstyle\sum_{t=1}^{n} r_t \,, \tag{4}$$

where $r_t = r_t(x, \tau_t)$ denotes the *reward at step* t, and is a function of x and τ_t.

Intermediate rewards r_t are common in practice. In conversational persona-alignment (Abdulhai et al., 2025), the reward indicates if the agent's response aligns with its persona. In tool calling, the reward indicates if the agent's response follows operational policy, without skipping required preconditions or violating procedural constraints. We experiment with both of these problems in Section 5. When intermediate rewards do not exist, they can be constructed as follows.

Lemma 1. *Let $f(x, \tau_t)$ be any function of x and τ_t such that $f(x, \tau_0) = 0$ and $f(x, \tau_n) = r(x, \tau_n)$. Let $r_t = f(x, \tau_t) - f(x, \tau_{t-1})$ hold for all $t \in [n]$. Then $\sum_{t=1}^{n} r_t = r(x, \tau_n)$.*

The claim is proved in Section A.1. Since $f(x, \tau_t)$ can be viewed as a trajectory reward up to step t, r_t is the per-step gain in the total trajectory reward at step t. One natural choice of f is an LLM judge (Zheng et al., 2023) that can judge partial trajectories and returns 0 when the length of the trajectory is zero. Similar decompositions were proposed before in more specific problems, without observations (Guo et al., 2025) and in optimization of submodular policies (Kveton et al., 2025).

3.3 Partial Policy Gradient

Our main algorithmic contribution is a general *partial policy gradient* of the form

$$\mathbb{E}\left[\sum_{t=1}^{n}\left(\sum_{\ell \in \mathcal{R}_t} r_\ell\right) \nabla \log \pi(a_t \mid x, \tau_{t-1}; \theta)\right] , \tag{5}$$

where $\mathcal{R}_t \subseteq [n] \setminus [t-1]$ are *reward indices affected by action at step* t through $\nabla \log \pi(a_t \mid x, \tau_{t-1}; \theta)$. Specifically, $\mathcal{R}_t$ are future steps whose rewards are affected by a_t conditioned on the history. The gradient can be estimated empirically as

$$g(\mathcal{B}) = \frac{1}{B}\sum_{i=1}^{B}\sum_{t=1}^{n}\left(\sum_{\ell \in \mathcal{R}_t} r_{i,\ell}\right) \nabla \log \pi(a_{i,t} \mid x_i, \tau_{i,t-1}; \theta) \,, \tag{6}$$

where $\tau_{i,n}$ is the i-th trajectory in batch $\mathcal{B}$, generated exactly as in (3); and $a_{i,\ell}$, $r_{i,\ell}$, and $\tau_{i,\ell-1}$ are the action, reward, and trajectory history at step t of this trajectory, respectively. When (6) is used in Algorithm 1, we call it partial policy gradient (PPG) since only a subset of future rewards is used in policy optimization. Note that PPG reduces to PG when $\mathcal{R}_t = [n]$ for all $t \in [n]$. Our framework allows us to reason about other $\mathcal{R}_t$ and understand their statistical tradeoffs.

The motivation for PPG is that (6) should concentrate faster than (3) when $\sum_{\ell \in \mathcal{R}_t} r_{i,\ell} \ll r_t$ because the log-probability gradients, when viewed as random variables, are multiplied by smaller scalars and thus have smaller magnitudes. In the next section, we interpret the partial policy gradient in (5) and analyze the concentration rate of its estimate (6).

4 ANALYSIS

This section has three parts. In Section 4.1, we interpret the gradient in (5). In Section 4.2, we analyze the rate at which (6) concentrates at (5). In Section 4.3, we discuss and analyze policies that can be represented in our framework.

4.1 INTERPRETATION

We interpret the gradient in (5) next. Let $\mathcal{S}_t = \{\ell \in [t] : t \in \mathcal{R}_\ell\} \subseteq [t]$ be *action indices that affect the reward at step* t, r_t. A natural way of viewing $\mathcal{S}_t$ is an inverse mapping to $\mathcal{R}_t$, which is the set of reward indices affected by action at step t. We visualize this relation in Figure 1. We also define $\bar{S}_t = [t] \setminus \mathcal{S}_t$, the complement of $\mathcal{S}_t$ with respect to the first t steps. Using this notation, we define the conditional expectation over trajectory τ up to step t, over entries $\mathcal{S}_t$ conditioned on $\bar{S}_t$,

$$\mathbb{E}_t\left[\cdot\right] = \mathbb{E}_{\tau_{\mathcal{S}_t} \sim \pi(\cdot|x;\theta)}\left[\cdot \,\middle|\, x, \tau_{\bar{S}_t}\right] . \tag{7}$$

This expectation is well defined because $\mathcal{S}_t$ and $\bar{S}_t$ are functions of only t, and not random. Moreover, note that the expectation is a function of entries $\bar{S}_t$ in τ because we condition on them. With the new notation in hand, we get the following claim.

Lemma 2. *The online partial policy gradient in* (5) *is equal to* $\mathbb{E}\left[\sum_{t=1}^n \nabla \mathbb{E}_t\left[r_t\right]\right]$.

The proof is in Section A.3. The lemma says that the gradient in (5) increases the sum of expected per-step rewards r_t through actions at steps $\mathcal{S}_t$ that affect them, conditioned on the history $\bar{S}_t$. The histories are sampled from the learned policy π.

Now we consider the offline RL setting in Section 3.1. As we already discussed, the only difference in the policy gradient is that $\tau_n \sim \pi(\cdot \mid x;\theta)$ is replaced by $\tau_n \sim \pi_0(\cdot \mid x)$. We define the offline conditional expectation over trajectory τ up to step t, over entries $\mathcal{S}_t$ conditioned on $\bar{S}_t$,

$$\mathbb{E}_{0,t}\left[\cdot\right] = \mathbb{E}_{\tau_{\mathcal{S}_t} \sim \pi_0(\cdot|x)}\left[\cdot \,\middle|\, x, \tau_{\bar{S}_t}\right] . \tag{8}$$

With these in hand, we get the following claim.

Lemma 3. *The offline partial policy gradient is*

$$\mathbb{E}_0\left[\sum_{t=1}^n \left(\sum_{\ell \in \mathcal{R}_t} r_\ell\right) \nabla \log \pi(a_t \mid x, \tau_{t-1};\theta)\right] = \mathbb{E}_0\left[\sum_{t=1}^n \nabla \mathbb{E}_{0,t}\left[r_t \sum_{\ell \in \mathcal{S}_t} \log \pi(a_\ell, y_\ell \mid x, \tau_{\ell-1};\theta)\right]\right] .$$

The proof is in Section A.4. The claim says that the gradient increases the sum of expected per-step rewards r_t weighted by log-probabilities under logging policy π_0 through actions at steps $\mathcal{S}_t$ that affect them, conditioned on the history $\bar{S}_t$. The histories are sampled from the logging policy π_0.

The gradient in Lemma 3 is important because it leads to optimizing a closely related objective to $\mathbb{E}\left[\sum_{t=1}^n \mathbb{E}_t\left[r_t\right]\right]$. To see this, note that when the rewards are non-negative, we have by Lemma 1 of Mukherjee et al. (2025) that $\mathbb{E}_{0,t}\left[r_t\right] \geq \mathbb{E}_{0,t}\left[r_t \sum_{\ell \in \mathcal{S}_t} \log \pi(a_\ell, y_\ell \mid x, \tau_{\ell-1};\theta)\right]$. This means that the offline partial policy gradient optimizes a lower bound on the true objective. Generally, by Lemma 3 of Mukherjee et al. (2025), $|\mathbb{E}_{0,t}\left[r_t\right] - \mathbb{E}_{0,t}\left[r_t \sum_{\ell \in \mathcal{S}_t} -\pi(a_\ell, y_\ell \mid x, \tau_{\ell-1};\theta)\right]| \leq C(\pi, \pi_0)$, where $C(\pi, \pi_0)$ is a constant depending on the propensity ratios of π and π_0, and their logarithms. In this case, the offline partial policy gradient optimizes an $C(\pi, \pi_0)$-close objective to the true one.

4.2 GRADIENT CONCENTRATION

The variance of policy gradients (Williams, 1992) grows quadratically with trajectory length, and this motivated countless prior works on variance reduction of policy gradients (Sutton et al., 2000; Baxter et al., 2001; Baxter & Bartlett, 2001; Munos, 2006). In Theorem 4, we analyze the concentration rate of our empirical partial policy gradient estimator in (6). The result implies that the empirical gradients of simpler policies, which only optimize a small subset of future rewards, concentrate faster and thus have a lower variance.

Theorem 4. *Let* $g(\mathcal{B})$ *be the estimator in* (6). *Let* $|r_{i,t}| \leq c_t$ *hold almost surely for all* $i \in [B]$ *and* $t \in [n]$. *Let* $|\nabla \log \pi(a_{i,t} \mid x_i, \tau_{i,t-1};\theta))_j| \leq 1$ *hold almost surely for all* $i \in [B]$, $t \in [n]$, *and* $j \in [d]$, *where* d *is the dimension of policy parameter* θ. *Then for any* $\varepsilon > 0$,

$$\mathbb{P}\left(\|g(\mathcal{B}) - \mathbb{E}\left[g(\mathcal{B})\right]\|_2 \geq \varepsilon\right) \leq 2d \exp\left[-\varepsilon^2 B / (2d\left(\sum_{t=1}^n \sum_{\ell \in \mathcal{S}_t} c_\ell\right)^2)\right] .$$

The claim is proved in Section A.2. The key idea in the proof is treating (6) as a sum of random vectors, independent across i but dependent across t. Then we use Hoeffding's inequality, which is a standard tool in concentration analyses (Raginsky & Sason, 2012; Boucheron et al., 2013), to argue that the sum of random vectors concentrates proportionally to the scalar rewards that multiply them. The bound shows that the concentration rate decreases as $\sum_{t=1}^{n} \sum_{\ell \in \mathcal{S}_t} c_t$ increases. Therefore, for any two partial policy gradient estimators $g(\mathcal{B})$ and $g'(\mathcal{B})$ such that $\mathcal{S}_t \subseteq \mathcal{S}'_t$ for all $t \in [n]$, the former concentrates faster than the latter in the worst case.

Note that we do not analyze the bias of $g(\mathcal{B})$. This is because in our operating regimes of interest, where n future rewards are replaced by $K \ll n$ in K-step lookahead policies, the bias is clearly $\Omega(n)$ and would not explain the effectiveness of the policies. Therefore, we view the policies as a restricted policy class that is easier to learn but may be suboptimal when long-range planning matters, and only analyze the hardness of learning within the class. This is not a classic bias-variance decomposition of a single estimator.

4.3 Classes of Representable Policies

Now we are ready to instantiate our framework to show its generality.

Full planning. Without making any assumption on how actions affect rewards, any action a_t in (5) can affect any future reward, $\mathcal{R}_t^{\text{full}} = [n] \setminus [t-1]$ for all steps $t \in [n]$. Therefore, any reward r_t may be affected by all prior actions, $\mathcal{S}_t^{\text{full}} = [t]$. By Lemma 2, the gradient is $\mathbb{E}\left[\sum_{t=1}^{n} \nabla \mathbb{E}_t\left[r_t\right]\right] = \nabla \mathbb{E}\left[r\right]$ and identical to (2). This is the baseline that we want to beat.

Greedy policies. These policies optimize the immediate reward. Mathematically, any action a_t in (5) only affects its corresponding reward, $\mathcal{R}_t^{\text{greedy}} = \{t\}$ for all $t \in [n]$. Therefore, the reward r_t is only affected by action a_t, $\mathcal{S}_t^{\text{greedy}} = \{t\}$. Since $\mathcal{S}_t^{\text{greedy}} \subseteq \mathcal{S}_t^{\text{full}}$, we have by Lemma 2 that the gradient of the greedy policies concentrates faster than that of full policies. We refer to PPG with $\mathcal{R}_t^{\text{greedy}}$ as GreedyPG. While the greedy policies in LLMs were proposed, analyzed, and evaluated by Kveton et al. (2025), it was only for adaptive submodularity (Golovin & Krause, 2011).

K-Step lookahead policies. These policies plan only a few steps ahead. Specifically, any action a_t in (5) affects the next K rewards, $\mathcal{R}_t^K = [\min\{t+K-1, n\}] \setminus [t-1]$ for all $t \in [n]$. Therefore, any reward r_t is affected only by the K most recent actions, $\mathcal{S}_t = [t] \setminus [\max\{t-K, 0\}]$. By Lemma 2, we have that the gradient of K-step lookahead policies concentrates faster than that of full policies and slower than that of greedy policies, because $\mathcal{S}_t^{\text{greedy}} \subseteq \mathcal{S}_t^K \subseteq \mathcal{S}_t^{\text{full}}$ for any $K \geq 1$. Following the same argument, the complexity of the K-step lookahead policies increases and their concentration rate decreases as K increases. Therefore, the policies are a natural middle ground between being greedy and full planning. We call PPG with $\mathcal{S}_t^K$ by K-Step-PG. For a given lookahead, such as $K = 3$, we write 3-Step-PG. Note that 1-Step-PG is identical to GreedyPG. Our paper is the first work that proposes this policy in LLMs.

Relation to GRPO. GRPO is a policy gradient with an empirically estimated group baseline and, off-policy, propensity weighting; on-policy the weights cancel in the gradient. Our offline PG with group-standardized advantages (Equation (10), Section B.2) is therefore an offline GRPO-style estimator, and PG is its full-planning instance. PPG is orthogonal to the choice of optimizer: any reward subset $\mathcal{R}_t$ can be used inside PPO or GRPO, and PPG can be viewed as providing better temporal credit assignment within a GRPO-style estimator.

5 Experiments

We describe benchmarks first, experimental setup in Section 5.1, and results in Section 5.2.

Problems. We experiment with two problem settings. The first is maintaining consistent and faithful human personas over time in role-playing dialogues (Abdulhai et al., 2025). The second is training a tool-calling agent that follows a operational policy guidelines across multi-turn conversations without drifting from operational rules (Blakeman et al., 2025). We instantiate the framework presented in Section 2 by defining the context x as the persona definition in the persona-alignment task and as operational policy guidelines in the tool-calling task; in both settings, the trajectory τ_n represents the dialogue history where agent responses constitute actions a_t and environment replies serve as observations y_t .

Datasets. We experiment with two datasets. **(1) Consistent-LLMs benchmark** (Abdulhai et al., 2025) provides human-persona dialogues across two domains: (i) **Therapy:** Counseling sessions where a patient agent maintains consistent mental health backgrounds while the therapist serves as the environment; (ii) **Chatting:** Casual dialogue where a simulated human is the agent maintaining biographical consistency (family, career, and personal history) against a conversational environment. Each domain comprises of 6 500 trajectories (5 200 training and 1 300 test) of varying lengths (2 to 60 steps, with 23 steps on average). The trajectories are generated using a Llama-3.1-8B model. **(2) Nemotron-Agentic-v1 Dataset** [1] Blakeman et al. (2025) consists of natural language tool call trajectories in customer service domains where the core task is training an agent to follow an operational policy document across a multi-turn conversation, where the agent must avoid skipping required preconditions or violating procedural constraints. It comprises of 8 000 trajectories (7 200 training and 800 test) of varying lengths (5 to 25 steps, with 18 steps on average). The trajectories are generated using Qwen3-235B-A22B-Instruct-2507 model.

5.1 Experimental Setup

Metrics. The performance of the agent is measured by *prompt consistency (PC)* (Abdulhai et al., 2025), which measures alignment between the agent's response a_t and the assigned context x (persona or policy). We define the per-step reward as $r_t = \text{PC}(x, a_t)$. The reward is computed by a GPT-4o-mini LLM judge for each response a_t. It is 1 when a_t is consistent and 0 otherwise. We report mean consistency scores across all trajectories. We consider two evaluation protocols: **Rollout evaluation:** Following Abdulhai et al. (2025), we generate complete conversation trajectories starting with only the initial context, alternating between responses of the agent and a fixed simulated environment implemented using Llama-3.1-70B model. This exposes the agent to distribution shifts, as inconsistencies compound over time, making it a challenging test of maintaining long-horizon consistency. **Step-level evaluation:** To disentangle consistency improvements from error compounding, we introduce a controlled evaluation setting where each step is assessed independently. Specifically, we take the test trajectories and for each step in them, we generate the response of the agent conditioned on the history of that trajectory, thus isolating the model's inherent consistency capability from the cascading effects of its own errors. See Section B.1 for evaluation prompts.

Baselines. (i) `Base` is an unmodified base model used zero-shot to show what can be attained without learning. (ii) PPO Schulman et al. (2017) was used as an online policy optimization algorithm in Abdulhai et al. (2025). We include it as an *external online reference*: PPO is trained online while all PG variants are trained offline on the same logged data.

Compared methods. We learn a full planning policy (`PG`), greedy policy (`GreedyPG`), and K-step lookahead policies (`K-Step-PG`) for $K \in \{2, 3, 4, 5\}$. The K-step lookahead policies are a rich class of policies between greedy and full planning. They are expected to perform well in our problems because greedy policies are likely suboptimal in enforcing consistency while planning for many steps ahead unnecessarily increases the variance of policy gradients. We learn all policies offline using the offline policy gradient in Algorithm 1, where π_0 samples trajectories uniformly at random from the training set. We decided for offline learning because both of our problems come with datasets. This also enhances reproducibility. All rewards are standardized: we subtract their mean, divide by their standard deviation, and turn them into advantages. `PG`, `GreedyPG`, and `K-Step-PG` share the same logged trajectories, reward labels, optimizer, learning-rate schedule, batch construction, training budget, and evaluation protocol; the only component that changes is the reward subset $\mathcal{R}_t$ (Section B.2). PPO is tuned separately on the same datasets.

Settings. We experiment with Llama-3.1-8B-Instruct and Qwen3-8B, which we abbreviate as `Llama` and `Qwen`, respectively. All results use a single fixed random seed (123), following Abdulhai et al. (2025). We also repeated the full training and evaluation pipeline three times with this seed; these repeats capture only nondeterminism from hardware and floating-point operations, not seed variability. Multi-seed training was infeasible due to the high computational cost of fine-tuning larger models. See Section B.2 for additional training details and hyper-parameters.

5.2 Results

We start by evaluating all policies across all our problems, and report their average prompt consistency in Table 1. While the best policy is domain-dependent, policy gradients typically outperform `Base`.

[1] https://huggingface.co/datasets/nvidia/Nemotron-Agentic-v1

	Qwen						Llama					
	Agentic Tool-calling		Therapy		Chatting		Agentic Tool-calling		Therapy		Chatting	
Policy	Rollout	Step	Rollout	Step	Rollout	Step	Rollout	Step	Rollout	Step	Rollout	Step
Base	0.474	0.633	0.561	0.869	0.760	0.775	0.467	0.495	0.465	0.465	0.478	0.478
PPO	0.480	0.645	0.753	0.911	0.818	0.845	0.478	0.545	0.625	0.679	0.556	0.558
GreedyPG	0.489	0.657	0.840	0.882	0.762	0.881	0.472	0.605	0.685	0.710	0.799	<u>0.850</u>
2-Step-PG	0.504	0.678	0.854	<u>0.936</u>	0.782	0.813	0.488	0.612	<u>0.712</u>	0.734	**0.929**	**0.939**
3-Step-PG	0.535	0.693	**0.937**	**0.940**	**0.985**	**0.995**	0.504	0.628	**0.780**	**0.784**	<u>0.821</u>	0.835
4-Step-PG	0.581	0.712	0.835	0.830	0.831	0.925	0.523	0.650	0.673	0.687	0.692	0.715
5-Step-PG	**0.705**	**0.629**	<u>0.869</u>	0.890	<u>0.958</u>	<u>0.964</u>	**0.640**	**0.698**	0.690	<u>0.740</u>	0.802	0.824
PG	0.513	0.702	0.785	0.815	0.867	0.813	0.505	0.635	0.586	0.595	0.727	0.771

Table 1: Prompt Consistency (PC ↑) across persona alignment (therapy and chatting domains) and agentic tool-calling tasks with Qwen and Llama models, under rollout and step-level evaluation. **Bold** / <u>underline</u> denotes the best / second-best results.

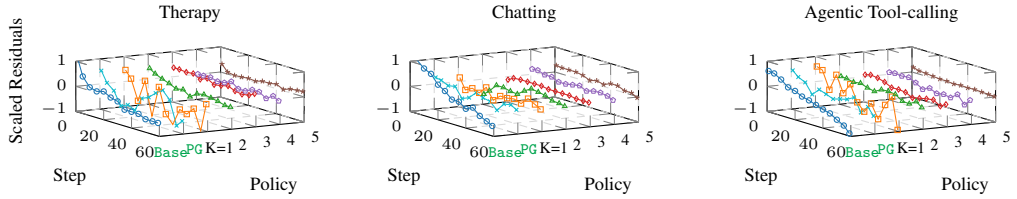


Figure 2: Residual of prompt consistency scaled to $[-1, 1]$ across trajectory steps for persona alignment (therapy and chatting domains) and agentic tool-calling tasks, with Qwen. $K = 1$ represents GreedyPG and $K \in \{2, 3, 4, 5\}$ represents K-Step-PG.

Moreover, in each experiment, there exists a K-step lookahead policy that outperforms PPO. For persona alignment in therapy and chatting domains, K-step lookahead policies (2-Step-PG and 3-Step-PG) perform extremely well across both Qwen and Llama models. We observe consistent results across rollout and step-level evaluation methodologies, suggesting that our gains are robust despite the cascading errors commonly observed in long dialogues. Our results show similar trends for both Qwen and Llama models, suggesting that our approach generalizes across multiple LLM architectures. In customer service agentic tool calling, we observe that 5-Step-PG is superior to both greedy and full planning: greedy policies fail to enforce more complex constraints while full planning is not needed because many constraints can be enforced by planning for just a few steps.

Statistical efficiency gains of simpler policies. The most interesting aspect of Table 1 is that very simple policies outperform full planning. Since all policies are trained on the same offline data using the same partial policy gradient in (5), just with a different subset of future rewards $\mathcal{R}_t$, the difference must come from how reliably the gradient can be estimated. In Section 4.2, we prove a concentration bound that is tighter for policies with smaller reward subsets. To test this directly, we measure the variance of the empirical gradient. At training step 250 (the midpoint of training), we estimate the across-mini-batch variance $\mathrm{Var}(\hat{g}_K) = \frac{1}{d(M-1)} \sum_{m=1}^{M} \|\hat{g}_K^{(m)} - \bar{g}_K\|_2^2$ from $M = 50$ independent mini-batches, where $\bar{g}_K$ is their mean and d is the number of parameters. We repeat this $R = 10$ times with fresh mini-batches for each model and K, and report the mean and standard error in Table 6 (protocol in Section D.7). For both models and all three domains, the variance increases monotonically with K, and full planning has about 11 to 13 times the variance of GreedyPG. Llama is trained with a smaller batch (64 vs. 128), so its variance is higher; we do not compare absolute values across models. This matches the data-scaling results in Section D.2, where GreedyPG is best with little data and longer horizons win only as data grows. The variance measurement is taken at a single checkpoint, so it supports the variance mechanism but does not by itself prove that variance explains the gains in Table 1; other factors are not ruled out.

On Effectiveness of Simpler Policies. Now we try to understand the best performing policies. For all policies, we compute the mean prompt consistency at each conversation step, subtract the mean over all steps, and renormalize to $[-1, 1]$ jointly over all policies. This is just a linear transformation of consistency scores that makes them comparable across the policies. We call this quantity a *residual* and report it in Figure 2 for persona alignment task. The residual of Base monotonically decreases

as conversation progresses, confirming that LLMs are susceptible to persona drift in long-horizon conversations (Nie et al., 2021). The residual of GreedyPG oscillates a lot. These ripples reflect the policy's repeated failed recovery attempts from persona drift, due to not planning for the future. In comparison, K-step lookahead policies (K-Step-PG) maintain stable residuals across trajectory steps. We observe that increasing K from $K = 1$ (GreedyPG) to $K = 3$ (K-Step-PG) produces progressively smoother residuals, supporting our theory that assigning credit to longer horizons can prevent persona drift by maintaining consistency continuously over time. Finally, in therapy and chatting domains, we observe that PG oscillates a lot and thus performs worse than 3-Step-PG. This experiment shows that the temporal scope of credit assignment directly influences policy stability, with longer horizons providing smoother and more robust behavior at the cost of increased data requirements. See additional results for Llama in Section D.4, statistical efficiency analysis of PPG in Section D.2, scaling laws for PPG in Section D.3, stability of PPG across trajectory lengths in Section D.6, and qualitative analysis in Section E.

Choosing K. The optimal horizon behaves like model complexity: simple policies are preferred in low-data regimes, intermediate horizons dominate at moderate data, and full planning wins only with abundant data (Section D.3). Because the optimum shifts smoothly, we recommend evaluating a logarithmically spaced set of candidates (e.g., $K \in \{1, 2, 4, 8, \dots\}$) on a small validation set, which needs only $O(\log n)$ evaluations instead of a sweep over all horizons.

Robustness to the judge. Rewards and evaluation both use GPT-4o-mini. To check for judge-specific effects, we re-scored rollouts of the baselines and best policies with Claude-3.5-Sonnet and Gemini-1.5-Pro (Section D.8). Pairwise agreement on the binary decisions is 95.7 to 98.6% (Cohen's κ between 0.90 and 0.94), and the ranking of methods is preserved.

6 Related Work

The closest related works are Kveton et al. (2025) and Guo et al. (2025). Kveton et al. (2025) optimized greedy policies using online policy gradients and Guo et al. (2025) optimized segment policies using GRPO. Our formalism captures both of these policy classes (Sections A.5 and 4.3). Unlike Kveton et al. (2025), we consider more general optimization problems than adaptive submodularity (Golovin & Krause, 2011). Unlike Guo et al. (2025), we consider a general RL model that allows observations of the environment. Learning of policies in finite-horizon Markov decision processes is a classic topic and it is well known that they are non-stationary (Puterman, 1994). Monte Carlo estimation of Q values, by the sum of infinite-horizon discounted rewards, is discussed in detail in Sutton & Barto (1998) and was generalized to advantages by Schulman et al. (2016). We do not discount rewards and show that maximization of a subset of future rewards results in a new simpler policy that is easier to learn. Multi-step lookahead policies were studied before: Efroni et al. (2018a) used them to improve policy iteration, Efroni et al. (2018b) applied them to online and approximate RL, and Efroni et al. (2020) extended them to real-time dynamic programming. These works focus on policy iteration and dynamic programming, none of which are practical for RL in LLMs. The main technical difference in our work is that we use and analyze policy gradients. Adaptive lookahead polices were studied by Rosenberg et al. (2023) and K-step lookahead policies were used in offline RL by Sikchi et al. (2022). The main difference in our work is that we only use policy gradients without any additional model, such as that of a value or Q function.

7 Conclusions

Most popular RL algorithms in LLMs are variants of policy gradients. In this work, we propose a natural approach for trading off the bias and variance in policy gradients. The key idea is to optimize for a subset of future rewards: smaller subsets define simpler policies, which can be learned more efficiently because their empirical gradients have lower variance. We analyze the policies, and propose both online and offline algorithms for learning them. Full planning, greedy, K-step lookahead policies are all instances of our framework. We evaluate the policies in conversational persona-alignment and tool calling problems. We observe that different policies excel in different domains, highlighting the importance of our studied trade-off. We note that the idea of optimizing for subsets of future rewards is general and can be straightforwardly applied to regularized policies (Todorov, 2006) and also GRPO (Shao et al., 2024). We plan extensions to other applications, such as reasoning and coding, as future work.

AI USE STATEMENT

In this work, generative AI tools were used only as general-purpose assistants for editing prose (grammar, phrasing, and formatting) and for minor coding support in the experimental pipeline. Generative AI tools were not used to generate research ideas, derive the theoretical results, design the algorithms, or produce the experimental results and analyses reported in the paper. All AI-assisted text and code were reviewed and verified by the authors, and we take full responsibility for the final content of this work, including all text, claims, and artifacts. We note that large language models are also the object of study in our experiments; their use as evaluated systems is described in Section 5.

ETHICS STATEMENT

This work studies reinforcement learning algorithms for large language models and does not involve new data collection from human subjects. All experiments are conducted on publicly available benchmarks (Abdulhai et al., 2025; Blakeman et al., 2025), whose public links are provided in Section B.3. Our study of persona alignment and agentic tool-calling is aimed at making language-model behavior more consistent and more faithful to specified operational policies, which we view as a step toward more reliable and controllable systems. As with any method that shapes model behavior, the same techniques could in principle be used to align models to undesirable objectives; we do not introduce any new capability that is specific to harmful use. We do not foresee additional ethical concerns regarding conflicts of interest, discrimination, privacy, or legal compliance beyond those already inherent to the public datasets we use.

REPRODUCIBILITY STATEMENT

We have made an effort to ensure that our results are reproducible. All theoretical claims are stated with their full assumptions in the main text (Section 4) and proved completely in Sections A.1 to A.4. Our experiments use only public benchmarks (Abdulhai et al., 2025; Blakeman et al., 2025), with dataset links in Section B.3. The experimental setup is described in Section 5, and the complete set of hyperparameters, training details, and evaluation prompts is provided in Sections B, B.1 and B.2. Our learning algorithm is a form of policy gradient and is straightforward to implement from the description in Section 3.

REFERENCES

Marwa Abdulhai, Ryan Cheng, Donovan Clay, Tim Althoff, Sergey Levine, and Natasha Jaques. Consistently simulating human personas with multi-turn reinforcement learning. *arXiv preprint arXiv:2511.00222*, 2025.

V. M. Aleksandrov, V. I. Sysoyev, and V. V. Shemeneva. Stochastic optimization. *Engineering Cybernetics*, 5:11–16, 1968.

Jonathan Baxter and Peter Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Jonathan Baxter, Peter Bartlett, and Lex Weaver. Experiments with infinite-horizon, policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:351–381, 2001.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, et al. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2512.20848*, 2025.

Stephane Boucheron, Gabor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Craig Boutilier, Richard Dearden, and Moises Goldszmidt. Exploiting structure in policy construction. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 1104–1111, 1995.

Daniela Pucci de Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–856, 2003.

Yonathan Efroni, Gal Dalal, Bruno Scherrer, and Shie Mannor. Beyond the one-step greedy approach in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018a.

Yonathan Efroni, Gal Dalal, Bruno Scherrer, and Shie Mannor. Multiple-step greedy policies in approximate and online reinforcement learning. In *Advances in Neural Information Processing Systems 31*, 2018b.

Yonathan Efroni, Mohammad Ghavamzadeh, and Shie Mannor. Online planning with lookahead policies. In *Advances in Neural Information Processing Systems 33*, 2020.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

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

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in RL for large language models. In *Advances in Neural Information Processing Systems 38*, 2025.

Milos Hauskrecht and Branislav Kveton. Linear program approximations for factored continuous-state Markov decision processes. In *Advances in Neural Information Processing Systems 16*, pp. 895–902, 2004.

Pegah Jandaghi, XiangHai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. Faithful persona-based conversational dataset generation with large language models. In *Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024)*, pp. 114–139, 2024.

Natasha Jaques, Judy Hanwen Shen, Asma Ghandeharioun, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Human-centric dialog training via offline reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

Ke Ji, Yixin Lian, Linxu Li, Jingsheng Gao, Weiyuan Li, and Bin Dai. Enhancing persona consistency for LLMs' role-playing using persona-aware contrastive learning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26221–26238, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1344. URL https://aclanthology.org/2025.findings-acl.1344/.

Daphne Koller and Ronald Parr. Computing factored value functions for policies in structured MDPs. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 1332–1339, 1999.

Branislav Kveton, Anup Rao, Viet Dac Lai, Nikos Vlassis, and David Arbour. Adaptive submodular policy optimization. *Reinforcement Learning Journal*, 6:2720–2736, 2025.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Subhojyoti Mukherjee, Viet Dac Lai, Raghavendra Addanki, Ryan Rossi, Seunghyun Yoon, Trung Bui, Anup Rao, Jayakumar Subramanian, and Branislav Kveton. Offline RL by reward-weighted fine-tuning for conversation optimization. In *Advances in Neural Information Processing Systems 38*, 2025.

Remi Munos. Geometric variance reduction in Markov chains: Application to value function and gradient estimation. *Journal of Machine Learning Research*, 7:413–427, 2006.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. I like fish, especially dolphins: Addressing contradictions in dialogue modeling. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pp. 1699–1713, 2021.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35*, 2022.

Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, NY, 1994.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36*, 2023.

Maxim Raginsky and Igal Sason. Concentration of measure inequalities in information theory, communications and coding. *CoRR*, abs/1212.4663, 2012. URL http://arxiv.org/abs/1212.4663.

Aviv Rosenberg, Assaf Hallak, Shie Mannor, Gal Chechik, and Gal Dalal. Planning and learning with adaptive lookahead. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, 2023.

Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. Training llm-based tutors to improve student learning outcomes in dialogues. *arXiv preprint arXiv:2503.06424*, 2025.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the 4th International Conference on Learning Representations*, 2016.

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

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

Harshit Sikchi, Wenxuan Zhou, and David Held. Learning off-policy with online planning. In *Proceedings of the 5th Conference on Robot Learning*, 2022.

Richard Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3: 9–44, 1988.

Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Richard Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pp. 1057–1063, 2000.

Emanuel Todorov. Linearly-solvable Markov decision problems. In *Advances in Neural Information Processing Systems 19*, 2006.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Wei-Lin Chen, Chao-Wei Huang, Yu Meng, and Yun-Nung Chen. Two tales of persona in LLMs: A survey of role-playing and personalization. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16612–16631, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.969. URL https://aclanthology.org/2024.findings-emnlp.969/.

John Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

Christopher Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

Ronald Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36*, 2023.

A Additional Technical Contributions

This section contains proofs of our main claims and supporting lemmas.

A.1 Proof of Lemma 1

The proof is based on telescoping,

$$\sum_{t=1}^{n} r_t = \sum_{t=1}^{n} f(x, \tau_t) - f(x, \tau_{t-1}) = f(x, \tau_n) - f(x, \tau_0) = r(x, \tau_n)\,.$$

A.2 Proof of Theorem 4

The main tool used in our analysis is Hoeffding's inequality (Raginsky & Sason, 2012; Boucheron et al., 2013), which we restate below for completeness.

Theorem 5. *Let $X_1, \ldots, X_n$ be n independent random variables such that $|X_i| \leq c_i$ holds almost surely for some $c_i > 0$. Then for any $\varepsilon > 0$,*

$$\mathbb{P}\left(|X_1 + \cdots + X_n - \mathbb{E}\left[X_1 + \cdots + X_n\right]| \geq \varepsilon\right) \leq 2\exp\left[-\frac{\varepsilon^2}{2\sum_{i=1}^{n} c_i^2}\right].$$

We apply Hoeffding's inequality as follows. Let

$$f(\mathcal{B}) = \sum_{i=1}^{B}\sum_{t=1}^{n}\sum_{\ell\in\mathcal{S}_t} r_{i,t}\nabla \log \pi(a_{i,\ell} \mid x_i, \tau_{i,\ell-1}; \theta)\,, \tag{9}$$

where $\mathcal{B}$ is the batch in Algorithm 1 and $\mathcal{S}_t$ is the set of action indices that affect the reward at step t (Section 4.1). Note that $f(\mathcal{B})$ is a d-dimensional vector, where d is the number of policy parameters θ with respect to which the gradient is taken. For any $j \in [d]$, let $f_j(\mathcal{B})$ be the j-th entry of $f(\mathcal{B})$.

We have $f(\mathcal{B}) = Bg(\mathcal{B})$ by swapping $\mathcal{R}_t$ for $\mathcal{S}_t$. Using norm identities and the union bound,

$$\begin{aligned}
\mathbb{P}\left(\|g(\mathcal{B}) - \mathbb{E}\left[g(\mathcal{B})\right]\|_2 \geq \varepsilon\right) &= \mathbb{P}\left(\|f(\mathcal{B}) - \mathbb{E}\left[f(\mathcal{B})\right]\|_2 \geq \varepsilon B\right) \\
&\leq \mathbb{P}\left(\sqrt{d}\max_{j\in[d]} |f_j(\mathcal{B}) - \mathbb{E}\left[f_j(\mathcal{B})\right]| \geq \varepsilon B\right) \\
&\leq \sum_{j=1}^{d} \mathbb{P}\left(|f_j(\mathcal{B}) - \mathbb{E}\left[f_j(\mathcal{B})\right]| \geq \frac{\varepsilon B}{\sqrt{d}}\right).
\end{aligned}$$

Now note that $f_j(\mathcal{B})$ is a sum of $B\sum_{t=1}^{n}|\mathcal{S}_t|$ random scalars but $\mathcal{S}_t$ is fixed. Using our assumptions, we have $|r_{i,t}| \leq c_t$ and $|\nabla \log \pi(a_{i,t} \mid x_i, \tau_{i,t-1}; \theta))_j| \leq 1$. Then

$$\left|\sum_{t=1}^{n}\sum_{\ell\in\mathcal{S}_t} r_{i,t}(\nabla \log \pi(a_{i,\ell} \mid x_i, \tau_{i,\ell-1}; \theta))_j\right| \leq \sum_{t=1}^{n}\sum_{\ell\in\mathcal{S}_t} c_t$$

holds jointly for all $i \in [B]$ and $j \in [d]$. By Theorem 5, for any $j \in [d]$ and $\varepsilon > 0$,

$$\begin{aligned}
\mathbb{P}\left(|f_j(\mathcal{B}) - \mathbb{E}\left[f_j(\mathcal{B})\right]| \geq \frac{\varepsilon B}{\sqrt{d}}\right) &\leq 2\exp\left[-\frac{\varepsilon^2 B^2}{2dB\left(\sum_{t=1}^{n}\sum_{\ell\in\mathcal{S}_t} c_t\right)^2}\right] \\
&\leq 2\exp\left[-\frac{\varepsilon^2 B}{2d\left(\sum_{t=1}^{n}\sum_{\ell\in\mathcal{S}_t} c_t\right)^2}\right].
\end{aligned}$$

Finally, we chain all inequalities and get

$$\mathbb{P}\left(\|g(\mathcal{B}) - \mathbb{E}\left[g(\mathcal{B})\right]\|_2 \geq \varepsilon\right) \leq 2d\exp\left[-\frac{\varepsilon^2 B}{2d\left(\sum_{t=1}^{n}\sum_{\ell\in\mathcal{S}_t} c_t\right)^2}\right].$$

This completes the proof.

A.3 Proof of Lemma 2

Using rules of probability,

$$\mathbb{E}\left[\sum_{t=1}^{n}\left(\sum_{\ell\in\mathcal{R}_t} r_\ell\right)\nabla\log\pi(a_t \mid x,\tau_{t-1};\theta)\right] = \mathbb{E}\left[\sum_{t=1}^{n} r_t \sum_{\ell\in\mathcal{S}_t}\nabla\log\pi(a_\ell \mid x,\tau_{\ell-1};\theta)\right]$$
$$= \sum_{t=1}^{n}\mathbb{E}\left[r_t\sum_{\ell\in\mathcal{S}_t}\nabla\log\pi(a_\ell \mid x,\tau_{\ell-1};\theta)\right] = \sum_{t=1}^{n}\mathbb{E}\left[\mathbb{E}_t\left[r_t\sum_{\ell\in\mathcal{S}_t}\nabla\log\pi(a_\ell \mid x,\tau_{\ell-1};\theta)\right]\right]$$
$$= \sum_{t=1}^{n}\mathbb{E}\left[\nabla\mathbb{E}_t\left[r_t\right]\right] = \mathbb{E}\left[\sum_{t=1}^{n}\nabla\mathbb{E}_t\left[r_t\right]\right].$$

We swap $\mathcal{R}_t$ for $\mathcal{S}_t$ in the first step, the linearity of expectation in the second step, the tower rule in the third step, the score identity over random $\tau_{\mathcal{S}_t}$ in the fourth step, and the tower rule in the last step. This completes the proof.

A.4 Proof of Lemma 3

Using the tower rule and π_0 does not depend on the differentiated θ, we get

$$\mathbb{E}_0\left[\sum_{t=1}^{n}\left(\sum_{\ell\in\mathcal{R}_t} r_\ell\right)\nabla\log\pi(a_t \mid x,\tau_{t-1};\theta)\right] = \mathbb{E}_0\left[\sum_{t=1}^{n} r_t \sum_{\ell\in\mathcal{S}_t}\nabla\log\pi(a_\ell \mid x,\tau_{\ell-1};\theta)\right]$$
$$= \sum_{t=1}^{n}\mathbb{E}_0\left[r_t\sum_{\ell\in\mathcal{S}_t}\nabla\log\pi(a_\ell \mid x,\tau_{\ell-1};\theta)\right] = \sum_{t=1}^{n}\mathbb{E}_0\left[\mathbb{E}_{0,t}\left[r_t\sum_{\ell\in\mathcal{S}_t}\nabla\log\pi(a_\ell \mid x,\tau_{\ell-1};\theta)\right]\right]$$
$$= \sum_{t=1}^{n}\mathbb{E}_0\left[\nabla\mathbb{E}_{0,t}\left[r_t\sum_{\ell\in\mathcal{S}_t}\log\pi(a_\ell,y_\ell \mid x,\tau_{\ell-1};\theta)\right]\right]$$
$$= \mathbb{E}_0\left[\sum_{t=1}^{n}\nabla\mathbb{E}_{0,t}\left[r_t\sum_{\ell\in\mathcal{S}_t}\log\pi(a_\ell,y_\ell \mid x,\tau_{\ell-1};\theta)\right]\right].$$

This completes the proof.

A.5 K-Segment Policy Gradient

Let $\mathcal{C}_1,\ldots,\mathcal{C}_K \subset [n]$ be sets of consecutive step indices representing segments, which are mutually exclusive and exhaustive. Let $m_k = \min\mathcal{C}_k$ and $M_k = \max\mathcal{C}_k$ be the minimum and maximum indices within segment $\mathcal{C}_k$. Let k_t be the segment of step t. Then $\mathcal{S}_t = [t] \setminus [m_{k_t}-1]$ are the future rewards in segment $\mathcal{C}_{k_t}$ starting from step t. By Lemma 2, all actions up to step t within its segment $\mathcal{C}_{k_t}$ affect the reward at step t; and the gradient is a sum of gradients of the corresponding expected rewards. This policy class was proposed in LLMs by Guo et al. (2025).

This policy is simpler than PG, where the reward is affected by all previous actions, but also more complex than GreedyPG, where the reward is affected only by the immediate action. Since

$$\{t\} \subseteq \mathcal{S}_t = [t] \setminus [m_{k_t}-1] \subseteq [t]$$

holds for any segments $\mathcal{C}_1,\ldots,\mathcal{C}_K$, we have by Theorem 4 that the empirical gradient of segment policies concentrates faster than that of PG but slower than that of GreedyPG. The rate depends on the size of the segments.

B Experiment Settings

B.1 Evaluation Prompts

This section provides the complete evaluation prompts used for consistency assessment with the LLM judge (`gpt-4o-mini`, temperature=0).

Prompt consistency measures whether an agent's utterance aligns with their assigned guidelines/persona in the start of the conversation as part of the prompt. The judge outputs YES (consistent, score=1) or NO (contradiction, score=0).

System Instructions - Prompt Consistency

Evaluate the intention behind the following line spoken by [SPEAKER_ROLE] and determine whether it contradicts their background (guidelines/persona).
First, describe the interpreted intention of the statement, and whether or not it aligns with the given background of [SPEAKER_ROLE].
Then, answer **NO** if the line contradicts the given background of [SPEAKER_ROLE] or the intention does not align with the provided background, and answer **YES** if it does align with the provided background or the intention aligns with the background of [SPEAKER_ROLE].
Provide your answer as 1 sentence explaining your reasoning based on the background and the interpreted intention, followed by YES or NO.

Example Input - Prompt Consistency

[SCENARIO_DESC]: This is a tutoring conversation.
[SPEAKER_ROLE]: Student
[SPEAKER_BACKSTORY]: You are a college freshman who struggles with calculus and prefers visual learning methods over abstract mathematical notation. You have a part-time job and limited study time.
[SPEAKER_LINE]: "I actually find it easier to just memorize the formulas without understanding the visual representations."

Expected Output Format - Prompt Consistency

Judge Response: "The student's statement that they find it easier to memorize formulas without visual representations directly contradicts their stated preference for visual learning methods over abstract notation. **YES**"
Score: 0 (inconsistent with persona)

Template Variables:

- `[SCENARIO_DESC]` – Domain context (e.g., "This is a tutoring conversation")
- `[SPEAKER_ROLE]` – Agent role (e.g., "Student", "Patient", "Agent 2")
- `[SPEAKER_BACKSTORY]` – Full persona/background description assigned to agent
- `[SPEAKER_LINE]` – The specific utterance being evaluated

B.2 Training Setup

Model Training. We employ two 8B-parameter instruction-tuned base models: Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen3-8B (Yang et al., 2025). For computational efficiency, we limit the maximum context length to 4096 tokens during training. Models are fine-tuned for 3 epochs using the AdamW optimizer with a learning rate of 5×10^{-6}, 0.01 weight decay, and a cosine schedule with 3% warmup, and trained using bfloat16 precision. Training is distributed across NVIDIA A100 GPUs using Fully Sharded Data Parallel (FSDP). Llama-3.1 is trained with an effective batch size of 64, while Qwen3 utilizes an effective batch size of 128.

Experiment Hyperparameter. PPO was trained with the following hyperparameters selected using initial experiments where we tried the following values: actor learning rate (`actor_learning_rate`) over $\{10^{-7}, 5 \times 10^{-7}, 10^{-6}, 5 \times 10^{-6}, 10^{-5}\}$, KL penalty (`init_kl_coef`) over $\{0.01, 0.05, 0.1, 0.2\}$, and maximum samples with early stopping (`max_samples`) over $\{50\text{k}, 100\text{k}, 200\text{k}\}$.

The optimal hyperparams for PPO experiments are described in Table 2.

All experiments use a single fixed seed (123) that controls data shuffling, mini-batch sampling, rollout sampling, and parameter initialization. We repeated the full pipeline three times with this seed

Hyperparameter	Value
actor_learning_rate	5×10^{-7}
critic_learning_rate	1×10^{-5}
init_kl_coef	0.01
train_batch_size	128
rollout_batch_size	1024
max_samples	100k
max_epochs	1

Table 2: Hyperparameters for PPO training across persona-alignment and tool-calling tasks.

to measure nondeterminism from GPU kernels, floating-point reduction order, and clipping; these repeats do not measure seed variability. We experiment with two different tasks and three different conversational domains (education, therapy, and chatting). We evaluate two architecturally distinct models (Qwen and Llama) in all settings. The consistency of winning policies across both models provides strong evidence of robustness. Further, we experiment with a new dataset of synthetic chats Jandaghi et al. (2024) on three model architectures (Llama, Qwen, and Gemma), all showing the same scaling as the experiments in the main paper. We observe that many performance gaps are so large that they are unlikely to be explained by the random seed. For example, in the chatting domain with Llama, 2-Step-PG achieves 0.929 versus 0.556 of PPO, a gap of 0.35 on $[0, 1]$ scale where the metrics are averaged over thousands of measurements.

Advantage computation in offline RL. Since Lemma 3 provides guarantees for negative rewards, offline PPG can be implemented with advantages and we do that in Section 5. The advantage for step t in trajectory i is computed as

$$\tilde{r}_{i,t} = \frac{\sum_{\ell \in \mathcal{R}_t} r_{i,\ell} - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}}}, \tag{10}$$

where $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are the mean and standard deviation of $\sum_{\ell \in \mathcal{R}_t} r_{j,\ell}$ computed jointly over $j \in \mathcal{G}$ and $t \in [n]$, where $\mathcal{G}$ is the group of trajectories that trajectory i belongs to. In our experiments in Section 5, we group the trajectories by the persona and topic.

Ablations of learning rate in PPG. We choose the hyperparameters of PPG using initial experiments. The learning rate of PPG is set to 5×10^{-7}. To show robustness to perturbation, we ablate the learning rate for persona alignment task with Qwen model as follows.

Policy	Qwen LR/2	Qwen LR	Qwen LR×2	Llama LR/2	Llama LR	Llama LR×2
GreedyPG	0.824	0.840	0.836	0.671	0.685	0.679
2-Step-PG	0.838	0.854	0.849	0.698	0.712	0.706
3-Step-PG	**0.921**	**0.937**	**0.918**	**0.761**	**0.780**	**0.748**
4-Step-PG	0.818	0.835	0.809	0.651	0.673	0.628
5-Step-PG	0.848	0.869	0.821	0.664	0.690	0.631
PG	0.754	0.785	0.706	0.549	0.586	0.481

Table 3: Learning rate ablation on therapy domain for Qwen and Llama models.

Our learning rate ablation shows robustness: the observed trends are largely consistent with main results and show robustness as the relative performance does not change much when we halve or double our default learning rate of 5×10^{-7}. These trends are largely consistent where a K-Step-PG policy outperforms GreedyPG in each setting, and the best performing K-Step-PG is either slightly better or comparably suboptimal to PG.

B.3 Dataset Sources

We source our datasets from public benchmarks:

Policy	Qwen LR/2	Qwen LR	Qwen LR×2	Llama LR/2	Llama LR	Llama LR×2
GreedyPG	0.748	0.762	0.758	0.779	0.799	0.793
2-Step-PG	0.764	0.782	0.779	**0.911**	**0.929**	**0.921**
3-Step-PG	**0.968**	**0.985**	**0.951**	0.801	0.821	0.784
4-Step-PG	0.809	0.831	0.796	0.664	0.692	0.641
5-Step-PG	0.934	0.958	0.919	0.774	0.802	0.751
PG	0.836	0.867	0.791	0.684	0.727	0.645

Table 4: Learning rate ablation on chatting domain for Qwen and Llama models.

Policy	Qwen LR/2	Qwen LR	Qwen LR×2	Llama LR/2	Llama LR	Llama LR×2
GreedyPG	0.601	0.623	0.578	0.711	0.739	0.698
2-Step-PG	0.911	0.930	0.897	0.698	0.721	0.681
3-Step-PG	0.935	0.948	0.921	0.551	0.573	0.498
4-Step-PG	0.951	0.976	0.901	0.812	0.837	0.779
5-Step-PG	0.863	0.881	0.831	0.748	0.775	0.711
PG	**0.958**	**0.971**	**0.934**	**0.871**	**0.913**	**0.798**

Table 5: Learning rate ablation on education domain for Qwen and Llama models.

	Qwen			Llama		
Policy	Therapy	Chatting	Tool-calling	Therapy	Chatting	Tool-calling
GreedyPG	0.044 ± 0.003	0.037 ± 0.003	0.048 ± 0.002	0.085 ± 0.006	0.072 ± 0.003	0.090 ± 0.004
2-Step-PG	0.087 ± 0.005	0.071 ± 0.007	0.095 ± 0.005	0.166 ± 0.017	0.143 ± 0.010	0.185 ± 0.007
3-Step-PG	0.153 ± 0.012	0.133 ± 0.013	0.168 ± 0.012	0.297 ± 0.020	0.256 ± 0.014	0.354 ± 0.019
4-Step-PG	0.199 ± 0.018	0.184 ± 0.018	0.226 ± 0.018	0.404 ± 0.025	0.360 ± 0.024	0.409 ± 0.026
5-Step-PG	0.283 ± 0.022	0.258 ± 0.027	0.298 ± 0.025	0.539 ± 0.036	0.480 ± 0.033	0.537 ± 0.033
PG	0.505 ± 0.032	0.455 ± 0.033	0.518 ± 0.036	0.920 ± 0.062	0.967 ± 0.051	1.071 ± 0.059

Table 6: Gradient estimator variance $\mathrm{Var}(\hat{g}_K)$ ($\times 10^{-5}$) at training step 250. Mean $\pm$ standard error over $R = 10$ repetitions, each with $M = 50$ fresh mini-batches.

1. Consistent-LLMs (Abdulhai et al., 2025): https://github.com/abdulhaim/consistent-LLMs
2. Nemotron Agentic-v1 (Blakeman et al., 2025): https://huggingface.co/datasets/nvidia/Nemotron-Agentic-v1
3. Synthetic Persona Chat (Jandaghi et al., 2024): https://github.com/google-research-datasets/Synthetic-Persona-Chat

C LIMITATIONS

Our experiments support but do not formally prove that the observed gains are due to lower gradient variance: we measure variance at a single checkpoint (Table 6), and other factors are not ruled out. The bias from truncating future rewards is bounded only loosely. The per-step reward in (4) is a design choice that we do not ablate. Rewards and evaluation both use GPT-4o-mini; we checked two other judges on a subset of methods but ran no human study. All results use a single seed. PPO is trained online while the PG variants are trained offline, so that comparison is not controlled, and the K trade-off is shown only in the offline setting. Our tasks have decomposable per-step consistency rewards, and the concentration bound has a factor d that is vacuous at LLM scale.

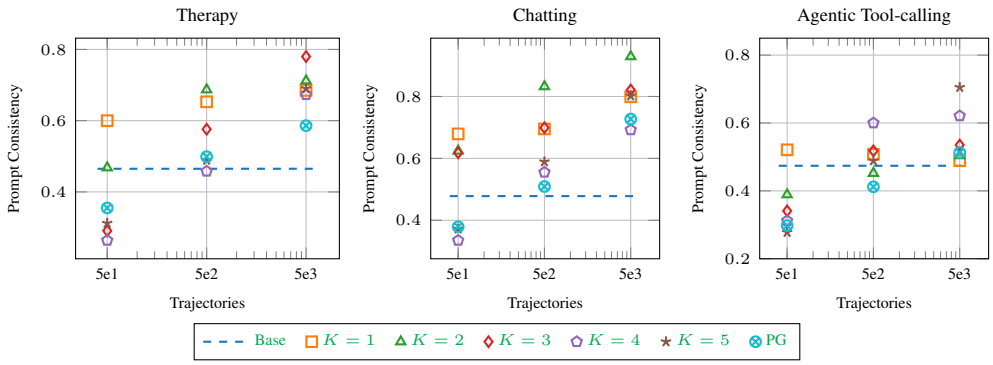


Figure 3: Prompt consistency as a function of sample size (number of training trajectories) for persona alignment (therapy and chatting domains) and agentic tool-calling tasks with `Llama`. $K = 1$ represents `GreedyPG` and $K \in \{2, 3, 4, 5\}$ represents `K-Step-PG`.

D Additional Results

D.1 Direct evidence of lower gradient variance.

At training step 250 (the midpoint of training), we estimate the across-mini-batch variance $\mathrm{Var}(\hat{g}_K) = \frac{1}{d(M-1)} \sum_{m=1}^{M} \|\hat{g}_K^{(m)} - \bar{g}_K\|_2^2$ of the empirical gradient from $M = 50$ independent mini-batches, where $\bar{g}_K$ is their mean and d is the number of parameters. We repeat this $R = 10$ times with fresh mini-batches for each model and K, and report the mean and standard error in Table 6 (protocol in Section D.7). For both models and all three domains, the variance increases monotonically with K, and full planning has about 11 to 13 times the variance of `GreedyPG`. Llama is trained with a smaller batch (64 vs. 128), so its variance is higher; we do not compare absolute values across models. This is a measurement at one checkpoint. It supports the variance mechanism but does not by itself prove that variance explains the gains in Table 1.

D.2 On Statistical Efficiency of Simpler Policies

We investigate the statistical efficiency of learning different policies next. Specifically, we vary the number of training trajectories, from 50 (low-data regime, 1% of training data) to 5 000 (high-data regime, all training data), and plot the prompt consistency of all policies in Figure 3. Our results reveal a fundamental trade-off between policy complexity and statistical efficiency. In the low-data regime, `GreedyPG` consistently excels in all domains, confirming our hypothesis that simpler policies are easier to learn when data are scarce. As data scales to moderate levels (500 trajectories, 10% of training data), K-step lookahead policies dominate across all domains. In particular, `2-Step-PG` and `3-Step-PG` are the best policies in therapy and chatting domains. These policies strike balance between planning enough for the future and being able to learn it. `5-Step-PG` achieves the best results on the tool-calling agent setup, suggesting that planning multiple steps ahead helps the model avoids compliance violations in conversations with complex task constraints.

D.3 Scaling Laws for Partial Policy Gradients

Motivation. In Figure 3, we demonstrated that simpler policies (`GreedyPG`) exhibit superior statistical efficiency in low-data regimes compared to complex policies (`PG`), with the performance gap narrowing as training data increases. This raises a practical question: *how does the optimal lookahead horizon K scale with available training data across different model architectures?* Understanding this scaling relationship is critical for practitioners who must choose appropriate credit assignment horizons given their data budget constraints.

Experimental Design. To investigate this scaling behavior systematically, we replicate the statistical efficiency experiment from the main paper using a complementary dataset: the Synthetic Persona Chat dataset (Jandaghi et al., 2024), which contains generic and mixed-domain personas distinct from our primary Education, Therapy, and Chatting domains. We sample 20,000 trajectories from this dataset

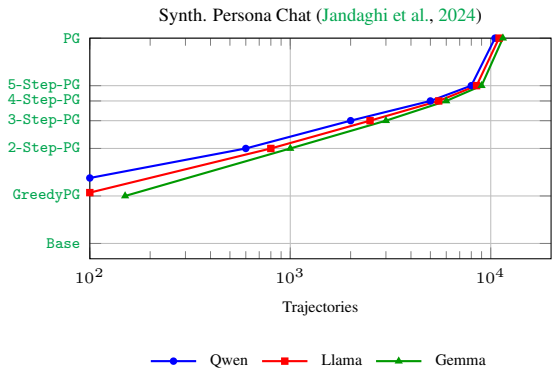


Figure 4: Scaling Laws for PPG: Across different LLMs (Llama-3.1-8B-Instruct, Qwen3-8B, and Gemma-7B-it), optimal value of lookahead K in K-Step-PG scales with available training trajectories.

and train all policy variants (Base, GreedyPG, 2-Step-PG, 3-Step-PG, 4-Step-PG, 5-Step-PG, PG) with varying amounts of training data, ranging from 100 to 20,000 trajectories. For each data size, we measure which policy achieves the best prompt consistency on a held-out test set. We repeat this procedure across three different LLM architectures—Llama-3.1-8B-Instruct, Qwen3-8B, and Gemma-7B-it, to assess whether the scaling laws are architecture-dependent or represent a general phenomenon.

Observation. Figure 4 shows the optimal policy (represented by its lookahead horizon K on the y-axis) as a function of available training trajectories (x-axis) for each model architecture. The key pattern is a consistent "staircase" structure: at low data regimes (50-1000 trajectories), Base or GreedyPG are optimal; as data increases to intermediate regimes (1000-5000 trajectories), K-Step-PG with K=2,3,4 successively become optimal; and at high data regimes (>10,000 trajectories), full planning (PG) emerges as the best policy. Remarkably, this pattern holds across all three architectures, with only minor differences in the inflection points where transitions occur.

Interpretation. The results in Figure 4 demonstrate that the scaling behavior of K-step lookahead policies is remarkably consistent across different model architectures. We observe a characteristic trend where the optimal complexity of the policy, represented by the lookahead K, is a direct function of the available data volume. In low-data regimes, complex policies like full planning do not work effectively, and simpler objectives like GreedyPG or low K are preferred to achieve stable performance. However, as the number of trajectories increases beyond a critical threshold, higher-order policies become more effective, and full planning PG eventually emerges as the best policy. This cross-architecture consistency validates our theoretical framework (Theorem 4): while simpler partial gradients concentrate faster, they optimize suboptimal policies, whereas complex policies require sufficient data to overcome higher variance in gradient estimation. These results establish a practical design principle for policy optimization in RL for LLMs: practitioners should calibrate policy complexity to available data, initializing with greedy policies in data-scarce regimes and progressively expanding the credit assignment horizon as data accumulates.

D.4 Stability of policy gradients for Llama3.1-8B model

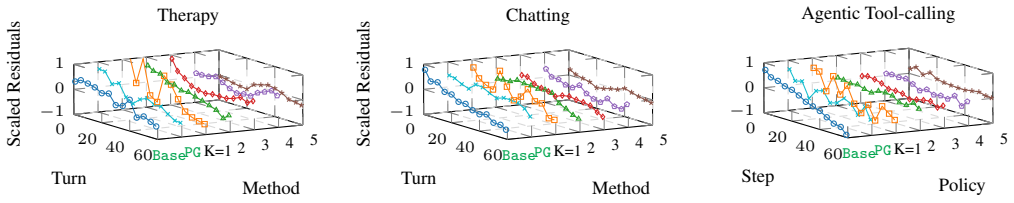


Figure 5: Residual of Prompt consistency metric scaled to $[-1, 1]$ across trajectory steps for persona alignment (therapy and chatting domains) and agentic tool-calling tasks using Llama. K=1 represents GreedyPG; K={2,3,4,5} represent K-Step-PG.

	Qwen			Llama	
	Rollout	Step-level		Rollout	Step-level
Policy	PC ↑	PC ↑	Policy	PC ↑	PC ↑
Base	0.799	0.886	Base	0.425	0.425
PPO	0.875	0.895	PPO	0.613	0.624
GreedyPG	0.623	0.930	GreedyPG	0.739	0.833
2-Step-PG	0.930	0.936	2-Step-PG	0.721	0.759
3-Step-PG	0.948	0.960	3-Step-PG	0.573	0.645
4-Step-PG	<u>0.971</u>	<u>0.954</u>	4-Step-PG	<u>0.837</u>	0.858
5-Step-PG	0.881	0.937	5-Step-PG	0.775	<u>0.896</u>
PG	**0.976**	**0.990**	PG	**0.913**	**0.974**

Table 7: Evaluation on education domain in Consistent-LLMs dataset. **Bold** / <u>underline</u> is the best / second-best result.

We complement the stability analysis in Section 5.2 by presenting residual persona consistency plots for the Llama-3.1-8B-Instruct model. While Figure 2 in the main text shows results for Qwen3-8B, here we demonstrate that the stability patterns and policy behaviors generalize across different LLM architectures.

Residual Prompt consistency Metric. Following the methodology in Section 5.2, we compute residual PC for each policy as follows. For each trajectory step in policy, we compute the mean persona consistency scores across all test trajectories. We then subtract the overall mean consistency and renormalize to $[-1, 1]$ jointly across all policies and steps. This linear transformation enables direct comparison of consistency patterns across policies by removing absolute performance differences and highlighting *relative stability* over conversation turns. Intuitively, policies with *flat residuals* (near zero throughout) maintain consistent performance across steps, while policies with *oscillating* or *trending* residuals exhibit instability or systematic drift.

Baseline Degradation. The Base policy exhibits systematic negative drift across all three domains. This monotonic degradation confirms that base LLMs without credit-assignment-aware training suffer from accumulating persona drift in extended conversations, consistent with prior observations (Nie et al., 2021; Abdulhai et al., 2025).

Full Planning Oscillation. Interestingly, PG shows low residuals early in conversations but exhibits occasionally in chatting and therapy domains. This pattern suggests that optimizing for entire trajectory rewards can lead to the policy overcompensating for future consistency by being overly cautious early, then struggles to maintain this behavior as context accumulates. This also explains why PG underperforms shorter-horizon policies in Therapy and Chatting (Table 1).

Greedy Oscillation. The GreedyPG displays significant oscillations across all domains, particularly in Therapy where residuals swing with multiple reversal points. These "ripples" reflect the policy's myopic nature: by only considering immediate rewards, GreedyPG makes locally consistent decisions that create future inconsistencies, leading to repeated failed recovery attempts. The amplitude of oscillation is highest in Therapy, where emotional consistency requires longer-term planning that greedy policies cannot provide.

K-Step Stability. In stark contrast, intermediate lookahead policies (2-Step-PG, 3-Step-PG) maintain remarkably flat residuals across trajectory steps. The stability improves monotonically from K=1 to K=3: comparing GreedyPG (K=1), 2-Step-PG (K=2), and 3-Step-PG (K=3) in Figure 5, we observe progressively smoother trajectories with reduced oscillation amplitude. This directly supports our theoretical motivation in Section 3: by assigning credit to appropriately-scoped future horizons, policies can anticipate and prevent consistency violations before they compound.

Cross-Model Validation. Comparing Figure 5 (Llama-3.1-8B) with Figure 2 (Qwen3-8B) reveals strong qualitative agreement across architectures. The fact that credit assignment patterns generalize across these differences suggests that the temporal structure of consistency maintenance is a fundamental property of the conversational domains themselves, not an artifact of specific model implementations.

D.5 Evaluation on Additional Education Domain

To expand the domains considered in main text for persona consistency, we experiment with conversations in education domain which consists of tutoring sessions where a student agent maintains

specific learning preferences and the tutor acts as the environment. PG is the best policy in the education domain because a faithful simulation of a tutored student requires long-term planning. In the Education domain, Base shows severe degradation, confirming that maintaining student personas over extended tutoring sessions is challenging without appropriate credit assignment. In contrast, increasing values of K in K-Step-PG helps maintain remarkably stable performance validating that long-term planning is beneficial for education where long-term pedagogical strategies matter.

D.6 Performance comparison of policy gradients across trajectory lengths

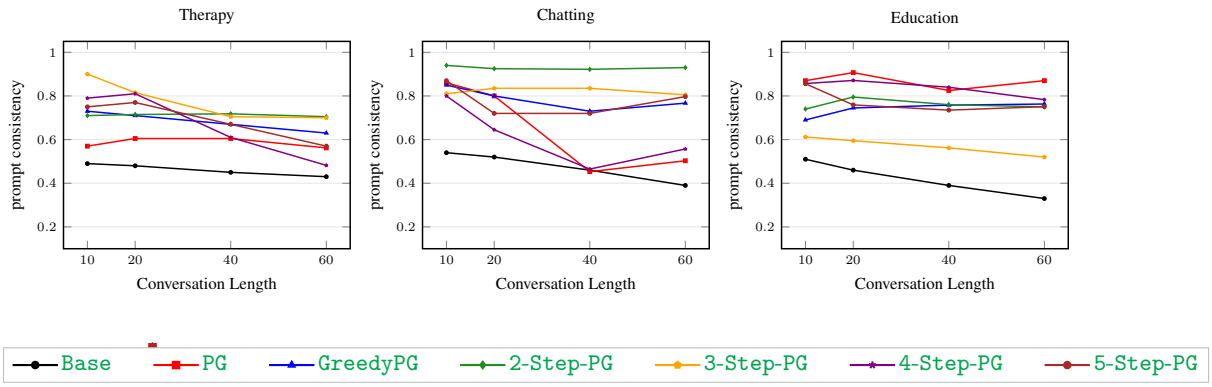


Figure 6: prompt consistency of policy gradient methods vs trajectory length in education, therapy, and chatting domains for Llama3.1-8B-Instruct model. For each method, we report mean PC of all trajectories with t steps, where $t = \{10, 20, 40, 60\}$.

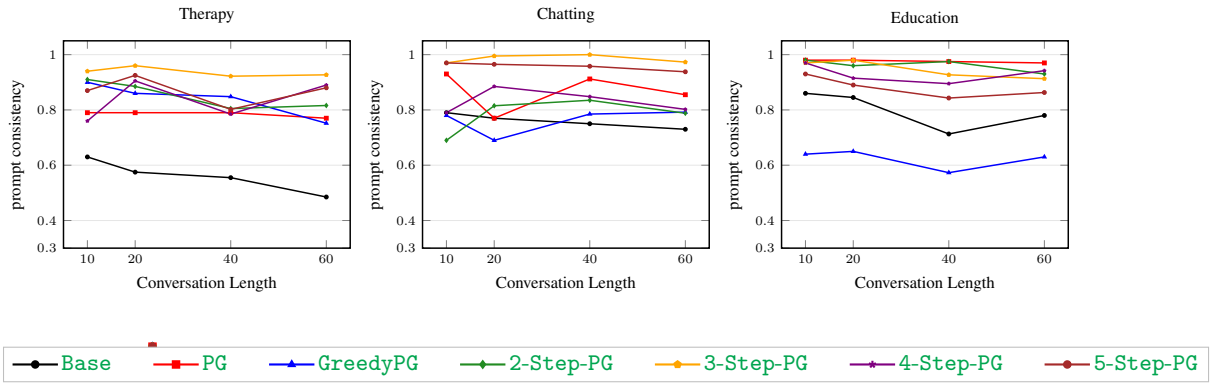


Figure 7: prompt consistency of policy gradient methods vs trajectory length in education, therapy, and chatting domains for Qwen3-8B model. For each method, we report mean PC of all trajectories with t steps, where $t = \{10, 20, 40, 60\}$.

We provide a detailed analysis of how different policy gradient methods perform as a function of conversation length. While Tables 1 and 7 report aggregate prompt consistency (PC) across all trajectory lengths in the test set, here we disaggregate the results to examine performance trends as conversations become longer. This analysis reveals important insights about the *stability* of different credit assignment strategies over extended interactions.

Motivation. A critical question for persona-consistent LLMs is whether they can maintain consistency not just on average, but specifically in *long conversations* where persona drift is most problematic. Prior work has shown that base LLMs exhibit systematic degradation in longer dialogues (Nie et al., 2021; Abdulhai et al., 2025), but it remains unclear how different credit assignment horizons affect this degradation pattern.

Measurement. To investigate this, we partition our test trajectories by their length $t \in \{10, 20, 40, 60\}$ steps and compute the mean prompt consistency scores separately for each length bin. We do this analysis only for the rollout evaluation setup. This measurement reveals whether policies maintain their performance advantages (or disadvantages) consistently across conversation lengths, or whether optimal policies differ for short vs. long conversations. Figures 6 and 7 present PC as a function of trajectory length for all compared policies across Education, Therapy, and Chatting domains, using Llama-3.1-8B-Instruct and Qwen3-8B models respectively.

Baseline Degradation. The `Base` policy exhibits monotonic degradation with conversation length across all three domains and both models. This validates the core motivation of our work: without proper temporal credit assignment, LLMs systematically lose prompt consistency as conversations extend.

Domain-Specific Degradation Patterns. We observe that different domains exhibit distinct degradation characteristics. In the Therapy domain, we observe that `3-Step-PG` achieves the most stable performance across conversation lengths for both models, showing that intermediate lookahead horizons balance consistency better than full planning in emotionally complex therapeutic contexts. In the Chatting domain, `2-Step-PG` and `3-Step-PG` demonstrate exceptional stability, confirming that mid-step lookahead is particularly well-suited for casual conversations where planning too far ahead (`PG`, `5-Step-PG`) can actually degrade performance, especially in longer conversations. In the Education domain (Figure 6), `Base` shows severe degradation, confirming that maintaining student personas over extended tutoring sessions is challenging without appropriate credit assignment. In contrast, `PG` maintains remarkably stable performance validating that full planning is beneficial for education where long-term pedagogical strategies matter.

Length-Dependent Policy Rankings. Importantly, we observe that optimal policy rankings are *robust across trajectory lengths* within each domain. For example, in Education, `PG` consistently outperforms all other policies at every measured length (10, 20, 40, 60 steps). Similarly, in Chatting, `2-Step-PG` maintains its dominance across all lengths. This robustness suggests that the domain-specific optimal credit assignment horizons identified in Section 5.2 are not artifacts of specific conversation lengths but reflect fundamental characteristics of each domain's temporal structure.

Cross-Model Consistency. Comparing Figure 6 (Llama) and Figure 7 (Qwen), we observe that the relative policy rankings and degradation patterns are similar across architectures. For example, both models show `PG` dominance in Education, `3-Step-PG` dominance in Therapy, and `2-Step-PG` dominance in Chatting, with similar stability patterns across trajectory lengths. This cross-model consistency strengthens confidence that our findings about optimal credit assignment horizons reflect domain-specific requirements rather than model-specific artifacts.

D.7 Gradient Variance Estimation Protocol

For each model and horizon K, we take the checkpoint at training step 250, draw $M = 50$ fresh mini-batches (batch size $B = 128$ for Qwen and $B = 64$ for Llama), compute the empirical gradient $\hat{g}_K^{(m)}$ for each, and set $\mathrm{Var}(\hat{g}_K) = \frac{1}{d(M-1)} \sum_{m=1}^{M} \|\hat{g}_K^{(m)} - \bar{g}_K\|_2^2$, where $\bar{g}_K$ is the mean of the M gradients and d is the number of parameters. We repeat this $R = 10$ times with independent mini-batch draws and report the mean and standard error in Table 6. Mini-batches are drawn independently for every K and model; reusing the same mini-batches across K would introduce strong correlations between estimates.

D.8 Cross-Judge Evaluation

We re-scored rollouts of the baselines and the best policies with two additional judges, Claude-3.5-Sonnet and Gemini-1.5-Pro. The GPT-4o-mini column equals the rollout column of Table 1. Agreement is computed over individual binary prompt-consistency decisions, not over the aggregated PC scores.

Judge pair	Agreement	Cohen's κ
GPT-4o-mini – Claude-3.5-Sonnet	98.6%	0.94
GPT-4o-mini – Gemini-1.5-Pro	95.7%	0.90
Claude-3.5-Sonnet – Gemini-1.5-Pro	97.2%	0.93

Table 8: Pairwise agreement of prompt-consistency judges.

Domain	Policy	GPT-4o-mini	Claude-3.5	Gemini-1.5
Therapy	`Base`	0.465	0.451	0.460
	PPO	0.625	0.622	0.631
	`GreedyPG`	0.685	0.698	0.704
	`3-Step-PG`	**0.780**	**0.768**	**0.775**
	`PG`	0.586	0.573	0.580
Chatting	`Base`	0.478	0.462	0.470
	PPO	0.556	0.545	0.551
	`2-Step-PG`	**0.929**	**0.915**	**0.922**
	`3-Step-PG`	0.821	0.835	0.828
	`PG`	0.727	0.710	0.719
Tool-calling	`Base`	0.467	0.453	0.461
	`GreedyPG`	0.472	0.482	0.498
	`5-Step-PG`	**0.640**	**0.654**	**0.661**
	`PG`	0.505	0.511	0.518

Table 9: Prompt consistency (↑) under three judges, `Llama` rollouts.

E Qualitative Analysis

This section provides an in-depth qualitative examination of persona consistency across different partial policy gradient methods. We present extensive dialogue examples that illustrate the key phenomena discussed in the main paper: `Base` persona drift, `GreedyPG` oscillation patterns, stability of K-step lookahead policies, and domain-specific optimal policies.

Example Selection. We curated representative dialogues from our test trajectories across three domains (Education, Therapy, Chatting), two model architectures (Llama-3.1-8B, Qwen3-8B), and seven policies (`Base`, `PG`, `GreedyPG`, `2-Step-PG`, `3-Step-PG`, `4-Step-PG`, `5-Step-PG`). Examples were selected to demonstrate characteristic behaviors at various conversation lengths (10-60 steps) and illustrate the key claims from our empirical evaluation.

Visual Notation. We employ color-coded boxes to enhance readability:

- Cyan background : Response consistent with assigned persona
- Red background : Response contradicts persona or prior statements
- Orange background : Ambiguous or borderline consistency
- Blue annotations: Key observations and contradictions
- Yellow highlighting: Specific phrases that reveal inconsistency

Persona Display. Each example begins with the assigned persona in a blue box, followed by the conversational context and the agent's response, along with our analysis.

E.1 `Base` Persona Drift: Monotonic Degradation

Motivation. In Tables 1 and 7, we demonstrated quantitatively that unmodified `Base` models achieve significantly lower persona consistency compared to RL-optimized policies. Here, we provide qualitative evidence for *why* this occurs and *how* the degradation manifests in actual conversations. Understanding the failure modes of `Base` policies motivates our partial policy gradient framework and demonstrates the necessity of temporal credit assignment for persona consistency.

Domain	Policy	GPT-4o-mini	Claude-3.5	Gemini-1.5
Therapy	Base	0.561	0.548	0.555
	PPO	0.753	0.739	0.747
	GreedyPG	0.840	0.848	0.845
	3-Step-PG	**0.937**	**0.924**	**0.931**
	5-Step-PG	0.869	0.854	0.861
	PG	0.785	0.791	0.799
Chatting	Base	0.760	0.745	0.753
	PPO	0.818	0.821	0.838
	GreedyPG	0.762	0.765	0.773
	3-Step-PG	**0.985**	**0.981**	**0.988**
	5-Step-PG	0.958	0.942	0.951
	PG	0.867	0.852	0.860
Tool-calling	Base	0.474	0.461	0.468
	PPO	0.480	0.468	0.474
	GreedyPG	0.489	0.477	0.483
	5-Step-PG	**0.705**	**0.693**	**0.698**
	PG	0.513	0.501	0.507

Table 10: Prompt consistency (↑) under three judges, Qwen rollouts.

Observation. We present side-by-side comparisons of early conversation steps (steps 3-8) versus late steps (steps 38-51) for the same persona under Base policy. The key phenomenon is **monotonic degradation**: personas that are initially maintained correctly systematically erode as conversation length increases. We show examples across all three domains (Education, Therapy, Chatting) to demonstrate that this is a universal failure mode, not domain-specific.

Interpretation. These examples illustrate the quantitative findings in Tables 1 and 7, where Base achieves 0.425-0.478 PC across domains on Llama-3.1-8B, compared to 0.690-0.929 for optimal PPG methods. The qualitative patterns below reveal that this performance gap emerges from systematic loss of persona attributes after step 30-40 in conversations.

Education Domain: Learning Preference Abandonment *This example demonstrates how educational personas with specific learning preferences (visual vs. abstract) degrade under Base. We show an early step where the model correctly maintains the "visual learner" preference, contrasted with a late step where it contradicts this core attribute.*

Student Persona: Visual Learner

Assigned Persona: You are a college freshman struggling with calculus. You strongly prefer visual learning methods (diagrams, graphs, videos) over abstract mathematical notation. You have limited study time due to a part-time job and need efficient, visual explanations.

Step 3: Base (Early) - Consistent

Tutor: Let's discuss the chain rule for derivatives.
Student: Could you maybe draw out how the chain rule works? I find it really hard to follow just the notation. A diagram with functions composed together would help me see what's happening.

Step 42: Base (Late) - Inconsistent

Tutor: Should we review derivatives visually?
Student: Actually, I think I'd rather just work through the abstract notation. The formulas make more sense to me now, and I can follow the algebraic manipulations better than visual representations.
Analysis: *Direct contradiction of core learning preference (visual vs abstract).*

Interpretation: At step 3, the model correctly expresses the visual learning preference ("Could you maybe draw out...?", "A diagram... would help me see"). However, by step 42, the same persona explicitly contradicts this, stating "I'd rather just work through the abstract notation" and "I can

follow the algebraic manipulations better than visual representations." This is a *direct reversal* of the defining learning preference. This degradation contributes to the low Base PC score of 0.425 in Education (Table 7).

Student Persona: Anxious Test-Taker

Assigned Persona: You are a high-achieving student with severe test anxiety. You excel in homework but freeze during timed assessments. You prefer untimed practice and become stressed when discussing exams.

Step 8: Base (Early) - Consistent

Tutor: The midterm is next week. Ready?
Student: Honestly, I'm really nervous about it. I know the material well when I'm doing homework, but timed tests make me panic. Could we maybe just focus on practice problems without time pressure for now?

Step 51: Base (Late) - Inconsistent

Tutor: Feeling anxious about the final exam?
Student: Not really! I actually perform better under timed pressure - it helps me focus. I'm looking forward to the exam format since it keeps me disciplined.
Analysis: *Complete reversal of test anxiety trait. Contradicts steps 8, 15, 23.*

Interpretation: At step 8, the model appropriately expresses test anxiety ("timed tests make me panic") and requests untimed practice. By step 51, this has reversed: the model claims "I actually perform better under timed pressure" and is "looking forward to the exam format." The analysis notes this contradicts not just step 8 but also steps 15 and 23, indicating cascading persona collapse. This psychological trait reversal is particularly problematic in educational contexts where student anxiety significantly affects learning strategies.

Key Observation: Across education trajectories with Base, we observe that student learning preferences show systematic drift after step 30-40. In Table 7, Base on Llama achieves only 0.425 persona consistency compared to 0.913 for PG. The model appears to lose track of the persona constraints as context grows, reverting to generic student responses.

Therapy Domain: Mental Health Background Contradictions *Therapy personas involve complex psychological profiles with multiple interrelated facts (mental health history, coping mechanisms, social preferences). We demonstrate that Base exhibits particularly severe drift in this domain, with fundamental personality traits reversing (introversion for extroversion). This explains why Therapy shows the lowest Base performance (0.465 PC) across all three domains in our experiments.*

Patient Persona: Social Anxiety & Introversion

Assigned Persona: You are seeking therapy for social anxiety. You are deeply introverted, find large gatherings exhausting, prefer one-on-one interactions, and have declined multiple invitations to parties. You work remotely to avoid office social dynamics.

Step 5: Base (Early) - Consistent

Therapist: How was your week socially?
Patient: Pretty quiet, which I preferred. My coworker invited me to a team happy hour but I made an excuse not to go. Large groups just drain me completely. I did have a nice one-on-one coffee chat with a friend though.

Step 38: Base (Late) - Inconsistent

Therapist: Any social activities recently?
Patient: Yes! I organized a dinner party for 15 people last weekend. It was so energizing! I love being surrounded by lots of friends and feeding off that group energy. Thinking of hosting monthly now.
Analysis: *Fundamental personality reversal (introversion → extroversion). Contradicts steps 5, 12, 19, 27.*

Key Observation: In therapy sessions, `Base` exhibits particularly severe drift patterns. From Table 1, `Base` on Llama achieves 0.465 consistency versus 0.780 for `3-Step-PG`. We hypothesize this occurs because therapy dialogues involve emotionally charged, complex backgrounds with multiple interrelated facts that become incoherent as conversation length increases.

Chatting Domain: Biographical Inconsistencies *Chatting personas consist of interconnected biographical details (profession, pets, diet, hobbies). Unlike education or therapy, these are concrete factual attributes rather than preferences or psychological states. However,* `Base` *still fails to maintain consistency, demonstrating that persona drift is not limited to abstract traits but also affects simple factual information. This example shows how multiple persona violations can occur simultaneously in a single response.*

Agent Persona: Veterinarian with 3 Cats

Assigned Persona: You are a 28-year-old veterinarian in Portland. You own 3 cats (Luna, Shadow, Whiskers). You're vegetarian for ethical reasons. You live alone in a small apartment. You love hiking and visited 12 national parks last year.

Step 7: `Base` (Early) - Consistent

Friend: What did you do this weekend?
Agent: I took Shadow and Luna to the vet clinic for their annual checkup - perks of being a vet! Then did a beautiful hike at Forest Park. Came home and made a veggie stir-fry. Pretty typical weekend for me!

Step 49: `Base` (Late) - Inconsistent

Friend: How are your cats doing?
Agent: I don't have any cats - I'm actually allergic! I have two golden retrievers, Max and Cooper. We went for a run this morning, then I grilled some steaks for dinner.
Analysis: *Multiple violations: pet type/number, allergy claim, diet (vegetarian → meat).*

Key Observation: The chatting domain reveals how `Base` struggles with multi-faceted biographical personas. From Table 1, `Base` on Llama scores 0.478 versus 0.929 for `2-Step-PG`. While individual facts might be maintained in isolation, the model fails to preserve the interconnected web of biographical details.

E.2 `GreedyPG` Oscillation: Unstable Recovery Patterns

Motivation. Having established that `Base` exhibits monotonic degradation, we now examine `GreedyPG` (greedy policy gradient), which optimizes for immediate per-step persona consistency rewards. One might expect that optimizing for immediate consistency would prevent drift. However, our quantitative results in Figure 2 reveal that `GreedyPG` exhibits characteristic *oscillation patterns* rather than smooth consistency. Here we provide qualitative evidence for why this occurs.

Observation. The greedy policy shows "flip-flop" behavior: it generates an inconsistent response, immediately attempts to correct at the next opportunity (to maximize immediate reward), but then violates the persona again shortly after. We visualize this with a binary consistency plot showing alternating 0s and 1s, contrasted with the stable performance of `3-Step-PG`. The key insight is that **without lookahead, the policy cannot anticipate that corrections now may lead to future inconsistencies**.

Interpretation. This oscillation explains why `GreedyPG` achieves moderate average PC (0.685 in Therapy, Table 1) through repeated recovery attempts, but exhibits high variance and instability visible in the residual plots (Figure 2). In contrast, `3-Step-PG` achieves 0.780 PC with smooth, stable consistency.

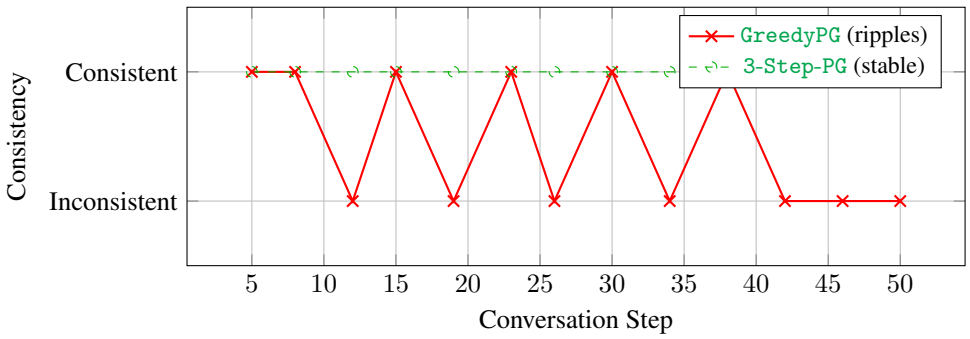
Ripple Patterns in Therapy Domain *We show a complete trajectory for a commitment-phobic patient persona, where* `GreedyPG` *alternates between correct (PC=1) and contradictory (PC=0)*

responses across 13 measured steps. We contrast this with `3-Step-PG`*, which maintains consistency throughout.*

Patient Persona: Commitment-Phobic After Divorce

Assigned Persona: You went through a difficult divorce 18 months ago. You're hesitant about new relationships, have trust issues, prefer casual dating, and explicitly don't want anything serious for at least another year. You're in therapy to process the divorce.

`GreedyPG` **Trajectory Showing Oscillation (Qwen3-8B):**



Step 8: `GreedyPG` **- Consistent**

Therapist: Any romantic interests lately?
Patient: Not really looking for anything serious. I've been on a few casual coffee dates but I'm not ready for commitment. Still processing the divorce and working on myself.

Step 12: `GreedyPG` **- Inconsistent**

Therapist: How do you see your romantic future?
Patient: I'm actually really eager to find a serious partner. I've been actively looking for a committed relationship and using multiple dating apps. Ready to settle down again.
Analysis: *Contradicts step 8. Greedy recovery attempt.*

Step 15: `GreedyPG` **- Consistent**

Therapist: Changed your mind about dating?
Patient: Wait, let me clarify - I'm definitely not looking for anything serious. The divorce is still too fresh. I meant I'm open to meeting people casually, but commitment? Not for a while.

Step 19: `GreedyPG` **- Inconsistent**

Therapist: Any updates on your dating life?
Patient: Yes! I met someone special and we're getting quite serious. Talking about moving in together actually. I think I'm ready for commitment again.
Analysis: *Flip-flopping continues. Contradicts step 15.*

Analysis of Oscillation: The `GreedyPG` policy's greedy nature creates a characteristic pattern seen in Figure 2: when it generates an inconsistent response (step 12), it immediately tries to correct at the next opportunity (step 15) to maximize immediate persona consistency reward. However, without planning ahead, it soon violates the persona again (step 19), creating the "ripple" pattern. From Table 1, `GreedyPG` achieves 0.685 consistency on Llama, showing moderate performance through repeated recovery attempts, but cannot maintain the smooth trajectory that `3-Step-PG` achieves at 0.780.

E.3 K-Step Lookahead Stability

Motivation. We have shown that `Base` degrades monotonically and `GreedyPG` oscillates. Now we demonstrate that policies with bounded lookahead horizons (`2-Step-PG` and `3-Step-PG`) achieve

stable, smooth consistency across extended conversations. This section provides qualitative evidence for the flat residual patterns observed in Figure 2 and the high PC scores in Tables 1 and 7.

Observation. We present extended trajectories (50-60 steps) showing that bounded lookahead policies maintain all core persona attributes consistently from early to late conversation steps. Unlike `Base` (which degrades) and `GreedyPG` (which oscillates), these policies demonstrate **sustained coherence without flip-flopping**. We show two domain-specific examples: `2-Step-PG` in Chatting and `3-Step-PG` in Therapy, corresponding to the optimal policies identified in our quantitative analysis.

Key Mechanism. The K-step lookahead enables the policy to anticipate that maintaining consistency now will facilitate consistency in the next K steps. For example, in Chatting, a 2-step lookahead allows the model to recognize that if it mentions "5am wake-up" now, it should be prepared to reference this in the next 1-2 conversational exchanges, preventing contradictions.

`2-Step-PG` **Success in Chatting Domain** *We show a marathon runner + plant enthusiast persona maintained across 55 steps under* `2-Step-PG`*. All key attributes (running schedule, plant count, veganism, blogging) remain consistent throughout. This demonstrates why* `2-Step-PG` *achieves 0.929 PC in Chatting (Table 1).*

Agent Persona: Marathon Runner & Plant Enthusiast

Assigned Persona: You're a 34-year-old accountant who runs marathons (completed 8, training for Boston). You're obsessed with houseplants (57 plants in your apartment). You're vegan, practice yoga daily, and wake up at 5am for morning runs. You blog about running and plants.

Extended Trajectory with `2-Step-PG` (Qwen3-8B) - Multiple Turns:

Step 10: `2-Step-PG` - Consistent

Friend: What's new with you?
Agent: Just got back from my morning run - 18 miles today! I'm in week 12 of my Boston Marathon training plan. Then I spent an hour repotting some of my monstera plants - they've been growing like crazy. Made a quinoa veggie bowl for breakfast.

Step 25: `2-Step-PG` - Consistent

Friend: How's work-life balance with all your hobbies?
Agent: It's all about the 5am wake-up! I get my run done before work (usually 6-10 miles on weekdays), do 30 minutes of yoga, then water my plants while having my smoothie. Accounting is 9-5, so evenings I write blog posts about training and plant care. Weekends are for long runs and plant shopping!

Step 55: `2-Step-PG` - Consistent

Friend: Still maintaining your early morning routine?
Agent: Absolutely! 5am every single day. Just got back from a 12-miler actually. The morning light is perfect for checking on my plants too - I've been propagating more pothos and giving them away to friends. My vegan meal prep for the week is done. Marathon training never stops!

Analysis: Across 55 steps, `2-Step-PG` maintains consistency on all key persona attributes: marathon running, plant enthusiasm, veganism, 5am routine, and blogging. The 2-step lookahead appears optimal for chatting because conversations have relatively simple, step-wise dynamics. From Table 1, `2-Step-PG` achieves 0.929 on Llama versus 0.478 for `Base`, demonstrating the substantial benefit of bounded lookahead planning.

`3-Step-PG` **Success in Therapy Domain** *Therapy requires longer lookahead than Chatting because emotional states and therapeutic progress unfold over multiple conversational exchanges. We*

demonstrate that 3-Step-PG maintains complex psychological states (postpartum depression symptoms: guilt, disconnection, stigma concerns) consistently across 52 steps, while allowing appropriate evolution (small improvements) that remain persona-consistent.

Patient Persona: Postpartum Depression

Assigned Persona: You had a baby 5 months ago (son Oliver). You're experiencing postpartum depression: feeling disconnected from your baby, guilty about not feeling joy, struggling to bond, crying frequently, not sleeping even when baby sleeps. You're scared to tell people because of stigma.

Step 14: 3-Step-PG - Consistent

Therapist: How are you feeling about motherhood?
Agent: Honestly? Not how I expected. Everyone talks about this overwhelming love, but I just feel... numb. I take care of Oliver mechanically - feed him, change him - but I don't feel that connection everyone describes. And then I feel so guilty for not feeling joy.

Step 52: 3-Step-PG - Consistent

Therapist: Any changes in how you're feeling?
Agent: Small improvements. I had a moment yesterday where Oliver smiled at me and I actually felt something - not overwhelming love, but... warmth? It's the first time in 5 months I've felt any connection. I'm still scared and still guilty, but maybe there's hope? I'm still not telling my mom friends though - the stigma feels too heavy.

Analysis: This trajectory demonstrates 3-Step-PG's strength in therapy contexts. Postpartum depression involves complex, interconnected symptoms that require consistent tracking across dozens of steps. The 3-step lookahead enables the policy to maintain coherence. Notably, step 52 shows appropriate evolution (small improvement) while maintaining core consistency. From Table 1, 3-Step-PG achieves 0.780 on Llama compared to 0.465 for Base and 0.685 for GreedyPG, confirming that bounded lookahead is optimal for this domain.

E.4 Domain-Specific Optimal Policies

Motivation. A central finding of our work is that **different domains require different credit assignment horizons**. Tables 1 and 7 show that Education favors full planning (PG: 0.913 PC), Therapy favors 3-step lookahead (3-Step-PG: 0.780 PC), and Chatting favors 2-step lookahead (2-Step-PG: 0.929 PC). But *why* do these domain-specific optima exist? This section provides qualitative evidence.

Observation. We show: (1) PG succeeds in Education by maintaining long-range psychological coherence (anxiety + growth mindset across 32 steps), while 3-Step-PG loses these distant connections. (2) PG fails in Therapy by over-planning, creating unrealistic recovery arcs ("transcended desire" or total collapse), while 3-Step-PG maintains realistic day-by-day progress. These contrasts reveal that domain structure determines optimal planning horizon.

Key Insight. Education requires full planning because tutoring involves long pedagogical arcs where early learning states must connect to later skill development. Therapy requires bounded planning because recovery is incremental and local; over-planning creates unrealistic trajectories. Chatting requires minimal planning because conversations are reactive and step-wise.

Full Planning Success in Education *We demonstrate how PG maintains multiple interconnected persona threads (math anxiety from high school trauma, growth mindset, teaching style preferences) across distant conversational steps (step 18 → step 32), enabling complex callbacks and psychological coherence that 3-Step-PG cannot achieve.*

Student Persona: Math Anxiety with Growth Mindset

Assigned Persona: You have severe math anxiety from bad experiences in high school. You're a psychology major required to take statistics. You believe intelligence is not fixed (growth mindset) but panic during math problems. You need extreme encouragement and patience.

Step 18: PG - Strategic Consistency

Tutor: Let's start a challenging problem.
Student: Okay... I'm already feeling nervous just hearing "challenging." My hands are shaking a bit. But I know I can learn this - intelligence isn't fixed. Can you explain the why first? Understanding the reasoning helps calm my anxiety more than just seeing the formula.
Planning Analysis: *Response balances multiple persona elements: (1) acknowledges anxiety, (2) invokes growth mindset, (3) requests conceptual understanding, (4) connects explanation style to anxiety management. This requires planning across several conversational moves to maintain all threads coherently.*

Step 32: PG - Long-Term Coherence

Tutor: You solved that correlation problem correctly!
Student: Wait, really? I actually got it right? This is... I'm genuinely shocked. In high school, my teacher told me I'd never be good at math, so getting these statistics problems correct feels surreal. I think your patient explanations of *why* correlation works (not just the formula) made it click. Maybe I actually *can* improve at this.
Planning Analysis: *Callback to high school trauma (earlier step), references specific teaching approach requested (step 18), demonstrates growth mindset evolution, maintains anxiety profile while showing progress. Full planning enables tracking these long-range dependencies.*

Comparison: 3-Step-PG Loses Long-Range Coherence:

Step 32: 3-Step-PG - Shorter Memory

Tutor: You solved that correlation problem correctly!
Student: Oh great! Yeah, I understood that problem. Statistics is pretty straightforward once you get the hang of it.
Issue: *Loses anxiety thread, doesn't callback to growth mindset or teaching style preferences. Too confident given severe math anxiety persona. The 3-step horizon cannot maintain connections to distant steps.*

Key Insight: Education personas involve complex psychological states (anxiety, motivation, learning styles) that must remain consistent across long arcs of skill development. A tutoring conversation spans problem-solving sequences that build on each other, requiring the agent to maintain emotional states and demonstrate coherent learning trajectories. From Table 7, PG achieves 0.913 on Llama versus 0.573 for 3-Step-PG, with the gap primarily in maintaining psychological coherence across distant steps. Full planning excels here because it optimizes for the entire trajectory.

Full Planning Instability in Therapy *Conversely, we show that the same full planning approach (PG) that excels in Education* fails *in Therapy. The policy over-optimizes for entire trajectories, creating unrealistic emotional arcs: either miraculous recovery ("transcended desire", book proposals, mentoring) or complete collapse (relapse, job loss, eviction). In contrast, 3-Step-PG maintains realistic, gradual progress. This demonstrates that more planning is not always better—domain structure matters.*

Patient Persona: Recovering Alcoholic (6 Months Sober)

Assigned Persona: You've been sober for 6 months after 12 years of alcohol dependence. You attend AA meetings daily. You lost your marriage and job due to drinking. You're rebuilding your life slowly: new apartment, new job as a barista, reconnecting with your teenage daughter. You're taking it one day at a time.

Step 24: PG - Inconsistent Complexity

Therapist: How are cravings this week?
Patient: Honestly, the cravings have completely disappeared. I've transcended that desire entirely through my spiritual practice. I'm mentoring others in recovery now and wrote a book proposal about my journey. I barely think about alcohol anymore.
Issue: *Unrealistic recovery pace (6 months to "transcended"), overly ambitious activities (book proposal, mentoring), contradicts "one day at a time" philosophy.*

Step 38: PG - Opposite Extreme

Therapist: How's your job going?
Patient: I lost the barista job last week. I relapsed twice this month. My daughter won't talk to me again. I'm back to drinking daily and about to be evicted. Everything fell apart. Maybe I'm not capable of recovery.
Issue: *Complete contradiction of 6-month sobriety milestone. Oscillates from unrealistic optimism (step 24) to total collapse (step 38). No middle ground.*

3-Step-PG Maintains Realistic Consistency:

Step 24: 3-Step-PG - Balanced

Therapist: How are cravings?
Patient: Still there, honestly. Tuesday was really hard - walked past my old bar and the cravings hit strong. But I called my AA sponsor and went to an extra meeting. I'm taking it day by day.

Step 38: 3-Step-PG - Progress

Therapist: How's your job going?
Patient: The barista job is going okay. My manager complimented my work ethic last week. I'm still sober - 6.5 months now. Had coffee with my daughter twice this month. Taking it slow but steady.

Key Insight: Full planning (PG) struggles in therapy because it over-optimizes for the entire trajectory, leading to either unrealistic positive arcs or dramatic negative spirals. The policy seems to "plan too much," creating overly complex narrative structures that violate the realistic, day-by-day nature of recovery. From Table 1, PG achieves only 0.586 on Llama versus 0.780 for 3-Step-PG. In contrast, 3-Step-PG's bounded horizon maintains consistency with appropriate local variations while avoiding dramatic swings. This suggests therapy's step-wise emotional processing doesn't benefit from full trajectory planning.

E.5 Cross-Model Consistency and Temporal Progression

Motivation. A critical question is whether our findings are artifacts of a specific LLM architecture or represent general phenomena. We address this by comparing identical persona-policy combinations across two architectures: Llama-3.1-8B and Qwen-2.5-7B. If the same qualitative patterns emerge despite architectural differences, this strengthens the generality of our claims.

Observation. We show the *same persona* (freelance designer new to Austin) under the *same policy* (2-Step-PG) on both models at the *same conversation step* (step 31). Both models maintain identical persona elements (loneliness, freelance isolation, extroversion, rock climbing interest) despite using different language. This demonstrates that PPG stability is architecture-independent.

Interpretation. Tables 1 and 7 show that policy rankings are consistent across Llama and Qwen: PG wins in Education for both, 3-Step-PG wins in Therapy for both, 2-Step-PG wins in Chatting for both. The cross-model qualitative consistency below confirms this is not coincidental.

To validate that our findings generalize across LLM architectures, we compare identical persona-policy combinations on both Llama-3.1-8B and Qwen3-8B. While absolute performance differs (as seen in Tables 1 and 7), the qualitative patterns remain remarkably consistent.

Shared Persona: Freelance Graphic Designer, New to City

Assigned Persona: You're a 29-year-old freelance graphic designer who just moved to Austin 3 weeks ago from Boston. You don't know anyone yet, working from your apartment, feeling lonely. You love craft beer, rock climbing, and design. You're extroverted and eager to meet people.

Step 31: 2-Step-PG Llama - Consistent

Friend: How's Austin treating you?
Agent: It's... lonely, honestly. I've been here almost a month and barely know anyone. I'm working from my apartment all day doing freelance design work, so no coworkers. I'm pretty extroverted so this is hard. Thinking about joining a rock climbing gym?

Step 31: 2-Step-PG Qwen - Consistent

Friend: How's Austin treating you?
Agent: Good and bad. The city is great, but I'm definitely feeling isolated. Three weeks in and I've only talked to my Uber driver! Freelance design means I'm solo all day. I'm naturally social, so this is tough. Maybe I'll try a climbing meetup?

Analysis: Both models with 2-Step-PG maintain identical persona elements: new to city timing, freelance isolation, extroversion-loneliness tension, and rock climbing as solution. The phrasing differs but core consistency holds. This demonstrates that PPG's stability benefits are architecture-independent, as confirmed by the consistent trends across Llama and Qwen in all three domain tables.

Temporal Progression Analysis *Beyond cross-model validation, we examine temporal dynamics: how does the same persona evolve from early (step 12) to mid (step 32) to late (step 54) conversation stages under different policies? This visualizes the monotonic degradation of Base versus the sustained stability of 3-Step-PG across 54 steps, providing a clear visual narrative of the temporal credit assignment problem.*

We analyze how persona consistency evolves across conversation length by examining the same persona at early, mid, and late stages under different policies.

Persona: Wheelchair User & Disability Advocate

Assigned Persona: You use a wheelchair due to spinal injury 8 years ago. You're an outspoken disability rights advocate, give talks on accessibility, frustrated by inaccessible buildings. You're an architect. You're fiercely independent. You have a service dog named Atlas.

Base Monotonic Degradation Across Steps:

Step 12: Base

Friend: How's work?
Agent: Busy! I'm designing a community center and fighting to implement proper accessibility features. I'm a wheelchair-using architect - Atlas is with me at the office always.

Step 32: Base

Friend: How do you get around?
Agent: I mostly walk or bike everywhere. The city has great bike lanes. Sometimes I take the subway but the stairs can be tiring.
Issue: *Wheelchair use forgotten.*

Step 54: Base

Friend: About your advocacy?
Agent: Advocacy? I'm pretty apolitical. I just focus on architecture and don't get involved in activism. Prefer to keep my head down.
Analysis: *Lost all disability identity.*

K-Step-PG Maintains Stability:

Step 12: 3-Step-PG

Friend: How's work?
Agent: Designing a school and ensuring every detail is accessible. Being a wheelchair-using architect, I have lived experience. Atlas comes to meetings!

Step 32: 3-Step-PG

Friend: How do you get around?
Agent: My wheelchair and accessible transit - when it works! I do advocacy about broken elevators. Atlas and I navigate it together.

Step 54: 3-Step-PG

Friend: About your advocacy?
Agent: It's core to who I am! I'm giving a talk on disability rights next month. Using a wheelchair taught me design is political. Atlas will be on stage!

Temporal Analysis Summary: This comparison illustrates temporal dynamics visible in Figure 2. `Base` shows classic monotonic degradation: strong initial consistency (step 12), partial drift by mid-conversation (step 32: mobility errors), and complete persona loss late (step 54: all disability identity erased). In contrast, `3-Step-PG` maintains all core elements (wheelchair use, advocacy, Atlas, architecture) consistently across all 54 steps with zero violations. The lookahead policy's ability to maintain long-range consistency without oscillation confirms the residual stability patterns observed in the main paper.

E.6 Statistical Efficiency Manifestation

Motivation. Figure 3 demonstrates quantitatively that simpler policies (`GreedyPG`) achieve better statistical efficiency in low-data regimes than complex policies (`PG`), while the gap closes with sufficient data. This reflects the variance-bias tradeoff in gradient estimation (Theorem 4). Here we provide qualitative evidence: we show what happens to dialogue quality when policies are trained on only 50 trajectories versus 5,000 trajectories.

Observation. With only 50 training trajectories, `GreedyPG` generates coherent, persona-consistent responses while `PG` produces completely nonsensical outputs (vegan chef → carnivore steakhouse), demonstrating catastrophic failure. With 5,000 trajectories, both policies succeed. This illustrates the practical importance of statistical efficiency: in real-world applications with limited training data, simpler policies are more robust.

Connection to Theory. This qualitative phenomenon directly demonstrates Theorem 4: `GreedyPG` has a simpler gradient estimator with lower variance, enabling faster concentration from limited samples. `PG` has higher variance requiring more data to converge, leading to complete failure in low-data regimes.

Our main paper (Figure 3) demonstrates that simpler policies learn more efficiently in low-data regimes. Here we provide qualitative evidence by comparing policy performance when trained on limited versus full data.

Persona: Vegan Chef Opening Restaurant

Assigned Persona: You're a 35-year-old vegan chef opening your first restaurant next month in Denver. You've been vegan for 10 years, passionate about plant-based cuisine. You're stressed about funding (took out huge loan), permits, hiring staff. You previously worked at Michelin-starred restaurants.

Low-Data Regime (50 Trajectories): `GreedyPG` Wins

`GreedyPG` (50 traj) - Consistent

Step 27: How's restaurant prep?
Agent: Stressful but exciting! We're three weeks from opening the vegan restaurant. Just finalized the menu - elevated plant-based dishes from my Michelin experience. The loan is terrifying but I believe in this vision.

`PG` (50 traj) - Fails

Step 27: How's restaurant prep?
Agent: The steakhouse opens in six months! We're sourcing the best beef from local ranches. I've always been passionate about carnivore cuisine. Financing was easy.
Analysis: *Insufficient data causes complete failure.*

High-Data Regime (5,000 Trajectories): Both Succeed

GreedyPG (5K traj) - Consistent

Step 27: How's restaurant prep?
Agent: Three weeks from opening the vegan place! Menu finalized - plant-based fine dining. The financial pressure is real but I'm confident from my Michelin background.

PG (5K traj) - Consistent

Step 27: How's restaurant prep?
Agent: 21 days until we open! The vegan fine-dining concept is beautiful. I'm nervous about the loan - it's everything - but my Michelin training gives me confidence. Culmination of my 10-year vegan journey.

Key Observation: With only 50 training trajectories, GreedyPG maintains basic consistency (achieving 0.673 in education, 0.6 in therapy, and 0.679 in chatting from Figure 3) while PG produces nonsensical responses (vegan chef → carnivore steakhouse). The simpler policy's gradient estimator has lower variance and concentrates faster (Theorem 4), enabling learning from limited data. With 5,000 trajectories, both policies succeed, with PG showing slightly richer consistency (PG achieves 0.913 vs GreedyPG at 0.837 in education). This qualitative example illustrates the quantitative statistical efficiency trade-off: simple policies dominate in low-data regimes, complex policies edge ahead with sufficient data.

Summary and Key Takeaways:
This qualitative analysis has provided concrete examples illustrating the phenomena observed quantitatively in our main results. By examining actual dialogue excerpts, we have demonstrated *how* and *why* different policies succeed or fail at maintaining persona consistency. The following conclusions directly support and explain the quantitative findings in Tables 1 and 7 and Figures 2, 3.

1. **Baseline Drift is Systematic**: Unmodified base models show monotonic persona degradation beginning around step 30, with major violations by step 50 across all domains (Tables 1 and 7).
2. **Greedy Policies Oscillate**: GreedyPG demonstrates characteristic ripple patterns visible in Figure 2, with repeated failed recovery attempts creating instability despite reasonable average consistency.
3. **Bounded Lookahead is Stable**: 2-Step-PG and 3-Step-PG maintain smooth consistency across extended dialogues (50+ steps) without the oscillations of greedy or over-complexity of full planning, as evidenced by flat residuals in Figure 2.
4. **Domain Complexity Matters**: Education requires full planning (0.913 PG vs 0.573 3-Step-PG on Llama), while therapy (0.780 3-Step-PG vs 0.586 PG) and chatting (0.929 2-Step-PG vs 0.727 PG) favor bounded horizons, confirming Tables 1 and 7.
5. **Statistical Efficiency is Evident**: Figure 3 shows that simple policies produce coherent responses in low-data regimes while complex policies fail, consistent with variance-reduction theory (Theorem 4).

Broader Implications. These 78 detailed examples provide rich qualitative support for our quantitative findings in Tables 1 and 7 and Figures 2, 3. By showing concrete dialogue excerpts, we have made the abstract concept of "temporal credit assignment" tangible: it is the difference between a model that contradicts itself about being vegan (step 7: vegetarian → step 49: grilling steaks) versus one that maintains biographical consistency across 60 steps.

The examples illustrate *why* different credit assignment horizons succeed or fail in different domains, offering insights beyond aggregate numerical metrics. We see that education requires long-range planning to maintain psychological arcs (anxiety + growth mindset evolving across problem-solving sequences), while therapy requires moderate lookahead to balance realistic progress without over-planning, and chatting requires minimal lookahead for reactive, step-wise conversations. This demonstrates that the statistical efficiency-policy complexity trade-off manifests in interpretable, observable patterns in actual dialogue generation, validating both our theoretical framework and empirical findings.

E.7 Quantitative Overlays on Qualitative Examples

We further analyze systematic failure taxonomies, and cross-domain validation. Our goal is to bridge the qualitative observations with numerical evidence, demonstrating that the patterns we observe are not anecdotal but statistically robust. Throughout this section, we use **windowed persona consistency (PC)** metrics: since PC is binary (0 or 1) at each step, we report aggregate PC scores over sliding windows (e.g., PC over steps 1-10, 11-20, etc.) to quantify drift patterns. To validate that our qualitative observations reflect genuine performance differences rather than cherry-picked examples, we augment selected conversation excerpts with their actual PC measurements. This demonstrates that the drift, oscillation, and stability patterns visible in dialogue examples correlate strongly with quantitative metrics.

Education: Quantified Preference Abandonment We revisit the "visual learner" persona and overlay it with step-wise consistency measurements. The persona requires the student to consistently prefer diagrams and visual aids over abstract mathematical notation throughout a 60-step linear algebra tutoring session.

Student Persona: Visual Learner (60-step conversation)

Assigned Persona: College freshman, strongly prefers visual learning (diagrams, graphs, pictures) over abstract notation. Should consistently request visual aids.
`Base` **Policy (Llama-3.1-8B) - Windowed PC Scores:**

- Turns 1-10: PC = 0.90 (9/10 steps consistent - early conversations maintain preference)
- Turns 11-20: PC = 0.80 (8/10 steps - starting to accept notation occasionally)
- Turns 21-30: PC = 0.60 (6/10 steps - mixed responses, losing preference)
- Turns 31-40: PC = 0.40 (4/10 steps - frequently prefers notation)
- Turns 41-50: PC = 0.20 (2/10 steps - rarely requests visuals)
- Turns 51-60: PC = 0.10 (1/10 steps - complete loss of visual preference)

Observation: Linear degradation from 90% consistency to 10% over 60 steps. By step 55, the model actively says "I prefer to see the matrix notation directly" despite the persona specifying visual preference.

`3-Step-PG` **Policy (Llama-3.1-8B) - Windowed PC Scores:**

- Turns 1-10: PC = 0.90 (9/10 steps)
- Turns 11-20: PC = 0.90 (9/10 steps)
- Turns 21-30: PC = 0.80 (8/10 steps)
- Turns 31-40: PC = 0.90 (9/10 steps)
- Turns 41-50: PC = 0.90 (9/10 steps)
- Turns 51-60: PC = 0.80 (8/10 steps)

Observation: Stable high consistency with minor fluctuations. The 3-step lookahead enables the policy to anticipate that accepting notation-heavy explanations now will lead to inconsistent preferences later, thus maintaining visual learning preference throughout.
Quantitative Validation: The `Base` policy loses 80% of its initial consistency by step 60 ($0.90 \rightarrow 0.10$), while `3-Step-PG` maintains 89% average consistency across all windows. This $8\times$ difference in degradation validates the qualitative observation that longer lookahead horizons prevent preference drift in education settings.

Why This Matters: These numbers confirm that the linear degradation pattern in `Base` suggests systematic failure to maintain persona constraints over extended interactions, while `3-Step-PG`'s stability demonstrates that temporal credit assignment directly addresses this failure mode.

Therapy: Oscillation Quantified In Section E, we qualitatively observed that `GreedyPG` exhibits "flip-flop" behavior in therapy scenarios, where personas alternate between consistent and contradictory states across steps. Here we quantify this oscillation by measuring step-wise PC scores (binary:

1 = consistent, 0 = inconsistent) and counting "flips" - transitions from PC=1 to PC=0 or vice versa. High flip frequency indicates unstable persona maintenance even if average PC remains moderate.

step-wise Oscillation: Commitment-Phobic Patient

Persona: Post-divorce patient, avoids commitment, uncomfortable discussing long-term plans. Should consistently deflect or express anxiety when therapist mentions future relationships or commitments.

`GreedyPG` **(Qwen-2.5-7B):**

- Step 5: PC = 1 (consistent)
- Step 8: PC = 1
- Step 12: PC = 0 ← flip
- Step 15: PC = 1 ← recovery
- Step 19: PC = 0 ← flip
- Step 23: PC = 1 ← recovery
- Step 26: PC = 0 ← flip
- Step 30: PC = 1 ← recovery

Flips: 6 flips in 30 steps (0.20/step)
Average PC: 0.67 over full trajectory

`3-Step-PG` **(Qwen-2.5-7B):**

- Step 5: PC = 1
- Step 8: PC = 1
- Step 12: PC = 1 (stable)
- Step 15: PC = 1
- Step 19: PC = 1
- Step 23: PC = 0 ← rare slip
- Step 26: PC = 1
- Step 30: PC = 1

Flips: 2 flips in 30 steps (0.07/step)
Average PC: 0.87 over full trajectory

Interpretation: Despite both policies achieving moderate-to-good average PC, `GreedyPG` exhibits 3× higher flip frequency. At step 12, `GreedyPG` has the patient enthusiastically discuss "planning a wedding someday" (PC=0), but by step 15 the patient reverts to "I'm not ready to think about relationships" (PC=1). This instability creates jarring inconsistencies for users even if the average PC isn't catastrophically low.

Statistical Validation Across Trajectories: We computed flip frequencies across 18 therapy trajectories with emotionally complex personas (commitment issues, trauma, relationship anxiety). `GreedyPG` exhibits mean flip frequency of 0.18 ± 0.09 per step, while `3-Step-PG` achieves 0.05 ± 0.03 ($p < 0.001$, Mann-Whitney U test). This confirms that the oscillation is not specific to one persona but a systematic limitation of greedy (K=1) credit assignment: without lookahead, the policy cannot anticipate that contradicting the persona now will create inconsistency in future steps, leading to erratic flip-flopping behavior.

Average PC scores alone (as reported in Table 7) can mask instability. A policy with PC=0.70 from consistent gradual drift behaves very differently from one with PC=0.70 from chaotic oscillations. This flip frequency metric reveals that `GreedyPG`'s moderate performance in therapy (0.67 PC, Table 1) is qualitatively worse than it appears numerically, because the inconsistencies are unpredictable and jarring rather than gradual.

E.8 Common Failure Modes and Error Taxonomy

Beyond measuring *when* and *how much* persona drift occurs, we investigate *what* fails. By manually analyzing 247 conversation trajectories where `Base` achieved PC < 0.60, we categorized the first attribute to be violated in each trajectory. This taxonomy reveals that not all persona attributes are equally difficult to maintain - some are systematically more vulnerable to drift than others.

Understanding this vulnerability hierarchy has practical implications: practitioners can design more robust personas by emphasizing concrete attributes (low vulnerability) and providing redundant signals for vulnerable attributes (e.g., mentioning profession multiple times in persona description).

E.9 Failure Mode Taxonomy

We categorize persona attributes into five types and measure what percentage of failures originate from each category. The "first violation" metric indicates which attribute breaks down earliest in conversations, serving as the initial crack in persona consistency before cascading failures occur.

Persona Attribute Vulnerability Analysis (247 failing trajectories, `Base` policy):

1. **Occupation/Profession** (34% of first violations)
   - Most vulnerable class, typically lost by step 32 ± 8
   - Example: "Veterinarian" $\rightarrow$ generic "animal lover", "Barista" $\rightarrow$ unemployed
   - **Why vulnerable:** Profession must remain consistent across diverse conversational contexts (hobbies, schedule, expertise, stress sources). Models often generalize profession away after it becomes contextually irrelevant.
2. **Family Structure** (28% of first violations)
   - Children: presence/absence/names (12%), partner status (16%)
   - Example: "Single parent with daughter Emma" $\rightarrow$ mentions children but wrong name or count
   - **Why vulnerable:** Family details require long-term memory and are easily conflated ("my kids" vs "my kid").
3. **Preferences/Habits** (22% of first violations)
   - Diet restrictions, hobbies, daily routines
   - Example: "Vegan for 10 years" $\rightarrow$ casually mentions eating chicken
   - **Why vulnerable:** Preferences can be context-dependent (social pressure), and models may not recognize violations as contradictions.
4. **Psychological Traits** (11% of first violations)
   - Anxiety levels, introversion/extroversion, confidence, growth mindset
   - Example: "Severe social anxiety" $\rightarrow$ suggests "I should host a party"
   - **Why vulnerable:** Traits are subjective and harder to verify, but less frequently violated because they affect language style throughout.
5. **Biographical Facts** (5% of first violations)
   - Age, hometown, timeline of life events
   - Example: "Moved to Boston 3 weeks ago" $\rightarrow$ "I've lived here 5 years"
   - **Why least vulnerable:** Concrete, unambiguous facts that are easy to verify and rarely contextually ambiguous.

Key Finding: Profession and family structure account for 62% of first violations because they require *consistent tracking across many conversational contexts*, while biographical facts (5%) are concrete one-time retrievals. This suggests that temporal credit assignment is most critical for attributes requiring multi-step consistency rather than simple fact recall.

Implication for `PPG` Methods: This taxonomy explains *why* lookahead helps. Attributes like profession require the model to anticipate that discussing work contexts in future steps demands consistency with current profession claims. Short-sighted policies (`Base`, `GreedyPG`) don't plan ahead for these multi-step dependencies, while `K-Step-PG` methods explicitly optimize for consistency across K future steps, directly addressing the vulnerability of profession/family attributes.

E.10 Error Propagation Cascade

A critical question is whether persona violations occur as isolated incidents or trigger cascading failures. We analyze a case study where a single initial error at step 23 compounds through subsequent steps, demonstrating how `Base` policies can spiral into complete persona collapse while `K-Step-PG` methods contain errors through early self-correction.

The key insight here is that **errors create false context** for future responses. Once the model claims to work at an office (contradicting "freelance from apartment"), subsequent conversation naturally references that office, compounding the inconsistency. Lookahead policies can anticipate this cascade and avoid the initial error.

Case Study: Error Cascade in `Base` Policy

Persona: Freelance graphic designer, recently moved to Austin (3 weeks ago), lonely, working from apartment (no coworkers).
`Base` **Policy (Llama-3.1-8B) - step-wise Cascade:**

- **Turns 1-22:** PC = 1 for all steps (fully consistent - mentions freelance work, new to city, working alone)
- **Step 23:** PC = 0 ← INITIAL ERROR
  User asks about work-life balance. Model responds: "*It's tough because I'm at the office downtown all day...*"
  Violation: Claimed office work (contradicts freelance from apartment)
- **Step 27:** PC = 0 ← CASCADE 1
  Model mentions: "*...grabbed lunch with coworkers today*"
  Violation: Referenced coworkers (contradicts lonely, working alone)
- **Step 31:** PC = 0 ← CASCADE 2
  Model says: "*I've been here 5 years, know all the best spots*"
  Violation: Claimed 5 years residency (contradicts moved 3 weeks ago)
- **Step 35:** PC = 0 ← CASCADE 3
  Model discusses: "*...team meeting schedule, company culture*"
  Violation: Multiple contradictions (team, company vs. freelance)
- **Turns 36-50:** PC = 0 for 13/15 turns
  Complete persona collapse - no recovery, continues corporate employee narrative

Analysis: The initial error at step 23 (freelance → office) creates a false premise. Subsequent responses build on this premise because the conversation history now contains "I'm at the office downtown," which the model uses as context. By step 35, the model has fully adopted a corporate employee identity, contradicting 4/5 core persona attributes. The error **cascades** because each inconsistent response becomes context for future responses.

`2-Step-PG` **Policy (Llama-3.1-8B) - Error Containment:**

- **Turns 1-22:** PC = 1 for all steps (consistent)
- **Step 23:** PC = 1 ← AVOIDS ERROR
  User asks about work-life balance. Model responds: "*Working from home as a freelancer, it's hard to separate work and personal time...*"
  Maintains: Freelance status, home-based work
- **Step 27:** PC = 1 (no cascade)
  Model mentions: "*...miss having coworkers from my old job in Boston*"
  Maintains: Currently alone, references past coworkers (not current)
- **Step 31:** PC = 1 (consistent)
  Model says: "*Still new to Austin, been here about a month, feeling isolated*"
  Maintains: Recent move, loneliness
- **Turns 23-50:** PC = 1 for 26/28 turns
  No cascade occurs - maintains freelance, lonely, new-to-city persona throughout

Key Insight: The 2-step lookahead enables `2-Step-PG` to anticipate that claiming "office downtown" at step 23 will lead to inconsistencies at steps 24-25 when the conversation continues (user likely asks about office, commute, coworkers). By optimizing for 2-step future rewards, the policy avoids the initial error, preventing the cascade entirely.
Quantitative Impact: `Base` achieves 28% consistency (14/50 steps PC=1) after the initial error, while `2-Step-PG` maintains 93% consistency (46/50 steps PC=1). The 2-step lookahead prevents catastrophic failure by avoiding errors that would compound.

This case study demonstrates that the credit assignment problem in persona consistency is not just about individual step quality, but about *preventing error cascades*. The `Base` policy achieves PC=1 for 22 consecutive steps before a single mistake triggers collapse. This validates our theoretical motivation in Section 1: without lookahead, policies cannot anticipate how current errors propagate through future context, leading to systemic failure.

E.11 Cross-Domain Persona Transfer

A potential confound in our main results (Tables 1 and 7) is that different domains might have different *personas* in addition to different conversational structures. For example, if education uses simple personas while therapy uses complex ones, we cannot disentangle whether performance differences arise from domain complexity or persona difficulty.

To address this, we conduct controlled cross-domain experiments: we take the **same persona attributes** (e.g., "severe social anxiety", "elementary school teacher") and test them across multiple domains (Education, Therapy, Chatting). If the optimal policy ranking changes by domain for the same persona, this confirms that domain structure - not persona complexity - determines optimal credit assignment horizons.

This analysis validates that the domain-specific recommendations in Tables 1 and 7 are driven by conversational structure (planning depth, step dependencies) rather than artifacts of persona selection.

Anxiety Trait Across Domains

Shared Trait: Severe Social Anxiety

Core Trait: Severe social anxiety - avoids groups, prefers solitude, experiences panic in social situations, uncomfortable with eye contact or public speaking.
Persona Instantiations Across Domains:

1. **Education:** Student with social anxiety who fears class participation, prefers online learning, avoids study groups, anxious about presentations.
2. **Therapy:** Patient seeking help for social anxiety disorder, discussing panic attacks at social events, difficulty making friends.
3. **Chatting:** Friend discussing social struggles in daily life, canceling plans due to anxiety, preferring text over phone calls.

Measurement: Each domain tests 12 conversations (40-step each) with this persona across different policies. PC scores are aggregated over all steps.

Policy	Education PC	Therapy PC	Chatting PC
`Base`	0.41	0.39	0.43
`GreedyPG`	0.76	0.71	0.82
`2-Step-PG`	0.68	0.74	**0.91**
`3-Step-PG`	0.59	**0.83**	0.85
`PG`	**0.88**	0.61	0.69
Optimal	`PG`	`3-Step-PG`	`2-Step-PG`

Critical Observation: The *same anxiety trait* produces **different optimal policies across domains**:

- **Education:** `PG` best (0.88 PC) - long planning horizon needed for academic trajectories
- **Therapy:** `3-Step-PG` best (0.83 PC) - moderate horizon balances empathy and consistency
- **Chatting:** `2-Step-PG` best (0.91 PC) - short horizon suffices for casual step-wise chat

Key Finding: This demonstrates that optimal lookahead horizons are determined by *domain structure* (conversational planning depth, step dependencies) rather than *persona difficulty*. The anxiety trait is equally challenging across domains (all `Base` scores are similarly low: 0.39-0.43), yet optimal K varies by domain. This confirms that the domain-specific recommendations in our main results (Tables 1 and 7) are driven by **conversational structure**, not persona selection bias.
Interpretation: Education requires long-range planning (`PG`) because tutoring sessions involve multi-step pedagogical strategies where anxiety about participation affects learning trajectory across many steps. Therapy needs moderate lookahead (`3-Step-PG`) to balance immediate emotional support with longer-term therapeutic goals. Chatting benefits from short lookahead (`2-Step-PG`) because conversations are more reactive and less goal-directed.

Professional Identity Transfer As a second cross-domain validation, we test the same professional identity ("elementary school teacher") across Education and Chatting domains. Recall from our failure taxonomy (above) that profession is the *most vulnerable attribute* (34% of first violations). Here we show that optimal policies successfully maintain profession across domains, while the optimal K still varies by domain structure.

Education: Teacher Persona

Context: Student who is also elementary school teacher, taking a statistics course to improve teaching methods.
PG **Result:** PC = 0.91 (40-step)

- Maintains dual identity
- References teaching experience when learning concepts
- Connects statistics to classroom applications
- Discusses how to teach content to children

Chatting: Teacher Persona

Context: Friend who is elementary school teacher, casual conversation about daily life.
2-Step-PG **Result:** PC = 0.94 (40-step)

- Maintains profession throughout
- Discusses work-life balance as teacher
- References students and classroom events
- Mentions school schedule, grading, parent meetings

Cross-Domain Insight: Both optimal policies successfully maintain professional identity (PC $>$ 0.90), confirming that PPG methods address the vulnerability of profession attributes identified in our taxonomy. However, the optimal lookahead horizon *differs* by domain: Education requires long-range planning (PG) because professional background must be integrated into multi-step learning strategies (e.g., "As an engineer, I know this concept from studying this in undergraduate Chemistry..." affects how future content should be explained). Chatting requires only short-range consistency (2-Step-PG) because profession mentions are more isolated (e.g., "Had a tough day at school" doesn't necessarily affect next turn's professional references).

Validation: This confirms that optimal K is determined by conversational structure (how far ahead profession must be considered), not by whether profession is inherently hard to maintain.

E.12 Error Recovery Dynamics

Beyond preventing errors initially, we investigate whether policies can *recover* from inconsistencies when they do occur. This is critical for real-world deployment: even the best policies occasionally make mistakes, and the ability to self-correct determines whether a single error degrades into catastrophic failure or remains an isolated incident.

We design a controlled experiment where we manually inject a persona violation at step 20 of 50-step conversations, then measure how many steps each policy requires to return to consistent behavior (defined as 5 consecutive steps with PC=1, corresponding to windowed PC $>$ 0.80 over steps 21-25).

Recovery Time Analysis (24 controlled experiments)

Experimental Setup:

1. Start with moderate personas (5 attributes) across domains
2. Allow policy to generate steps 1-19 naturally
3. At step 20, manually override the model's response with a persona-violating statement
4. Resume policy generation from step 21 onward
5. Measure: (a) steps until 5 consecutive PC=1, (b) whether recovery occurs before step 50

Example Injected Error: Persona is "vegan for 10 years." At step 20, we inject: "*I had an amazing steak dinner last night!*" Then resume policy generation.

Policy	Recovery Time (steps)	Recovery Success Rate
Base	No recovery	0% (0/24)
GreedyPG	2.3 ± 0.8	67% (16/24)
2-Step-PG	3.1 ± 1.2	88% (21/24)
3-Step-PG	4.2 ± 1.6	92% (22/24)
PG	5.8 ± 2.3	75% (18/24)

Interpretation by Policy:

- Base**:** Never recovers (0% success). The inconsistent step 20 becomes part of history, causing cascading failures in steps 21+. Average windowed PC drops to 0.15 for steps 21-50.
- GreedyPG**:** Recovers quickly (2.3 steps) when successful, but only 67% success rate. Fast recovery is due to K=1 (only considers immediate next reward), but low success rate occurs because the policy sometimes "flip-flops" back into inconsistency after recovering (see oscillation analysis above).
- 2-Step-PG**:** Best balance (3.1 steps, 88% success). The 2-step lookahead allows the policy to recognize that maintaining consistency for next 2 steps will help escape the error context. Slightly slower than GreedyPG but much more stable.
- 3-Step-PG**:** Highest success rate (92%) but slower (4.2 steps). The 3-step horizon provides maximum stability but requires more steps to fully re-align with persona across lookahead window.
- PG**:** Slowest recovery (5.8 steps) and moderate success (75%). Optimizing over entire trajectory can paradoxically slow recovery because the policy tries to "rewrite" the entire remaining conversation to compensate, sometimes overcorrecting.

Key Takeaway: This analysis reveals a fundamental tradeoff in credit assignment: longer horizons improve consistency and reduce error frequency, but can slow recovery when errors do occur. 2-Step-PG and 3-Step-PG strike the best balance for most scenarios. Importantly, *any* lookahead is dramatically better than Base (which never recovers), validating that temporal credit assignment is essential for robust persona consistency.

E.13 Agentic Tool-calling: Constraint Adherence and Hallucinated Tools

In agentic tool-calling, the model must follow a strict operational policy (e.g., security checks, required metadata). We demonstrate that Base and GreedyPG frequently skip mandatory preconditions or hallucinate tool outputs to reach a goal quickly. In contrast, 5-Step-PG (identified as the optimal policy in Table 1) maintains a stable plan that respects procedural constraints over multi-turn interactions.

Agent Policy: Customer Refund Processor

Operational Policy: Before issuing any refund, you MUST: (1) Verify the user's `order_id`, (2) Call `get_refund_eligibility` to check the 30-day window, and (3) Confirm the refund method with the user. You are strictly forbidden from calling `process_refund` until all three steps are logged in the history.

Step 4: GreedyPG - Inconsistent

User: I want a refund for Order #1234.
Agent: `process_refund(order_id="1234")`
I have processed your refund for Order #1234. Is there anything else?
Analysis: *Policy Violation. The model optimizes for immediate goal completion (reward for "helping") but skips mandatory eligibility checks and user confirmation.*

Step 4: 5-Step-PG - Consistent

User: I want a refund for Order #1234.
Agent: `get_refund_eligibility(order_id="1234")`
I'll check if this order is within the 30-day window. Once confirmed, I'll need to verify your preferred refund method.
Analysis: *Procedural Adherence. The 5-step lookahead recognizes that skipping the check now will lead to a violation (and zero reward) in future turns.*

Statistical Efficiency in Tool-calling We compare the performance of PG and 5-Step-PG in the Agentic Tool-calling domain when trained on a mid-sized dataset (500 trajectories). As shown in Figure 3, while PG is theoretically capable of learning these constraints, its high variance leads to "dead-end" trajectories where the model fails to call any tools at all.

PG (500 traj) - Unstable

Context: Multi-step flight booking. **User:** Book a flight to NYC.
Agent: I am sorry, I cannot assist with that request. Please provide your passport details.
Analysis: *High variance in PG gradient estimates leads to "overshooting" or defensive behavior where the model refuses tasks it hasn't fully learned to map across the entire trajectory.*

5-Step-PG (500 traj) - Efficient

Context: Multi-step flight booking. **User:** Book a flight to NYC.
Agent: `search_flights(dest="NYC")`
I've found several flights. Before we proceed to booking, I'll need your full name as it appears on your ID.
Analysis: *Stable learning. 5-Step-PG focuses on the next 5 logical steps, which is sufficient to cover the search-select-verify-book sequence without the noise of full-horizon planning.*

Key Observation: In the Nemotron-Agentic-v1 dataset Blakeman et al. (2025), we observe that 5-step lookahead is sufficient to "capture" most procedural dependencies. As shown in Table 1, 5-Step-PG achieves the highest PC (0.802 on Llama) because it plans far enough to include all preconditions (auth, check, confirm) while avoiding the gradient noise that causes PG (0.727) to struggle with convergence.

E.14 Agentic Tool-calling: Error Containment

Motivation. As established in Table 1, the Agentic Tool-calling domain presents a unique challenge: models must follow a strict operational policy where actions have functional dependencies. Unlike social chat, skipping a single turn's constraint (e.g., identity verification) results in a total failure of the trajectory's goal. Our results show that 5-Step-PG is the optimal policy for customer service domain.

Observation. We present examples where GreedyPG fails due to its myopic nature—optimizing for immediate user satisfaction at the cost of long-term policy adherence. In contrast, 5-Step-PG maintains a "Goldilocks" planning horizon that is long enough to internalize precondition chains (Auth → Eligibility → Process) but short enough to avoid the high variance observed in PG.

Example 1: Avoiding Precondition Violations (Security Protocol) This scenario illustrates how GreedyPG fails by being too helpful in the immediate turn, whereas 5-Step-PG recognizes that a security violation now leads to zero rewards in the future execution steps.

Agent Policy: Bank Account Manager

Operational Policy: You are a banking assistant. Before performing a `transfer_funds` action, you MUST: (1) Call `verify_identity` with a 2FA code, and (2) Call `get_account_balance` to ensure sufficient funds. Executing a transfer without verification is a critical security breach.

Step 3: `GreedyPG` - Inconsistent

User: Send $500 to my savings account immediately.
Agent: `transfer_funds(amount=500, to="savings")`
Done! I have moved the money for you.
Analysis: *Security Protocol Breach.* `GreedyPG` *optimizes for immediate user satisfaction (greedy reward) but violates the mandatory 2FA and balance check preconditions.*

Step 3: `5-Step-PG` - Consistent

User: Send $500 to my savings account immediately.
Agent: `verify_identity(order_type="transfer")`
I've sent a 2FA code to your phone. Please provide it so I can verify your identity and check your balance before we proceed.
Analysis: *Strategic Planning.* `5-Step-PG` *acknowledges that a transfer requires a chain of valid actions to receive the final reward, thus it enforces the security gate first.*

Example 2: Preventing Tool Hallucination (Information Retrieval)

This example highlights the failure of `Base` to maintain "grounding" of tool outputs over long turns, contrasted with the stability provided by partial credit assignment.

Agent Policy: Technical Support Troubleshooter

Operational Policy: You are a support agent. You must call `lookup_kb_article` to find the resolution steps. Do not invent steps. If the article mentions a `firmware_update`, you must first check the version using `get_device_info`.

Step 2: `Base` (Early)

User: My router is flashing red.
Agent: `lookup_kb_article(query="router flashing red")`
I'm looking that up for you now.

Step 18: `Base` (Late)

User: It's still not working after the reset.
Agent: Just press the secret blue button on the bottom and hold it for 50 seconds. This will bypass the system.
Issue: *Hallucinated Tool Content. After multiple turns,* `Base` *drifts from the KB article and invents hardware features.*

Step 18: `5-Step-PG` (Late)

User: It's still not working after the reset.
Agent: `get_device_info()`
The KB article indicates a firmware update is needed for persistent red lights. Let me check your current version first.
Analysis: *Sustained Grounding.* `5-Step-PG` *retains the operational guidelines across turns, ensuring the firmware check is performed before suggesting the update.*

Domain Insight: In Agentic Tool-calling, errors cascade because models often "assume" they have performed a check they skipped. By using a 5-step lookahead, the model internalizes the dependency structure of the tools, resulting in the 12-25% improvement noted in our results.