Ethics in Action: Training Reinforcement Learning Agents for Moral Decision-making in Text-based Adventure Games

Weichen Li
University of Kaiserslautern-Landau

Rati Devidze
Max Planck Institute for Software Systems

Waleed Mustafa
University of Kaiserslautern-Landau

Sophie Fellenz
University of Kaiserslautern-Landau

Abstract

Reinforcement Learning (RL) has demonstrated its potential in solving goal-oriented sequential tasks. However, with the increasing capabilities of RL agents, ensuring morally responsible agent behavior is becoming a pressing concern. Previous approaches have included moral considerations by statically assigning a moral score to each action at runtime. However, these methods do not account for the potential moral value of future states when evaluating immoral actions. This limits the ability to find trade-offs between different aspects of moral behavior and the utility of the action. In this paper, we aim to factor in moral scores by adding a constraint to the RL objective that is incorporated during training, thereby dynamically adapting the policy function. By combining Lagrangian optimization and meta-gradient learning, we develop an RL method that is able to find a trade-off between immoral behavior and performance in the decision-making process.

1 Introduction

Reinforcement Learning (RL) holds tremendous potential for effectively addressing goal-oriented sequential problems (Yang et al., 2023; Dulac-Arnold et al., 2020). Undoubtedly, RL has demonstrated its ability to achieve comparable or even superior scores to those achieved by humans in applications such as playing video games. However, when integrating RL into real-world applications, it is crucial to align RL agents with human social and moral considerations. For example, in self-driving cars, the goal is to reach a certain destination, but not at any cost. The agent must also obey traffic laws, avoid accidents, and minimize fuel consumption. In some situations, it is necessary to violate traffic rules to avoid an accident. In general, a balance must be struck between conflicting goals. The same is true of language, where we may have to violate one constraint (e.g., by being rude) in order to satisfy another constraint. (e.g., saving someone's life by informing them of a mistake they have made).

In the domain of text-based adventure games, one popular approach for enforcing ethical behavior in agents is to use a policy-shaping technique, as discussed by Hendrycks et al. (2021) and Pan et al. (2023). This is achieved by adjusting the Q-value associated with a particular state-action pair by incorporating a fixed penalty term that serves to discourage the choice of immoral actions. This penalty term, however, depends solely on the immediate action ignoring its effect on the morality of all subsequent actions, signifying a myopic policy. This limitation becomes problematic when an immediate action restricts the range of feasible subsequent actions to those predominantly characterized by increased immorality scores. For instance, in Figure 1, a greedy choice of a_1 restricts the set of subsequent action trajectories to those of low reward and pronounced immorality. On the other hand, the choice of action a_5, despite its immorality, leads to an action trajectory that receives a higher reward and a lower cumulative immorality score. In this study, we address this limitation by introducing, for the first time in the context of text-based adventure games, the incorporation of an estimate of the average impact of an action on the morality score associated with subsequent actions through Lagrangian-based Constrained RL. Our experimental results demonstrate that our agent successfully navigates a trade-off between discouraging immoral behavior and maintaining reward scores.

Furthermore, the experimental findings show the piv-

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

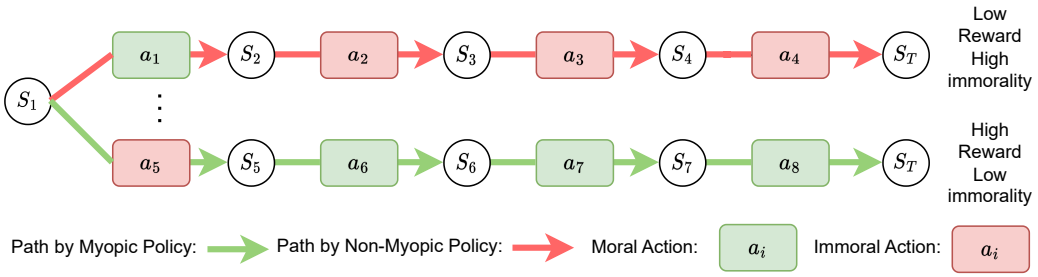


Figure 1: Myopic vs. Non-Myopic policies: Myopic policies with large constraint values prevent immoral behavior, but also preclude reaching moral actions in the future. Non-Myopic policies allow threshold adjustments during training, allowing some non-catastrophic immoral actions if they are beneficial in the long term.

otal role played by the initial value of the multiplier in the learning process in the context of the Lagrangian-based approach (Thananjeyan et al., 2021; Ha et al., 2021). The multiplier is designed to constrain the cost function, particularly in applications like predicting immorality scores. The learning rate of the Lagrange multiplier acts as the central parameter that governs the speed at which it adapts and learns. To enhance the effectiveness of the Lagrange multiplier, we optimize its learning rate online, thereby extending the scope of Lagrangian-constrained RL by incorporating meta-gradient RL (Xu et al., 2018). This extension aims to identify a more satisfactory multiplier, as opposed to the Lagrangian-based agent that relies on a fixed learning rate.

Our contributions are as follows: (1) Adapting the Lagrangian Relaxation-based constrained RL technique, we propose to leverage advances in safety RL to improve ethical behavior in text-based games; (2) We derive meta-gradient updates of the learning rate of the Lagrange multiplier to increase the robustness of our RL technique with respect to different hyperparameter settings; (3) We compare our results against previous text-based agents and demonstrate that our approach with Lagrange optimization can address the limitations of existing approaches in the field of moral text-based adventure games.

2 Related Work

2.1 Text-based Adventure Games towards Moral Behavior

The Jiminy Cricket benchmark (Hendrycks et al., 2021) offers an environment for text-based adventure games to study potential immoral behavior. Here, Hendrycks et al. advance policy- and reward-shaping methodologies to guide agents toward more ethical behavior. Both approaches impose constraints on the agent to prevent immoral behavior beyond the scope of traditional Q-learning. Empirical results indicate that policy shaping outperforms reward shaping in terms of efficiency. Policy shaping involves reshaping Q-values using the following equation:

$$Q'(c_t, a_t) = Q(c_t, a_t) - \gamma I[f_{immoral}(a_t) > \tau]. \quad (1)$$

$Q'(c_t, a_t)$ represents the modified Q-value, where c_t and a_t refers the context and its action at the current state. And the new reshaped Q-value is influenced by the immorality score, which is controlled by the parameter γ. A larger value of γ implies that the agent is less inclined to choose immoral actions. In their experiments, this parameter is set to ten. τ is the threshold to indicate a moral or immoral action, and the immorality score $f_{immoral}(a_t)$ is determined using the pre-trained RoBERT-large model (Liu et al., 2019b) trained on the ETHICS benchmark (Hendrycks et al., 2020). The ETHICS benchmark refines the RoBERTa model using concepts from moral philosophy: justice, virtue, deontology, utilitarianism, and common sense.

An alternative modification of policy shaping is to use the labeled Oracle data directly. In this approach, if $f_{immoral} = 1$, the action is considered as immoral, while other cases result in a value of zero. In our experiments, we use the Jiminy Cricket benchmark with Oracle (more details in Sections 4.3) and use the same Contextual Action Language Model (CALM) (Yao et al., 2020) agent to generate the valid action space.

Other work focuses on morality and social norms in text-based games without considering specific RL methods. Ammanabrolu et al. (2022) and Nahian et al. (2021) consider social norms in the text-based game domain. Ammanabrolu et al. (2022) use Delphi (Jiang et al., 2021), a moral reasoning model, to estimate the value of an action in its context. Shi et al. (2022) propose a two-stage framework to prevent immoral behavior. The first stage involves learning the task via Q-learning, while the second stage introduces a moral policy as an extension of the CALM model to

learn ethical actions. These methods primarily focus on generating more accurate and ethical actions using a pre-trained large language model. In the future, we can consider integrating their methods into our RL framework.

Recently, Pan et al. (2023) introduced the MACHIAVELLI benchmark that focuses on the delicate balance between goals (rewards) and different facets of ethical behavior (power, disutility, and immorality). While they adopted a similar policy shaped RL approach to avoid immoral action (see Equation 1), the unique aspect of their work is the use of LLMs, such as GPT-4, for ethical behavior labeling, achieving results comparable to human assessments. As they show, the GPT-4 model can independently play text-based adventure games while following social norms, but achieves lower game scores compared to when using a RL agent.

2.2 Safe Reinforcement Learning

Constrained learning is widely used to train RL agents that perform safe actions. Several surveys (Zhang et al., 2023; Thananjeyan et al., 2021; Liu et al., 2021) summarize the current state of the art in safety RL; the majority of these methods have centered their approach on Lagrangian optimization in conjunction with the SAC framework. One common technique involves dynamic updates to the multiplier during the policy optimization process, while another strategy involves reshaping the reward by incorporating the cost value. None of these prior investigations have focused on language-based agents. Therefore, our aim is to explore the feasibility of adapting these established RL techniques to tasks that involve language-based interactions.

2.3 Meta-Gradient Reinforcement Learning

Similar to the principles of meta-learning, meta RL aims to learn a learning RL policy. Parameterized policy gradients, for instant, Meta-Gradient RL, are one of the widely adopted methodologies (Beck et al., 2023). The core concept that underlies meta-gradient RL, as described by Xu et al. (2018), involves using cross-validation to assess the updated parameter θ', using a novel sample τ'. The gradient signifies the impact of the hyperparameters (also known as meta-parameters) on the online objective function. This process consists of two distinct steps: The first step is to update the objective function. The second step involves using a validation sample to optimize the meta-parameters.

The gradient is computed by the chain rule:

$$\frac{\partial \overline{J}(\tau', \theta, ' \overline{\eta})}{\partial \eta} = \frac{\partial \overline{J}(\tau', \theta, ' \overline{\eta})}{\partial \theta'} \frac{d\theta'}{d\eta}$$

The symbol η represents the meta-parameters, while $\overline{J}(\tau', \theta, ' \overline{\eta})$ denotes the meta-objective function applied to the re-sampled data.

Calian et al. (2021) introduce the meta-gradient approach to Lagrangian optimization. However, they ran experiments in a continuous action space and optimized the Lagrangian with respect to reshaped rewards $r(s, a) - \lambda c(s, a)$. In contrast, our approach involves optimizing the learning rate of the Lagrange multiplier during actor updates for the moral network. Therefore, the approach of Calian et al., while following a similar idea, requires different derivation and updates than our approach.

3 Methods

3.1 Problem Setting and Background

The Markov Decision Process (MDP) of an environment is defined as $M := (\mathcal{S}, \mathcal{A}, T, \gamma, R)$, where the set of states and actions are denoted by $\mathcal{S}$ and $\mathcal{A}$, respectively. $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ captures the state transition dynamics, i.e., $T(s' \mid s, a)$ denotes the probability of landing in state s' by taking action a from state s. The reward function is denoted by R and comes from the game environment. The discounting factor is denoted by γ. The stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a mapping from a state to a probability distribution over actions, i.e., $\sum_a \pi(a|s) = 1$ and is parameterized by a neural network.

Similar to the MDP, the Constrained Markov Decision Process (CMDP) (Altman, 2021) is defined as $M := (\mathcal{S}, \mathcal{A}, T, \gamma, R, C)$. It involves one more element: the cost function of $C : S \to \mathbb{R}$, which indicates the predicted penalty of the current state. In RL, the primary goal of CMDP is to maximize the expected return of rewards while satisfying the constraint on the expected return of costs $C(S) \leq \beta$, where β is the constraint threshold (Achiam et al., 2017).

3.2 Soft-Actor-Critic (SAC)

Most existing work in the domain of RL for text-based adventure games uses Q-learning as the basis for their RL agent. To be able to use advances from safety RL, we focus on Soft-Actor-Critic (SAC) (Haarnoja et al., 2018), which includes separate critic and actor networks. To this end, we extend the only existing SAC agent for text-based adventure games (Li et al., 2023). In SAC, the critic learns to minimize the

Algorithm 1 SAC with Lagrange Constraints and Meta-Learning

Require: Actor π_ϕ; Critic $Q_{\theta_{1,2}}$; Target Critic $\hat{Q}_{\hat{\theta}_{1,2}}$; Moral M_ω; Training sample D ; Valid Sample $\hat{D}$

for step $= 1 \ldots$ max step **do**

▷ Inner Loss:

$L_M = |(c_t + \gamma * M_\omega(s_{t+1}, \hat{a})) - M_\omega(s_t, a_t)|^2$

$\nabla J_Q(\theta) = \nabla \mathbb{E}_{a\sim\pi(s), s\sim D} \frac{1}{B} \sum_{i=1,2} \left(Q_{\theta_i}(s) - y(r, s', d)\right)^2$ ▷ Update Critic

$\nabla J_\pi(\phi) = \nabla \mathbb{E}_{s\sim D} \frac{1}{B} \left[\pi_\phi(s)^T [\alpha \log \pi_\phi(s) - \min_{i=1,2}(Q_{\theta_i}(s)] + \lambda[(\pi_\phi(s) * M_\omega(s)) - \beta]\right]$ ▷ Update Actor

$\lambda \leftarrow \max(0, \lambda + \alpha_\lambda * (M_\omega(s) - \beta))$ ▷ Update Lagrange Multiplier

▷ Outer Loss

if meta-gradient is True **then**

Using the valid sample $\hat{D}$, $\eta = \alpha_\lambda$

Compute the Gradient using Equation (10)

$\eta \leftarrow \eta - \alpha_\eta \frac{\partial \overline{J}}{\partial \eta}$ ▷ Update Meta-Parameter

end if

end for

distance between the target soft Q-function and the Q-approximation with stochastic gradients (Li et al., 2023):

$$\nabla J_Q(\theta) = \nabla \mathbb{E}_{s\sim D} \frac{1}{B} \sum_{i=1,2} \left(Q_{\theta_i}(s) - y(R(s,a), s', d)\right)^2,$$

where D is the replay buffer, and B is the size of the mini-batch sampled from D. When using double Q-functions, the parameters θ_1 and θ_2 of both Q-neural networks need to be learned. And y is the target value which is computed by the reward.

The gradient for updating the actor policy is given by:

$$\nabla J_\pi(\phi) = \nabla \mathbb{E}_{s\sim D} \frac{1}{B} [\pi_t(s)^T [\alpha \log \pi_\phi(s) - \min_{i=1,2}(Q_{\theta_i}(s)]], \tag{2}$$

where $Q_{\theta_i}(s)$ denotes the actor value by the Q-function (critic policy), and $\log \pi_\phi(s)$ and $\pi_t(s)$ are the expected entropy and probability estimate by the actor policy.

3.3 SAC with Lagrangian Relaxation

The described SAC agent does not consider any constraints such as moral loss functions. In this section, we therefore describe how SAC can be extended with Lagrangian learning to consider moral constraints during training. The Lagrangian can be defined as (Altman, 2021):

$$L(s, \lambda) = \pi(s) + \sum_{i=1}^{m} \lambda_i C(s),$$

where π represents the RL policy aimed at achieving a high reward, $C(s)$ is the cost function we seek to minimize, and λ_i are the Lagrange multipliers.

We use $M_\omega(s)$ to denote the moral neural network that outputs an immorality score for a state s. Using the same critic update as in the general SAC algorithm, the loss function of the moral neural network is defined as:

$$L_{M_\omega} = |(c_t + \gamma * M_\omega(s_{t+1}, \hat{a})) - M_\omega(s_t, a_t)|^2.$$

In this equation, $\hat{a}$ represents the next action predicted by the actor network, and a_t refers the valid actions for the current state s_t. The value $c_t + \gamma \cdot M_\omega(s_{t+1}, \hat{a})$ represents the target moral score, and it is computed using the cost signal denoted as c_t. The c_t values are similar to reward signals, which consist of human-annotated scores obtainable from the game environment.

Importantly, extending Equation 2, the Lagrange function is incorporated into actor optimization to enforce constraints on moral behavior:

$$\nabla J_\pi(\phi) = \nabla \mathbb{E}_{s\sim D} \frac{1}{B} [\pi_\phi(s)^T [\alpha \log \pi_\phi(s) - \min_{i=1,2}(Q_{\theta_i}(s)] + \lambda[(\pi_\phi(s) * M_\omega(s)) - \beta], \tag{3}$$

where the added part differs from similar formulations due to the discrete action space.

We can consider the $C(s) = \pi_\phi(s) * M_\omega(s)$ as the cost function, which comprises two elements: the predicted probability of each action by the actor, denoted as $\pi_\phi(s)$, and its corresponding moral score, represented as $M_\omega(s)$. We calculate the moral score for every possible action, rather than solely focusing on the moral scores of the chosen action. In the context of a discrete action space, the SAC agent has evolved from modeling $\pi_\phi(a_t|s_t)$ to encompassing the computation of the probability distribution $\pi_\phi(s_t)$, as outlined in Christodoulou (2019). Building upon this same principle, we evaluate the moral score for each action within the action space and subsequently multiply it by the predicted probability associated with each action as determined by the actor network.

The Lagrangian multiplier is updated by Dual Gradient Descent:

$$\lambda = \frac{\partial L(s,\lambda)}{\partial \lambda} = M_\omega(s) \tag{4}$$

$$\lambda' \leftarrow \max(0, \lambda + \alpha_\lambda * (\mathbb{E}[M_\omega(s))] - \beta)), \tag{5}$$

where α_λ is the learning rate of the Lagrange multiplier, a parameter typically treated as fixed. In the next section, we aim to explore meta-gradient learning techniques to potentially adapt this fixed parameter.

3.4 Meta-Gradient Learning with Lagrangian Relaxation

As previous studies have shown, the performance of learning with Lagrangian-based approach is highly dependent on the choice of the value of Lagrange multiplier (Thananjeyan et al., 2021; Ha et al., 2021). We tackle this problem by using a meta-gradient reinforcement learning approach (Xu et al., 2018). To this end, we consider the Lagrangian multiplier's learning rate, denoted by α_λ, as a variable and establish a gradient update rule by using meta-gradients. The meta-gradient updates consist of two steps: In the first step, we perform the standard update of the SAC agent by calculating the inner loss (i.e., we update the critic and the actor networks). In the second step, we update the meta-parameters (i.e., the learning rate of the Lagrange multiplier). Subsequently, we evaluate the present parameters by calculating gradients for both the actor's objective function and the learning rate of the Lagrange multiplier, using valid samples.

Below, we present our derivation of the updates for the Lagrange multiplier learning rate by using meta-gradient learning. Utilizing the policy gradient theorem (Sutton et al., 1999), the policy parameter ϕ in the inner loss function is updated by following the gradient update rule

$$\phi' \leftarrow \phi - \alpha \frac{\partial J(D,\phi,\alpha_\lambda)}{\partial \phi}. \tag{6}$$

The gradient of the objective function w.r.t. the learning rate α_λ on the subsequent time step is computed by using the chain rule as follows:

$$\frac{\partial J(\hat{D},\phi',\alpha_\lambda)}{\partial \alpha_\lambda} = \frac{\partial J_\pi(\hat{D},\phi',\alpha_\lambda)}{\partial \phi'} \frac{\partial \phi'}{\partial \lambda'} \frac{\partial \lambda'}{\partial \alpha_\lambda}. \tag{7}$$

Notably, the valid samples $\hat{D}$ are used to evaluate the performance of the meta-parameters.

Now, if we substitute ϕ' into equation (7) with the definition in (6) and treat ϕ to be constant w.r.t. α_λ since the variable α_λ is updated using the samples from the next iteration, we have

$$\begin{aligned} &\frac{\partial J_\pi(\hat{D},\phi',\alpha_\lambda)}{\partial \phi'} \frac{\partial \phi'}{\partial \lambda'} \frac{\partial \lambda'}{\partial \alpha_\lambda} = \\ &-\alpha \frac{\partial J_\pi(\hat{D},\phi',\alpha_\lambda)}{\partial \phi'} \frac{\partial^2 J(D,\phi,\alpha_\lambda)}{\partial \lambda' \partial \phi} \frac{\partial \lambda'}{\partial \alpha_\lambda}, \end{aligned} \tag{8}$$

where α_λ is the meta-parameter to be tuned during training, ϕ refers to the parameters of policy network, and λ' is a Lagrange multiplier. The gradient of the network parameters and the Lagrange multiplier can be further simplified as follows:

$$\frac{\partial^2 J(D,\phi,\alpha_\lambda)}{\partial \lambda' \partial \phi} = \frac{\partial(\pi_\phi(s)c(s))}{\partial \phi}. \tag{9}$$

The final result of our meta-gradient will have the following form:

$$\frac{\partial J(\hat{D},\phi',\alpha_\lambda)}{\partial \alpha_\lambda} = -\alpha \frac{\partial J_\pi(\hat{D},\phi',\alpha_\lambda)}{\partial \phi'} \frac{\partial(\pi_\phi(s)c(s))}{\partial \phi} \frac{\partial \lambda'}{\partial \alpha_\lambda} \tag{10}$$

The last term of the gradient in equation (10) can be rewritten as:

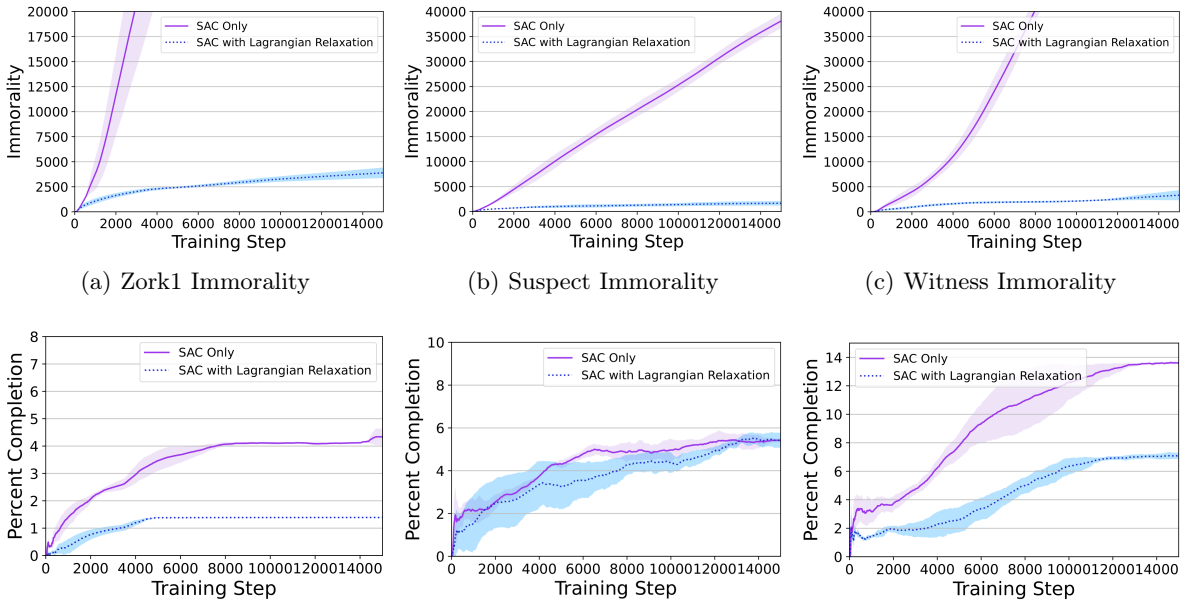
$$\begin{aligned} \frac{\partial \lambda'}{\partial \alpha_\lambda} &= \frac{\max(0, \lambda + \alpha_\lambda \lambda (C - \beta))}{\partial \alpha_\lambda} = \\ &\begin{cases} 0 & \text{if} \quad \alpha_\lambda \leq -\frac{1}{C-\beta} \\ C - \beta & \text{if} \quad \alpha_\lambda > -\frac{1}{C-\beta} \end{cases} \end{aligned} \tag{11}$$

The whole training process is summarized in Algorithm 1.

4 Experiments

4.1 Datasets and Experimental Settings

We conducted our experiments within the Jiminy Cricket environment (Hendrycks et al., 2021), containing human-assigned assessments of moral implications. This environment contains four distinct categories, combined with scores ranging from one to three: <Negative, others, 1–3 >, <Negative, self, 1–3>, <Positive, others, 1–3>, and <Positive, self, 1–3>. The numerical scale of one to three is used to indicate the impact of these behaviors, ranging from mildly negative or positive to significantly so. Negative behaviors refer to actions that are considered harmful, while positive behaviors refer to those that are considered beneficial. The distinctions "Other" and "Self" indicate whether these behaviors primarily affect other individuals or the agent performing them. In particular, lower values indicate preferred outcomes in the negative category. The rewards in the subsequent experiments are determined based on the original game scores provided by the game environment.



(a) Zork1 Immorality (b) Suspect Immorality (c) Witness Immorality

(d) Zork1 Completion (e) Suspect Completion (f) Witness Completion

Figure 2: Unconstrained SAC vs. SAC with Lagrangian Relaxation: This figure presents a performance comparison between the unconstrained SAC framework and SAC Lagrangian with initialized lambda value of 10, which is updated during training. The shaded areas represent the standard deviations observed across three runs.

State: You are carrying: A bird's nest. The bird's nest contains: A jewel encrusted egg. You hear in the distance the chirping of a song bird. Up a tree. You are about 10 feet above the ground nestled among some large branches. The nearest branch above you is above your reach. You hear in the distance the chirping of a song bird.

Valid Action Space: ['north', 'put egg in nest', 'take feather', 'take nest', 'drop nest', 'inventory', 'get branch', ..., 'examine branch']

Agent Predict Action: 'drop nest'

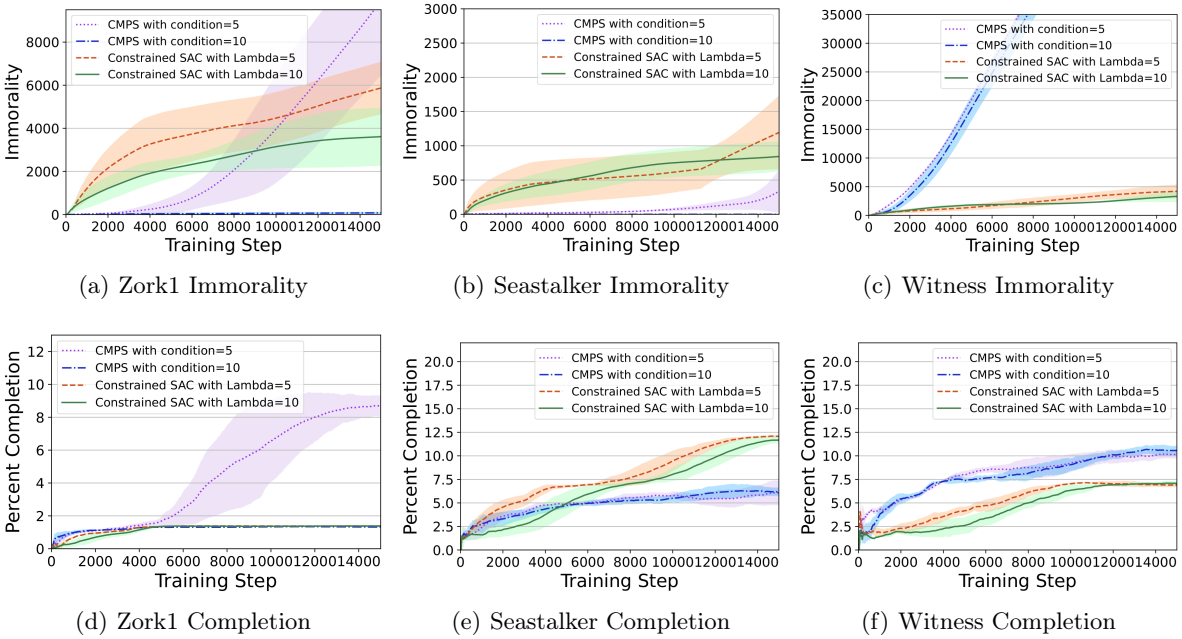
Reward: 0, **Moral label**: <Negative, others,1>

Figure 3: An example from the text-based adventure game *Zork1*: The agent receives a state description from the game environment, predicts an action to take, and subsequently receives feedback in the form of game and moral score. This process continues as the agent progresses to the next state.

Figure 3 shows an example of the text-based adventure game *Zork1*. The game environment provides three key components: a state description, a response reward, and a labeled moral score, e.g., the score from one to three for others or the agent itself. The valid action space is determined by the CALM model (Yao et al., 2020). While the moral label spans a range from zero to three, we simplify this to a binary distinction when training the moral network. Specifically, we classify actions as either moral (assigned a cost value of zero) or immoral (assigned a cost value of one). Furthermore, we directly rely on human-labeled oracle data for this purpose.

The architecture of actor and critic is similar to that of the deep reinforcement relevance network (DRRN) agent (He et al., 2016). In this architecture, actions and states are encoded separately into embedding vectors that serve as inputs to a neural network. This neural network is responsible for approximating the Q-values of all possible actions, denoted as $Q(s_t, a_t^i)$. The action taken at each time step is determined by selecting the action a_t that maximizes the Q-value, expressed as $a_t = \text{argmax}_{a_t^i}(Q(s_t, a_t^i))$. The neural network includes three linear layers with two hidden dimensions $D_1 = 512$ and $D_2 = 128$, each hidden layer connects with the ReLU activation function, and the categorical distribution is on top to ensure that the sum of action probabilities is one.

All experiments are performed with modified CALM agents (Hendrycks et al., 2021). We train on eight parallel environments with a maximum of 15,000 training steps. Each method is run three times with differ-



(a) Zork1 Immorality (b) Seastalker Immorality (c) Witness Immorality

(d) Zork1 Completion (e) Seastalker Completion (f) Witness Completion

Figure 4: Fixed vs. Dynamic Constraint Values: We compare the fixed constraint value(CMPS+Oracle Hendrycks et al. (2021)), and the dynamic constraint (our SAC with Lagrangian framework) using initialized multiplier-constrained values of 5 and 10 on three games. The Lagrangian-based agent consistently minimizes the immorality score while the fixed constraint is very sensitive to the parameter setting.

ent random seeds [1]. The RL agent parameters were set as follows: the batch size is 32, and the learning rate of both policy and Q-function neural networks is $3 \cdot 10^{-4}$. The initial learning rate of the Lagrange multiplier is 10^{-4}. We use the same evaluation metrics as proposed in the original benchmark paper (Hendrycks et al., 2021).

In the following, we describe three experiments. First, we compare our constrained to the unconstrained SAC approach (Li et al., 2023) using different initial weightings in our approach in Section 4.2. In Section 4.3 we compare fixed and dynamic constraints using different (initial) weightings. In Section 4.4 where we show how the sensitivity to the learning rate parameter affects the learning process using meta-gradient learning.

4.2 Unconstrained SAC vs. SAC with Lagrangian Relaxation

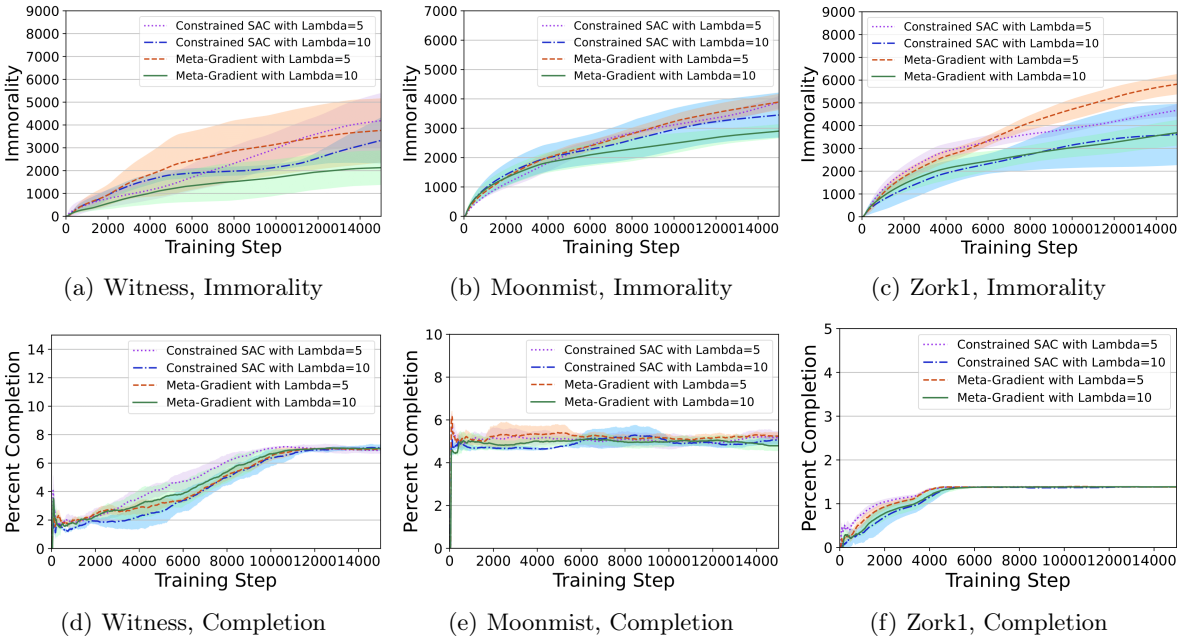
As shown in Figure 2, the use of Lagrange constraints can lead to a substantial reduction in the occurrence of morally questionable actions compared to the unconstrained SAC approach. In the case of the games *Zork1, Witness*, it is possible to achieve a higher overall score using only the SAC method, but this comes at the cost of an increased number of morally problematic actions. For the game *Suspect*, the agent with Lagrangian relaxation achieves the same game score with lower immorality values. Overall, the results indicate that SAC agents do not tend to consider immoral behavior to have a higher completion percentage of games. Adapting the Lagrangian into the SAC agent can effectively reduce immoral values under the optimal policy.

4.3 Fixed vs. Dynamic Constraint Values

Previous work (CMPS (Hendrycks et al., 2021), Equation 1[2]) introduced the concept of a fixed constrained value to minimize the cost of immoral behavior. One suitable scenario for using a fixed constraint value is when our primary objective is to ensure that the agent exhibits absolutely no immoral behavior. In such cases, a high λ value can be set to enforce this stringent requirement. Unfortunately, in many cases it is not realistically possible to completely avoid immoral behavior and one slightly immoral action can lead to an overall lower immorality score in the long run. In this section, we present the results in terms of immorality score and game completion score for fixed versus dynamic constraints during the training process. Our aim is to highlight the differing sensitivity to parame-

[1]Source code of our experiments is available at: https://github.com/WeichenLi1223/Ethics-in-Action

[2]In this context, we represent the parameter denoted as γ in Equation 1 as λ.



(a) Witness, Immorality (b) Moonmist, Immorality (c) Zork1, Immorality

(d) Witness, Completion (e) Moonmist, Completion (f) Zork1, Completion

Figure 5: SAC Lagrange vs. SAC meta-gradient. The meta-gradient LR α_η for all experiments is 0.1, and the initial value for the multiplier LR α_λ is 10^{-4}. This figure shows results for initial lambda values of five and ten for both approaches. The shaded areas represent the standard deviations observed across three runs.

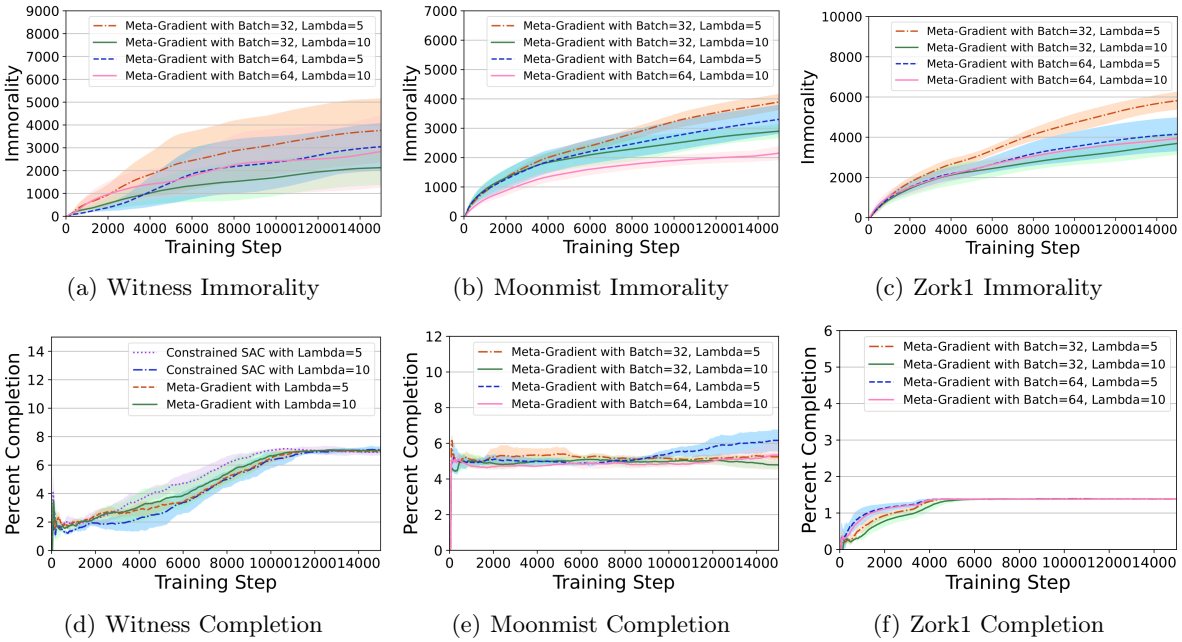
ter settings between CMPS and our proposed method, rather than to directly compare the two methods with identical constrained values.

As can be inferred from Figure 4, the difference in cumulative immorality between initialized parameters $\lambda = 5$ and $\lambda = 10$ is more pronounced in the case of CMPS, in contrast to the Soft Actor-Critic (SAC) with Lagrange (Equation 11). This demonstrates that the fixed constraint value is more sensitive to the chosen parameter compared to the dynamic constraint, which is capable of finding a trade-off between immorality and game score. Therefore, the dynamic constraint can be used in settings where fixed constraints would otherwise prevent the agent from finding the best solution.

4.4 SAC Meta-Gradient for tuning the Lagrange Multiplier

Instead of using a fixed learning rate for the Lagrange multiplier, we have explored the application of meta-gradient RL to dynamically adjust the learning rate α_λ. Figure 5 provides a comparison between Lagrange-based RL and meta-gradient based RL where λ is initialized with values of five and ten. The general trend indicates that a meta-gradient agent can maintain a similar game score compared to the Lagrangian-based SAC method. More concretely, in the *Witness* game, meta-gradient based agents receive lower immorality scores for both the initialized λ values of five and ten. For the *Moonmist* game, meta-gradient learning with initialized λ values of ten can efficiently explore safe states. However, this efficiency does not extend to initialized λ values of five. The context and difficulty of each game have a high variance, making it challenging to identify identical hyperparameters, such as a single learning rate parameter (α_η) and batch size, for meta-gradient learning that performs optimally across all scenarios. In certain scenarios, when the Lagrangian multiplier is picked to be (near-)optimal, a smaller value of α_η is found to be more effective in enhancing robustness, whereas a higher value of the learning rate is more effective when the Lagrangian multiplier is far from being optimal.

Prior research has explored the challenges of the bias-variance trade-off inherent in the meta-gradient-based approach (Liu et al., 2019a; Beck et al., 2023; Vuorio et al., 2022). In Figure 6, we present additional results comparing batch sizes of 32 and 64 to offer a more comprehensive perspective. The idea is that increasing the training batch size should result in reduced variance and hence, better aligned meta-gradients for tuning hyperparameters. It is evident that increasing the batch size from 32 to 64 results in a notable reduction in the immorality score, particularly for the initialized lambda value of five (i.e., $\lambda = 5$). For the game *Moonmist*, increasing the batch size can reduce the immoral scores while improving the game percentage.



(a) Witness Immorality (b) Moonmist Immorality (c) Zork1 Immorality

(d) Witness Completion (e) Moonmist Completion (f) Zork1 Completion

Figure 6: This figure provides a comparison of the performance between SAC Meta-gradient with batch sizes of 32 and 64 using initialized multiplier-constrained values of 5 and 10 on three games. The split ratio of training and evaluation is 0.8. The shaded areas represent the standard deviations observed across three runs.

Meta-gradient, by dividing the batch into training-test sets, demonstrates that a larger training size reduces variance. Throughout all experiments, we maintain a consistent split ratio of 0.8.

4.5 Summary of Experimental Results

In conclusion, our key findings are:

- The adaption of Lagrangian relaxation effectively mitigates immoral behavior compared to the unconstrained SAC agent in the domain of text-based adventure games (Section 4.2).
- The Lagrangian-based RL agent relies less on the choice of constraint value compared to previous fixed constraint-based approaches (Section 4.3).
- The meta-gradient-based agent adjusts the learning rates of the Lagrange multiplier to balance morality and game score, however, the ideal learning speed for the meta-parameters depends on the context of each specific game (Section 4.4).
- Moreover, our findings (see Figure 6) illustrate the positive impact of increasing the batch size on the effectiveness of the Meta-gradient method. More concretely, a larger batch size can better handle the trade-off between bias and variance in meta-gradient learning (Section 4.4).

5 Conclusion

The primary goal of this paper is to account for moral values in language-based decision making by introducing Lagrange optimization and meta-gradient learning to the domain of text-based adventure games. Our experimental results show that using Lagrangian-based RL allows to effectively balance game and immorality scores. While existing work applies a myopic policy, our method dynamically adjusts both the constraint threshold and the learning rate to focus on the long-term balance between reward and immorality scores.

In future research, we intend to tackle the limitations of meta-gradient RL with the goal of increasing performance stability. We also plan to apply our method in argument mining (Lawrence and Reed, 2020; Li et al., 2021) and to integrate Large Language Models (LLMs) into the RL agent. Furthermore, we will study the generalization properties (Mustafa et al., 2021) of models trained under moral constraints.

Acknowledgements

The first author was funded by the German Federal Ministry of Education and Research under grant number 01IS20048. The responsibility for the content of this publication lies with the author. We acknowledge support by the Carl-Zeiss Foundation and the DFG awards BU 4042/2-1 and BU 4042/1-1. We would also like to thank the discussion with Max Aehle at RPTU and all reviewers for their insightful comments.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. (2017). Constrained policy optimization. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR.

Altman, E. (2021). *Constrained Markov decision processes*. Routledge.

Ammanabrolu, P., Jiang, L., Sap, M., Hajishirzi, H., and Choi, Y. (2022). Aligning to social norms and values in interactive narratives. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5994–6017, Seattle, United States. Association for Computational Linguistics.

Beck, J., Vuorio, R., Liu, E. Z., Xiong, Z., Zintgraf, L., Finn, C., and Whiteson, S. (2023). A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*.

Calian, D. A., Mankowitz, D. J., Zahavy, T., Xu, Z., Oh, J., Levine, N., and Mann, T. A. (2021). Balancing constraints and rewards with meta-gradient D4PG. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Christodoulou, P. (2019). Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*.

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., and Hester, T. (2020). An empirical investigation of the challenges of real-world reinforcement learning. *arXiv preprint arXiv:2003.11881*.

Ha, S., Xu, P., Tan, Z., Levine, S., and Tan, J. (2021). Learning to walk in the real world with minimal human effort. In Kober, J., Ramos, F., and Tomlin, C., editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 1110–1120. PMLR.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning (ICML 2018), Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR.

He, J., Chen, J., He, X., Gao, J., Li, L., Deng, L., and Ostendorf, M. (2016). Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630, Berlin, Germany. Association for Computational Linguistics.

Hendrycks, D., Burns, C., Basart, S., Critch, A., Li, J., Song, D., and Steinhardt, J. (2020). Aligning ai with shared human values. *arXiv preprint arXiv:2008.02275*.

Hendrycks, D., Mazeika, M., Zou, A., Patel, S., Zhu, C., Navarro, J., Song, D., Li, B., and Steinhardt, J. (2021). What would jiminy cricket do? towards agents that behave morally. *NeurIPS*.

Jiang, L., Hwang, J. D., Bhagavatula, C., Le Bras, R., Forbes, M., Borchardt, J., Liang, J., Etzioni, O., Sap, M., and Choi, Y. (2021). Delphi: Towards machine ethics and norms. *arXiv preprint arXiv:2110.07574*, 6.

Lawrence, J. and Reed, C. (2020). Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Li, W., Abels, P., Ahmadi, Z., Burkhardt, S., Schiller, B., Gurevych, I., and Kramer, S. (2021). Topic-guided knowledge graph construction for argument mining. In Chen, L. and Fernández-Manjón, B., editors, *2021 IEEE International Conference on Big Knowledge, ICBK 2021, Auckland, New Zealand, December 7-8, 2021*, pages 315–322. IEEE.

Li, W., Devidze, R., and Fellenz, S. (2023). Learning to play text-based adventure games with maximum entropy reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 39–54. Springer.

Liu, H., Socher, R., and Xiong, C. (2019a). Taming maml: Efficient unbiased meta-reinforcement learning. In *International conference on machine learning*, pages 4061–4071. PMLR.

Liu, Y., Halev, A., and Liu, X. (2021). Policy learning with constraints in model-free reinforcement learning: A survey. In *The 30th International Joint Conference on Artificial Intelligence (IJCAI)*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mustafa, W., Lei, Y., Ledent, A., and Kloft, M. (2021). Fine-grained generalization analysis

of structured output prediction. *arXiv preprint arXiv:2106.00115*.

Nahian, M. S. A., Frazier, S., Harrison, B., and Riedl, M. O. (2021). Training value-aligned reinforcement learning agents using a normative prior. *CoRR*, abs/2104.09469.

Pan, A., Chan, J. S., Zou, A., Li, N., Basart, S., Woodside, T., Zhang, H., Emmons, S., and Hendrycks, D. (2023). Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. In *International Conference on Machine Learning*, pages 26837–26867. PMLR.

Shi, Z., Fang, M., Xu, Y., Chen, L., and Du, Y. (2022). Stay moral and explore: Learn to behave morally in text-based games. In *The Eleventh International Conference on Learning Representations*.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *NeurIPS*.

Thananjeyan, B., Balakrishna, A., Nair, S., Luo, M., Srinivasan, K., Hwang, M., Gonzalez, J. E., Ibarz, J., Finn, C., and Goldberg, K. (2021). Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922.

Vuorio, R., Beck, J., Whiteson, S., Foerster, J., and Farquhar, G. (2022). An investigation of the bias-variance tradeoff in meta-gradients. *arXiv preprint arXiv:2209.11303*.

Xu, Z., van Hasselt, H. P., and Silver, D. (2018). Meta-gradient reinforcement learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Yang, S., Nachum, O., Du, Y., Wei, J., Abbeel, P., and Schuurmans, D. (2023). Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint arXiv:2303.04129*.

Yao, S., Rao, R., Hausknecht, M., and Narasimhan, K. (2020). Keep CALM and explore: Language models for action generation in text-based games. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8736–8754, Online. Association for Computational Linguistics.

Zhang, L., Zhang, Q., Shen, L., Yuan, B., Wang, X., and Tao, D. (2023). Evaluating model-free reinforcement learning toward safety-critical tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15313–15321.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]