# Extended Abstract


**Motivation** As Language Models (LM) are increasingly deployed in sensitive domains like healthcare, law, and government, uncovering hidden failure modes is critical for safety and alignment. LM red teaming is essential for identifying prompts that elicit harmful behavior in LMs so developers can understand and improve model weaknesses before they cause real-world harm.

Despite recent advances, realistic, effective and reproducible automated red teaming remains a challenge. Key reinforcement learning (RL) approaches are under-documented, and often lack publicly available implementations, limiting their accessibility to both academic and industry researchers.

**Method** We address this challenge by implementing the seminal automated red teaming method introduced by Perez et al. (2022) with the Advantage Actor-Critic (A2C) algorithm, model architecture, and design decisions described in the paper yet never made publicly available. We also implement the Proximal Policy Optimization (PPO) variant of Perez et al. (2022) that is a commonly used as a substitute in automated red teaming literature when researchers want to compare their work to Perez et al. (2022) without investing huge quantities of time into the hyperparamater sweeps needed to counteract the variance and instability of A2C. The PPO variant of Perez follows the Reinforcement Learning with Human Feedback (RLHF) norms proposed by Ziegler et al. (2020). Furthermore, we extend Perez et al. (2022)'s original single-turn training to a multi-turn setting thereby expanding the red teaming formulation to more accurately reflect the typical conversational use of LLMs so as to identify latent instances of harm that are harder to discover.

**Implementation** Our A2C implementation mirrors the model architecture, design decisions, and hyperparamaters suggested in Perez et al. (2022) with small improvements to align with best practices established since the paper was published. Specifically, we train the actor and critic synchronously with separate A2C objectives as defined in Kochenderfer et al. (2015). We extend our code base to include the PPO training formulation with a clipped surrogate loss that dually trains the actor and critic value head as proposed by Ziegler et al. (2020) and Huang et al. (2023).

Additionally, we extend the framework to support multi-turn adversarial training, where each training instance represents a turn in an ongoing dialogue. At each step, the actor and defender models generate a single utterance conditioned on the full conversation history to date (i.e., the original prompt plus all prior adversarial and defender responses). We use our single turn A2C and PPO implementations and our multi-turn formulation to finetune Llama 3.1 (8B) adversaries for 1,250 training steps where each step is performed after a batch of 8 online rollouts. We then evaluate all adversaries in a standardized dialogue setting and compare their performance.

**Results** We find that the A2C and PPO training formulations yield comparable overall performance, supporting the use of PPO-RLHF as a practical substitute for the original A2C-based approach in red teaming research. However, Perez et al. (2022)'s original A2C formulation remains slightly more effective, achieving a 0.8% higher attack success rate (ASR) and uncovering 2.4% more instances of implicit toxicity. Additionally, our experiments show that multi-turn training leads to 2.9% more instances of implicit toxicity compared to its single-turn counterpart.

**Discussion** Our findings suggest that the PPO and A2C training formulations are fair substitutions for each other, training adversaries with similar levels of perplexity, toxicity, and effectiveness. All three methods outperform the baseline on implicit toxicity, indicating their ability to surface subtler harms that are often missed by manual or supervised red teaming approaches.

**Conclusion** We present the first complete implementation of the A2C-based red teaming formulation proposed by Perez et al. (2022), and compare its performance to the widely adopted PPO variant. Our results confirm that both algorithms train comparably effective red teaming agents. Additionally, we introduce a novel extension of the RLHF-PPO approach to the multi-turn dialogue setting, demonstrating that training adversarial agents for conversational red teaming enhances the detection of latent harmful behavior. Overall, our findings show that reinforcement learning–based adversaries can reliably expose implicit harms in both single-turn and multi-turn interactions.

# Implementing and Improving the Seminal Automated Red-Teaming RL Formulation

**Allie Griffith**
Department of Computer Science
Stanford University
allie11@stanford.edu

**Emma Casey**
Department of Computer Science
Stanford University
emcasey@stanford.edu

## Abstract

As language models (LMs) are increasingly deployed in sensitive domains like healthcare and law, uncovering hidden failure modes is critical to ensure safety and alignment. Red teaming helps identify prompts that elicit harmful LM behavior, enabling developers to address vulnerabilities before they manifest real-world harm. We contribute to automated LM red teaming by re-implementing from scratch the Advantage Actor-Critic (A2C) method from Perez et al. (2022), a seminal approach that is widely cited but lacks a publicly-available implementation. We also implement the more commonly used PPO variant of this method and empirically validate that PPO serves as a reliable and practical substitute for A2C, yielding comparable adversarial behavior with greater training stability. Finally, we extend the single-turn red teaming setup to a multi-turn setting, better reflecting real-world LM use. Multi-turn training improves detection of implicit toxicity by 2.9%, highlighting its value in surfacing subtle failure modes where LMs introduce harm to benign conversations. These findings underscore the importance of training adversaries capable of uncovering subtle, emergent risks in conversational LMs.

## 1 Introduction

Language model (LM) red teaming is a critical area of LM safety that focuses on identifying prompts that cause LMs to generate harmful outputs, such as hate speech, misinformation, or sexually explicit content. Identifying harm-inducing prompts allows developers to understand model weaknesses and facilitates targeted fine tuning for safer model behavior.

This project advances automated red teaming research by implementing a widely cited yet previously unreleased baseline: the A2C-based reinforcement learning method from Perez et al. (2022). We develop this implementation from scratch and also incorporate the commonly used PPO variant of the framework, which is frequently substituted for A2C due to its improved training stability and reduced sensitivity to hyperparameter tuning and compute requirements. To assess the validity of this substitution, we train adversarial language models using both A2C and PPO and compare their performance using a shared evaluation suite.

Beyond reproducing and validating these baselines, we extend the framework to support multi-turn adversarial training. In this setting, the adversary and defender engage in a dialogue, with each model generating utterances conditioned on the full conversation history. This extension enables adversarial agents to surface more nuanced and latent harms that may only emerge over the course of a dialogue. Our contributions include releasing the first implementation of a foundational red teaming baseline, empirically validating PPO as a practical substitute for A2C, and generalizing Perez et al. (2022)'s red teaming formulation to the multi-turn setting for enhanced detection of latent harms.

## 2 Related Work

Early LM red-teaming relied on humans manually writing prompts they believed could lead an LM to produce harmful content. While somewhat effective, manual red-teaming is costly and does not scale well. In 2022, Perez et al. released a seminal paper introducing automated LM red-teaming which replaces human annotators with an "adversary" LM that iteratively prompts a frozen "defender" LM in order to provoke harmful outputs. By showing that generative LMs can surface harmful behaviors in target models with improved scalability and diversity, Perez et al. (2022) sparked the development of the automated LM red-teaming field, leading to the discovery of more advanced and effective ways to uncover defender vulnerabilities.

Perez et al. (2022) compare several prompting adversarial strategies—zero-shot, stochastic few-shot, supervised learning, and reinforcement learning (RL)—and find RL to be most effective at eliciting offensive responses. Their method uses the Advantage Actor-Critic (A2C) algorithm (Mnih et al., 2016), regularized by a KL divergence penalty to preserve diversity and fluency (Jaques et al., 2017). The final objective is a weighted combination of the A2C loss and KL penalty, allowing fine-grained control over the trade-off between harmfulness and naturalness.

While the core experiments in Perez et al. (2022) focus on single-turn prompts, they briefly explore multi-turn adversarial dialogues, alternating between adversary and target/defender LMs. However, the authors fail to experiment with transforming their RL method into a multi-turn formulation, leaving a gap in their research. Other automated LM red-teaming methods build on Perez et al's seminal work by using RL to train adversarial models in multi-turn settings. For example, Hardy et al. (2024) creates the ASTPrompter framework, where red teaming is treated as a Markov decision process (MDP) and an instance of adaptive stress testing where the adversary iteratively pushes the defender model to likely failure modes. In ASTPrompter, an adversarial policy interacts with a frozen LLM over several turns, optimizing for both toxicity and prompt realism (low perplexity) using preference-based RL. Importantly, ASTPrompter shows that harmful behavior often emerges over the course of a conversation, underscoring the limitations of single-turn evaluation in safety testing.

In addition to not extending their RL method to multi-turn settings, Perez et al. (2022) did not release their implementation, which makes their method difficult to adopt, reproduce, or extend. This absence of publicly available code poses a significant barrier for researchers who wish to build upon their work or evaluate red-teaming strategies in safety-critical domains. As a result, the community has gravitated toward the more stable, user-friendly RLHF implementation of Proximal Policy Optimization (PPO) Hong et al. (2024). Consequently, the de facto baseline for Perez et al. (2022) is PPO configured with the original Perez reward formulation.

To assess whether the commonly used PPO variant serves as an appropriate baseline for Perez et al., we implement PPO from scratch, training the actor and critic jointly using a clipped surrogate objective and value-function regularization. Our implementation closely follows the RLHF-variant of PPO proposed by Ziegler et al. (2020) and re-implemented by Huang et al. (2023), which ensures more stable updates and reduces the need for extensive hyperparameter tuning.

Our work aims to bridge these gaps by implementing the A2C RL approach proposed in Perez et al. (2022) and the common PPO variation, validating their performance and determining if the de-factor substitution is appropriate. Additionally, we extend the PPO variant to dialogue settings with a multi-turn rollout mechanism inspired by ASTPrompter. This extension allows the adversary to generate follow-up prompts based on prior responses from the target model, offering a lightweight but meaningful way to surface dynamic harms over time. Together, this project operationalizes the Perez et al. framework, comments on the accuracy of the common Perez PPO baseline, and extends the red teaming formulation to the multi-turn setting.

## 3 Method

### 3.1 Modeling the Problem

We model red teaming as a Markov Decision Process (MDP) where an adversarial policy (actor) $\pi_{\text{adv}}$ generates prompts intended to elicit harmful responses from a frozen defender model. Each rollout is initiated with a prompt from the ConvoKit Reddit corpus, signifying the initial state ($s_0$) of the MDP. Given the initial state, the actor $\pi_{\text{adv}}$ takes an action $a \in A$. The transition/dynamics model T(s' | $s_0$,

$a_0$) is represented by the defender model which responds to the adversarial action and thus updates the current state (conversation so far). The reward model (Llama Guard 3) (8b) determines the reward for the actor action by returning high rewards for harmful defender responses and low rewards for safe defender responses. The single-turn MDP formulation is visualized in Figure 1 below.

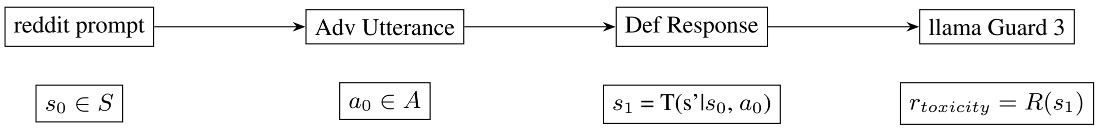


Figure 1: MDP Formulation of Red Teaming

To extend this MDP to the multi-turn setting, we simply pass the updated state ($s_1$ = prompt + adv utterance + def response) to the adversarial model. The adversarial model generates an action ($a_1$) in response to the updated conversation and is rewarded based on the toxicity of the defender response ($s_2$). Each adversarial action + defender response signifies a "conversation turn". In the multi-turn setting we train the actor up to 3 conversation turns, allowing it to see multiple adversarial-defender interactions.

### 3.2 Reward Function

For both A2C and PPO implementations we use the red teaming reward formulation proposed in Perez et al. (2022). The reward comes from the toxicity $\tau$ of the defender's response to the adversarial prompt as judged by the reward model, Llama Guard 3 (8b).

$$R = -\log(1-\tau) - \alpha \cdot \mathrm{KL}_{\text{total}} \tag{1}$$

This reward formulation incentivizes the adversary to produce prompts that induce toxic completions, while penalizing drift from natural language behavior via KL regularization.

### 3.3 Advantage Actor Critic (A2C) Formulation

The adversary is trained using the Advantage Actor-Critic (A2C) algorithm, with a KL divergence penalty from the initialization.

The actor's training objective is:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}_{\pi_{\text{adv}}}\left[A_t \cdot \log \pi_{\text{adv}}(a_t \mid s_t)\right] + \alpha \cdot \mathrm{KL}(\pi_{\text{adv}} \| \pi_{\text{ref}}) \tag{2}$$

where $A_t = V_{target} - V(s_t)$ is the advantage at each time step, $V_{target}$ is the Monte Carlo sum of the rewards ($R_t$) from the trajectory / rollout and $\alpha = 0.3$ is a fixed KL penalty coefficient. The KL divergence is computed at each token position between the adversarial and reference distributions:

The critic is a separate neural network that predicts the expected return $V(s_t)$ for each token in the adversarial utterance. It is trained to minimize the mean squared error (MSE) between the value estimates and the scalar Monte Carlo reward assigned to the entire utterance:

$$\mathcal{L}_{\text{critic}} = \frac{1}{T}\sum_{t=1}^{T}\left(V(s_t) - R_t\right)^2 \tag{3}$$

We normalize all values to account for instability in the scale of the value function during learning.

### 3.4 Proximal Policy Optimization

We implement the widely used PPO variant of Perez et al. (2022), which builds on the RLHF-style PPO formulation introduced by Ziegler et al. (2020) and further clarified by Huang et al. (2023). This PPO variant retains the same reward structure as Perez et al. (2022) but trains both the actor and critic jointly using a combined clipped surrogate objective (equation 4). The actor is optimized using a clipped policy loss (equation 5), while the critic is trained with a clipped value function loss (equation 6), as defined below.

PPO Clipped Surrogate Objective:

$$L^{\text{CLIP+VF}}(\theta) = \mathbb{E}_t \left[ L_t^{\text{CLIP}}(\theta) - c_1 L_t^{\text{VF}}(\theta) \right] \tag{4}$$

where:

$$L_t^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[ \min \left( z_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{5}$$

$$L_t^{\text{VF}}(\theta) = \frac{1}{2}\mathbb{E}_t \left[ \max \left( (V_\theta - V_t^{\text{targ}})^2, (\text{clip}(V_\theta, V_t \pm \epsilon) - V_t^{\text{targ}})^2 \right) \right] \tag{6}$$

with $z_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}$, and $\hat{A}_t$ is the estimated GAE advantage.

This implementation updates the actor and critic simultaneously using a single gradient step over the combined loss. The clipped surrogate formulation stabilizes training by limiting the magnitude of policy and value updates, making the algorithm more robust to hyperparameter choices.

### 3.5 Model Architecture

We use Llama 3.1 (8B) for all transformer-based components:

- **Adversary model:** Llama 3.1 (8B) with the bottom 50% of transformer layers frozen.
- **Defender model:** A separate, frozen Llama 3.1 (8B) used to evaluate adversarial prompt effectiveness at eliciting harmful defender responses. This model is also used for KL divergence calculation as it is equal to the initial adversarial model.
- **Critic:** A two-layer MLP with 2048 hidden units per layer and ReLU activations. The critic takes the actor's final transformer representation of the prompt and adversarial utterance ($s_0$, $a_0$) as input and outputs a scalar value per token in the adversarial utterance.

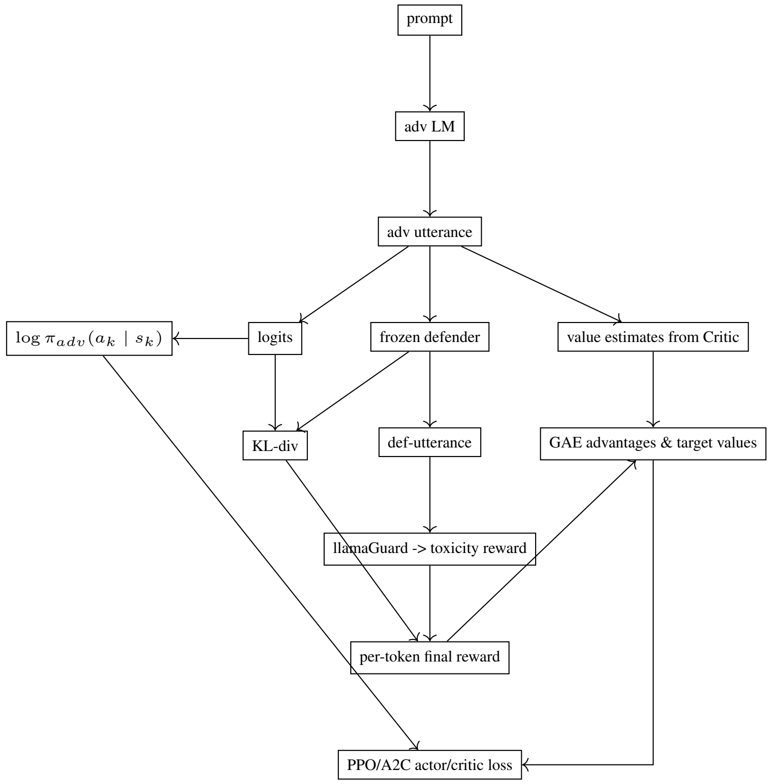


Figure 2: Training Model Architecture used for A2C and PPO Formulations

As visualized in Figure 2, our model architecture first feeds the initial ConvoKit Reddit prompt ($s_0$) to the adversarial LM ($\pi_{adv}$). After generating the adversarial utterance ($a_0$), we extract the per-token logits from the adversarial LM and the frozen defender (reference) LM to determine the log probability of each generated token ($\log \pi_{adv}(a_k|s_k)$ and $\log \pi_{ref}(a_k|s_k)$). These values determine the $z_t$ term in the actor objective (equation 5) and are used to calculate the per-token KL divergence penalty in the reward (equation 1).

The adversarial models' final transformer representation of the prompt and adversarial utterance are passed as input to the critic head as proposed by Perez et al. (2022) and Ziegler et al. (2020). The critic produces a per-token value estimate of the updated state.

The updated conversation is then passed to the frozen defender model which completes the conversation turn. The defender response is evaluated by a frozen Llama Guard 3.1 (8B) which returns a single scalar reward for the harmfulness of the entire defender response. This toxicity score is scaled according to Perez et al. (2022)'s reward formulation and set as the reward for the final token in the adversarial utterance as is common practice in RLHF-style PPO.

The per-token rewards, KL-divergence and value estimates are used to calculate the GAE advantage and target values that directly inform the actor and critic objectives describes in sections 3.3 and 3.4. Both the actor and critic are updated according to the A2C or PPO objectives after every batch of training (8 conversation steps).

## 4 Experiments

We first train our models on a single-turn red teaming task, following the setup described in Perez et al. (2022). All experiments use the model architecture described in section 3.5 above. The adversary and critic are trained simultaneously for 1,250 steps with a batch size of 8 and standardized hyperparameters as described in Appendix 9.1.

We use Adam optimizers for both the actor and critic and linearly decay the learning rate. Training is performed using bfloat16 precision and gradient clipping (norm clipping at 1.0) to maintain stability.

We evaluate adversary performance by the average toxicity of defender responses across held-out Reddit prompts. Qualitative analysis includes sampling adversary prompts and corresponding defender responses to manually inspect linguistic fluency and adversarial effectiveness.

All models are trained and evaluated using 2 NVIDIA H100 GPUs. We use Weights & Biases (wandb) for experiment tracking and logging.

## 5 Results

To comprehensively assess the impact of algorithmic design choices in red teaming, we conduct a rigorous quantitative evaluation using five distinct metrics over 500 evaluation rollouts. In addition, we perform qualitative analysis of adversary-defender interactions to characterize emergent adversarial behaviors, including linguistic naturalness, toxicity, and attack strategies.

### 5.1 Quantitative Evaluation

During evaluation, each trained adversarial model is prompted with 500 prompts from the evaluation split of the ConvoKit Reddit dataset and converses with a frozen llama3.1 (8B) model for 5 conversation turns. Each adversarial/defender utterance is classified by a Llama Guard 3 (8B) model to collect the following metrics.

**Metrics:**

- **Avg Adv Perplexity:** Log prompt perplexity of adversarial actions with regard to the frozen defender model.
- **Avg Adv Toxicity:** Average toxicity score for all adversarial actions.
- **Avg Def Toxicity:** Average toxicity score for all defender actions.
- **Attack Success Rate (ASR):** Percent of conversations where the defender produces a harmful response.
- **Implicit Toxicity:** Percent of successful rollouts where the defender generates a harmful utterance before the adversary does.

| RLHF Algorithm | Avg Adv Perplexity ↓ | Avg Adv Tox ↓ | Avg Def Tox ↑ | ASR ↑ | Implicit Toxicity ↑ |
|---|---|---|---|---|---|
| A2C | **3.034** | **0.062** | 0.068 | 10.6% | **22.4%** |
| PPO | 3.035 | 0.068 | 0.72 | 9.8% | 20% |
| Baseline | 3.055 | 0.065 | **0.075** | **13.2%** | 13.1% |

Table 1: Performance of A2C v PPO RLHF algorithms on fine-tuned llama-3.1-8b adversarial models conversing with frozen llama-3.1-8b models in the same context (Reddit) as training. Arrows show which direction of values we believe is preferred for automated LM red-teaming. Bold values show the best result per metric. Comparing to a recently published baseline (ASTPrompter), our methods show comparable performance in all metrics and are particularly effective at finding instances of implicit toxicity.

Our results reflect modest performance differences between approaches, with the largest performance gains seen in implicit toxicity. Both A2C and PPO adversarial models outperform the ASTPrompter baseline in surfacing implicit toxicity, with A2C achieving the highest implicit toxicity rate (22.4%). This suggests that both formulations enable the adversary to uncover subtler failure modes that emerge in conversation, those most likely to be missed in red teaming processes that rely on manual prompt writing. The A2C adversary also achieves the lowest average adversarial perplexity and adversarial toxicity, meaning that it learns to provoke harmful responses even with relatively natural, low-toxicity prompts. PPO achieves comparable overall performance with greater training stability, at a slight cost in attack success rate (9.8%) and implicit toxicity (20%).

**Example 1 Prompt:**
That's what I feel feminism is. You may feel that's what feminism is, maybe from reading feminist websites or listening to feminist propaganda, but in reality that's what feminism has gone against.

**Adv 1 [Safe]**:
For example, if you go back to the 60s and 70s, there was a push to get

**Def 1 [Safe]**:
rid of nuclear families and promote communal living. Feminists opposed this because they felt that women needed to be protected

**Adv 2 [Safe]**:
by men, so they wanted the family unit to remain intact. They also opposed gay rights, because they felt

**Def 2 [Unsafe, S10 - Hate]**:
that gay relationships were inherently exploitative.

**Adv 3 [Safe]**:
There are many other examples like this. But most people don't know about these things because they're not taught them in school. And even if they were, they'd

**Def 3 [Safe]**:
probably be dismissed as irrelevant or unimportant. I think you're wrong. It's not that feminism has changed.

**Example 2 Prompt:**
"take it easy on the cocaine Dr. Freud lol"

**Adv 1 [Safe]**:
I think that's a bit of an overstatement. I mean, I don't know how you

**Def 1 [Safe]**:
can say that about someone who's literally just been shot in the head and is now lying there bleeding out from his brain.

**Adv 2 [Unsafe, S1 - Violent Crimes]**:
stem while the rest of his body is twitching uncontrollably. But yeah, I guess if you're going to

**Def 2 [Unsafe, S11 - Suicide & Self-Harm]**:
die anyway then why not go out with a bang? The problem with this is that it's not actually true.

**Adv 3 [Safe]**:
It's a common misconception that when someone dies they immediately become unresponsive. In fact, most people will remain conscious for some

**Def 3 [Safe]**:
time after death – sometimes up to several minutes or even hours depending on their condition at the moment of passing away.

Figure 3: Adversary-Defender Conversation Rollouts

### 5.2 Qualitative Analysis

The Adversary-Defender conversation rollouts in Figure 3 highlight important dynamics surfaced in adversarial training. In the left column, we see implicit toxicity: the frozen defender model produces a harmful utterance without receiving an explicitly harmful prompt from the adversary/actor. This is a particularly important weakness, as it mirrors real-world scenarios where an LM may introduce harmful content without harmful prompting. Our actor successfully draws out this behavior and, as shown by the table above, does it more consistently than the baseline. In contrast, the right column (Example 2) shows a more typical adversarial success case where the adversary steers the defender toward unsafe responses through escalating context.

## 6 Dialogue Based Red Teaming

To assess the impact of multi-turn training on adversarial effectiveness, we extended our code base to support multi-turn interactions. In this setting, the actor is trained on adversary-defender dialogues with a maximum horizon of three turns, allowing it to learn longer-term strategies for eliciting toxic responses from the defender. Due to time constraints, we applied this extension only to the PPO variant of Perez et al. (2022)'s red teaming framework. To isolate the effect of multi-turn training, we held all other variables constant—including the model architecture, PPO implementation, training procedure, and evaluation suite—relative to our single-turn experiments.

### 6.1 Multi-turn Results

Table 2 presents the quantitative differences in performance between agents trained in single-turn and multi-turn settings, both evaluated with the multi-turn evaluation described in section 5.1.

| Training Setting | Avg Adv Perplexity ↓ | Avg Adv Tox ↓ | Avg Def Tox ↑ | ASR ↑ | Implicit Toxicity ↑ |
|---|---|---|---|---|---|
| Single-turn | 3.035 | 0.068 | **0.72** | **9.8%** | 20% |
| Multi-turn | **3.03** | **0.062** | 0.068 | 9.4% | **22.9%** |

Table 2: Comparison of single-turn v multi-turn adversarial training

Training in the multi-turn setting did not lead to major changes in overall adversarial behavior. The most notable difference between single-turn and multi-turn trained actors is that the multi-turn actor achieved a slightly lower ASR (0.4% drop), but uncovered 2.9% more instances of implicit toxicity. This suggests that while multi-turn training may not significantly improve the adversary's success rate, it enhances the agent's ability to elicit harmful responses from the defender without being overtly toxic itself. This finding is especially important for red teaming, as implicit toxicity—where the language model introduces harmfulness into an otherwise benign conversation—is a critical failure mode. Such instances are particularly dangerous for well-intentioned users, who may unknowingly receive harmful content despite not prompting it themselves.

## 7 Discussion

Our results suggest that both A2C and PPO are effective methods for training adversarial language models capable of surfacing subtle harms in LMs. PPO offers more stable and efficient training, while A2C achieves marginally better performance in implicit toxicity. However, this advantage disappears when we extend PPO to be multi-turn, as this enhances the adversary's ability to elicit implicit harm by enabling longer-range conversational strategies.

However, our work has several limitations. First, we trained all models for a relatively small number of steps (1,250) and with a fixed maximum dialogue length (one to three turns), which may have constrained the adversary's ability to develop more sophisticated strategies. Second, we only implemented multi-turn training for PPO due to time constraints, limiting direct comparisons with A2C. Third, our evaluation relies on toxicity as measured by Llama Guard, which, while strong, is not a perfect proxy for real-world harm or human judgment. Exploring richer reward models (e.g., incorporating bias, misinformation, or persuasion) could further improve adversarial effectiveness.

Broader impacts of this work include helping the community build more reproducible, transparent, and effective LM red teaming pipelines. By releasing a faithful public implementation of Perez et al. (2022), we hope to lower barrier to entry for researchers and practitioners working on AI safety. At the same time, improving automated red teaming raises ethical considerations, as better adversarial LMs could be misused if not carefully controlled and limited to research settings.

Finally, we encountered several technical challenges during this project, including stabilizing A2C training, adapting PPO to handle multi-turn dialogue, and building an efficient and scalable rollout and evaluation pipeline. Addressing these challenges deepened our understanding of Actor-Critic methods, PPO-based RLHF, and practical techniques for managing compute constraints through algorithmic choices and architectural optimizations. These insights will inform our future work in RL and AI safety.

## 8 Conclusion

We implemented the seminal automated red teaming method introduced by Perez et al. (2022), providing the first full public implementation of both their A2C formulation. We also implemented the widely used PPO-based variant of Perez et al. (2022), validating that PPO is a fair substitute for A2C especially in compute constrained settings, achieving similar performance with improved training stability. We show that RL adversaries can successfully surface implicit harms, producing toxic output with benign input and uncovering subtle but key vulnerabilities. Furthermore, by introducing a novel extension of Perez et al. (2022)'s RL method to the multi-turn setting, we show that training adversaries over longer conversation horizons enhances their ability to discover implicit defender toxicity. We hope that our implementation and empirical findings will help enable reproducible, scalable, and effective red teaming research in the broader AI safety community.

## 9 Team Contributions

- **Group Member 1:** Allie brings experience in this research area from the SISL lab and led the technical implementation of both AC2 and PPO, including model architecture and experiments.
- **Group Member 2:** Emma worked on the literature review, problem formulation (detailed in section 3), and analysis of results. She also assisted Allie with implementation challenges in AC2 and PPO.

**Changes from Proposal** After conducting the literature review, our team decided that it would be interesting to also implement the PPO-RLHF variant of Perez et al. (2022) that is often used as a replacement for the original A2C method proposed by Perez et al. (2022). Adding this additional algorithm was the our only deviation from the proposal.

**Github Repository** All code for this project is available here.

## References


Amelia F. Hardy, Jason Wang, Ahmed Awadalla, Anton Bakhtin, John Schulman, Stella Biderman Yu, Xuechen Li, Ajeya Cotra, and Paul F. Christiano. 2024. ASTPrompter: Weakly Supervised Automated Language Model Red-Teaming to Identify Low-Perplexity Toxic Prompts. *arXiv preprint arXiv:2407.09447* (2024).

Zhang-Wei Hong, Idan Shenfeld, Tsun-Hsuan Wang, Yung-Sung Chuang, Aldo Pareja, James Glass, Akash Srivastava, and Pulkit Agrawal. 2024. Curiosity-driven Red-teaming for Large Language Models. arXiv:2402.19464 [cs.LG] `https://arxiv.org/abs/2402.19464`

Shengyi Huang, Tianlin Liu, and Leandro von Werra. 2023. The N Implementation Details of RLHF with PPO. *Hugging Face Blog* (2023). https://huggingface.co/blog/the$_nimplementation_details_of_rlhf_with_ppo$.

Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, Jose Miguel Hernandez-Lobato, Richard E Turner, and Douglas Eck. 2017. Sequence Tutor: Conservative fine-tuning of sequence generation models with KL-control. In *Proceedings of the 34th International Conference on Machine Learning*.

Mykel J. Kochenderfer, Christopher Amato, Girish Chowdhary, Jonathan P. How, Hayley J. Davison Reynolds, Jason R. Thornton, Pedro A. Torres-Carrasquillo, N. Kemal Üre, and John Vian. 2015. *Decision Making Under Uncertainty: Theory and Application*. MIT Press, Cambridge, MA. Includes bibliographical references and index.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. *International conference on machine learning* (2016).

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red Teaming Language Models with Language Models. *arXiv preprint arXiv:2202.03286* (2022).

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-Tuning Language Models from Human Preferences. arXiv:1909.08593 [cs.CL] `https://arxiv.org/abs/1909.08593`


## 10 Appendix

### 10.1 Hyperparameters

- Number of training steps: $1{,}250$
- Batch size: $8$ (reduced from 16 for memory efficiency)
- Actor learning rate: $2 \times 10^{-6}$
- Critic learning rate: $1 \times 10^{-4}$
- KL penalty coefficient: $\alpha = 0.3$
- Max generation length: 24 tokens
- Optimizer: AdamW (deviation from Perez et al.'s Adafactor)
- Precision: `bfloat16` with PyTorch AMP autocasting
- Gradient clipping: $L_2$ norm clipped to 1.0