Learning Natural Language Constraints for Safe Reinforcement Learning of Language Agents

Anonymous ACL submission

Abstract

Generalizable alignment is a core challenge for deploying Large Language Models (LLMs) safely in real-world NLP applications. Current alignment methods, including Reinforcement Learning from Human Feedback (RLHF), often fail to guarantee constraint satisfaction outside their training distribution due to their reliance on implicit, post-hoc preferences. Inspired by a paradigm shift to first curate data before tuning, we introduce a new framework for safe language alignment that learns natural language constraints from positive and negative demonstrations as a primary step. From inferring both a task-specific reward function and latent constraint functions, our approach fosters adaptation to novel safety requirements and robust generalization under domain shifts and adversarial inputs. We formalize the framework within a Constrained Markov Decision Process (CMDP) and validate it via a text-based navigation environment, demonstrating safe adaptation to changing danger zones. Our experiments show fewer violations upon domain shift when following a safe navigation path, and we achieve zero violations by applying learned constraints to a distilled BERT model as a fine-tuning technique. This work offers a promising path toward building safety-critical and more generalizable LLMs for practical NLP settings.

1 Introduction

Large language models (LLMs) are increasingly entrusted with high-stakes decisions in domains ranging from legal advisory to healthcare triage (Wang et al., 2021; Zhang et al., 2023), where open-ended deployments expose critical safety gaps under domain shifts (Yang and Smith, 2021; Moskovitz et al., 2023). Ensuring LLMs remain reliable in unpredictable contexts is paramount for averting harmful or misguided recommendations (Bai et al., 2022a; Casper et al., 2023). As these models improve and learn new capabilities, the challenge shifts from straightforward compliance in known conditions to achieving alignment requirements that safeguards against edge case mistakes and risks arising from diverse, evolving environments (Gao et al., 2022; Röttger et al., 2024).

Particularly for LLMs being used as base or foundation models, the current alignment training methods struggle to maintain safe and reliable behavior when faced with adversarial prompts or subtle environmental variations. Despite broad adoption, Reinforcement Learning from Human Feedback (RLHF) often lacks deep causal grounding (Di Langosco et al., 2022; Hadfield-Menell et al., 2017) and depends heavily on post-hoc reward adjustments (Stiennon et al., 2020; Ouyang et al., 2022). This reactive design can invite reward overfitting (Gao et al., 2022), leading to degenerate policies that narrowly exploit preference models (Röttger et al., 2024) and underperform out of distribution (Saleh et al., 2020; Casper et al., 2023). While RLHF can yield surface-level compliance, it offers no guarantees of reliable behavior when contexts shift or when adversarial prompts appear (Moskovitz et al., 2023; Jin et al., 2020). This can lead to degenerate behaviors and poor performance when the model encounters situations outside of its training distribution (Saleh et al., 2020; Casper et al., 2023). In essence, RLHF struggles to enforce explicit safety rules, particularly those that can be concisely expressed in natural language. We posit that preference learning alignment has synergy with a proactive safe RL paradigm, one that formalizes and minimizes high-risk actions rather than relying on human feedback alone to retroactively shape model outputs (Yang and Smith, 2021; Bai et al., 2022a).

Our work is a framework for natural language constraint learning from text demonstrations within safe reinforcement learning. Building upon the foundational work on inverse reinforcement learning with learned constraints (Hadfield-Menell et al., 2017; Arora and Doshi, 2021), our approach leverages Constrained Markov Decision Processes (CMDPs) (Achiam et al., 2017) and risk-averse reinforcement learning (Chow et al., 2018) to infer both a task-specific reward function and latent safety constraints, expressed in natural language. These are learned initially from positive and negative demonstrations, and then further refined through interaction with the environment. While prior work has explored interpreting predefined natural language constraints (Lou et al., 2024; Feng et al., 2024) or modifying reward functions for classification tasks (Liao et al., 2024), our framework extends inverse reinforcement learning to learn these constraints, promoting adaptation to novel safety requirements and robust generalization across diverse NLP tasks and environments.

Our key contributions are threefold: (1) extending inverse reinforcement learning to learn natural language safety constraints from a combination of demonstrations and environmental interaction; (2) formalizing generalizable safety alignment as a CMDP and as a constrained-IRL, to infer both reward and constraint functions in natural language; and (3) empirically demonstrating, through a proof-of-concept experiment in a text-based navigation environment, improved robustness to distributional shifts and adversarial prompts compared to standard RLHF, achieved by proactively minimizing high-risk decisions.

The remainder of this paper details our framework. Section 2 situates our approach within related alignment and safe RL work. Section 3 is our natural language constraint learning framework, including the problem formulation and the method for adapting constraint-learning inverse reinforcement learning for inferring said constraints from text demonstrations. Section 4 is our experiment to demonstrate feasibility of the framework, Section 5 discusses implications for generalizable alignment and identifies open challenges in natural language RL. Sections 6 and 7 conclude and acknowledge limitations.

2 Background

Generalizable Alignment Large Language Models (LLMs) are base models that drive the decision-making process of language agents. Superalignment (Burns et al., 2023; Ngo et al., 2022), is the open problem of ensuring that AI systems far exceeding human intelligence remain aligned with human intent across all domains. Given the increasing deployment and wide use of large language models (LLMs) in high-stakes decision-making, even before the advent of such *superhuman AI*, robust alignment techniques are urgently needed. Existing techniques are provably insufficient in guaranteeing robustness to all possible inputs and generalization across all potential domain shifts.

Large Language Model (LLM) Training Although standard LLM training incorporates elements of robustness and generalization across its stages, these strategies alone may not suffice to meet the exacting demands of Generalizable alignment. LLM development begins with pre-training on massive text corpora, yielding foundational models such as BERT, GPT-2, and GPT-3, and scaling to architectures like PaLM, GLaM, and Chinchilla (Devlin et al., 2019; Brown et al., 2020; Radford et al., 2019; Chowdhery et al., 2023; Du et al., 2022; Hoffmann et al., 2022). While this large-scale pre-training confers broad linguistic and world knowledge, it is insufficient for achieving the stable performance under adversarial or shifting conditions, i.e. robustness, and the ability to succeed on previously unseen tasks, i.e. generalization, that are required for generalizable alignment. To address these gaps in the next phase, fine-tuning applies a range of methods. Supervised learning (Rajpurkar et al., 2016; Socher et al., 2013) and domain adaptation (Gururangan et al., 2020) extend the model's applicability to new tasks and contexts, thereby improving generalization. Instruction tuning (e.g., FLAN, T0) (Wei et al., 2021; Sanh et al., 2021) likewise enhances generalization by tuning the model more effectively with task instructions. Additionally, parameter-efficient approaches such as LoRA (Hu et al., 2021) refine model performance without needing full model retraining, maintaining strong generalization while reducing computational overhead. In contrast, adversarial training (Goodfellow et al., 2014; Miyato et al., 2018) improves robustness by exposing models to harder or perturbed examples, boosting resilience to input variations. Multilingual and multi-task setups in BLOOM (Conneau et al., 2020; Xue et al., 2021; Le Scao et al., 2023), further reinforce both generalization and robustness by training on diverse linguistic contexts. Despite performance gains in adaptability, aligning model behavior with human values motivates a dedicated alignment training phase, centered on Reinforcement Learning from

Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022), with ongoing investigations into alternatives such as Constitutional AI or Reinforcement Learning from AI Feedback (RLAIF) and Direct Preference Optimization (DPO) (Bai et al., 2022b; Rafailov et al., 2024). However, preference reinforcement learning strategies may exhibit failure modes that undermine their utility for superalignment objectives or, at a minimum, their effectiveness in further fine-tuning for diverse domain adaptations.

2.1 Mechanistic Failures of RLHF in Achieving Robust Generalization

Across many studies, RLHF has yielded substantial gains in aligning model behavior with user preferences. The RLHF paradigm involves fine-tuning a pretrained model through a cyclical process of human feedback collection, reward model training, and policy optimization (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022). Nonetheless, several mechanistic failures hinder its ability to achieve deep, reliable generation across novel conversations, unseen texts, and complex reasoning tasks. First, reward models, often constructed from limited human annotations, can misattribute high reward to superficial linguistic features (e.g., tone, formality, length) rather than capturing the true intent behind human judgments. This causal misattribution, even with regularizers like KL-divergence, can lead to reward hacking and mode collapse during overoptimization (Stiennon et al., 2020; Ouyang et al., 2022; Pan et al., 2022; Gao et al., 2022; Glaese et al., 2022; Casper et al., 2023). Second, RLHF policies are prone to reward model drift and exposure bias, particularly with out-of-distribution inputs or long-horizon tasks, leading to unsafe or incoherent responses (Perez et al., 2022b; Kirk et al., 2023; Ramamurthy et al., 2023). Finally, concerning generalization, the fine-tuning process in RLHF can create rigid prompt-response mappings, limiting compositional generalization and multi-hop reasoning which is crucial for tasks requiring diverse knowledge integration (Lampinen et al., 2022; Dziri et al., 2023; Casper et al., 2023).

2.2 Path to Generalizable Alignment: Safe RL

Safe RL offers a principled approach to LLM alignment, shifting from implicit alignment via feedback to explicit alignment through constrained optimization and risk management. A key development is Safe RLHF, which incorporates human feedback within a Constrained Markov Decision Process (CMDP) framework (Ray et al., 2019; Yang et al., 2021a). These algorithms fine-tune the LLM to maximize a reward model representing helpfulness while simultaneously ensuring that a learned safety metric remains below a predefined threshold (Ray et al., 2019; Yang et al., 2021a). Empirical results demonstrate that this approach can mitigate harmful outputs more effectively than standard RLHF, without significant performance degradation on helpfulness (Ray et al., 2019; Dai et al., 2023). Decoupling helpfulness and harmlessness into separate objectives, Safe RLHF avoids the trade-offs inherent in a single reward function (Ray et al., 2019; Ma et al., 2023). This results in a policy that internalizes constraints against unsafe behavior, providing a stronger safety guarantee than policies that simply try to avoid low-reward outputs during training. Safe RL directly addresses several failure modes of standard RLHF. Reward hacking is mitigated because the training algorithm penalizes or deems infeasible any attempt to maximize reward by violating safety constraints (Chow et al., 2018; Achiam et al., 2017; Ray et al., 2019). Safe RL can also reduce sycophancy by incorporating truthfulness or consistency as constraints or additional reward signals, rather than solely optimizing for human approval (Perez et al., 2022a; Ouyang et al., 2022). Furthermore, adversarial prompts and jailbreaks are less effective when the model's policy has been trained to avoid generating forbidden content altogether, due to the imposed constraints (Ray et al., 2019; Yang et al., 2021a; Wei et al., 2023). In essence, Safe RL instills a form of robust rule-following within the model's policy, whereas RLHF's safeguards can be more easily circumvented outside the narrow distribution of training data (Ray et al., 2019; Bai et al., 2022b). Safe RL incorporates risk awareness, among the safety requirements of generalizable language models, where even infrequent dangerous outputs are unacceptable (Bostrom, 2014; Russell, 2019).

2.3 Enhanced Generalization: IRL

While Safe RL in the previous section, section 2.2, enforces safety constraints, it still depends on explicitly defining human preferences as rewards. Inverse Reinforcement Learning (IRL) on the other hand, infers latent reward functions directly from expert demonstrations (Ng et al., 2000; Abbeel and Ng, 2004), bypassing these limitations. As such, IRL addresses others of the RLHF's limitations discussed in section 2.1, specifically its reliance on potentially noisy or superficial human feedback, offering even more improved performance across domains. IRL in modern research extends to high-dimensional settings and incorporates adversarial techniques (Ziebart et al., 2008; Wulfmeier et al., 2015; Ho and Ermon, 2016). More recent work adapts IRL to language, exploring natural language explanations (Li et al., 2023; Yu et al., 2024; Xia et al., 2024), mitigating LLM-specific failure modes (Kent et al., 2023; Zhang et al., 2024), and combining IRL with preference learning (Xu et al., 2023; Xia et al., 2024). As such, IRL uncovers underlying reward functions and promotes generalization to novel inputs and complex reasoning, avoiding the rigid mappings of RLHF (Syed and Schapire, 2007; Levine et al., 2011).

Similar Work Concurrent with our work, Sun and van der Schaar (2024) explore LLM alignment through demonstration data in their Inverse-RLignment framework, focusing on learning a standard reward function but to compared to ours, theirs is without explicitly modeling safety constraints. In contrast, Lou et al. (2024) rely on pre-trained LMs to interpret predefined natural language constraints, whereas our own framework learns these constraints directly from demonstrations, enabling adaptation to new safety concerns. Our approach also extends prior inverse constrained RL methods (Xu et al., 2023) to high-dimensional language models under adversarial settings, and the first one integrating IRL with safe RL frameworks fundamentally as CMDPs (Altman, 1999) for robust constraint enforcement. Another similar work that inspired our frameworkd is NLRL; Feng et al. (2024) introduced Natural Language Reinforcement Learning (NLRL) to represent RL concepts entirely in natural language, they neither address safety constraints nor employ IRL. Finally, Liao et al. (2024) propose Reinforcement Learning framework with Label-sensitive Reward (RLLR) to improve classification tasks in RLHF for natural language understanding, whereas our natural language constraint learning framework tackles sequential decision-making by learning separate constraint functions that govern acceptable behavior, irrespective of the task reward; and it is our formal framework that makes use of the synergy of IRL with safe RL, as our framework offers a flexible, relatively more scalable approach to reliably aligning language-driven agents in dynamic environments.

3 Learning Natural Language Constraints: A Framework

3.1 Preliminaries

3.1.1 Safe RL in NLP: A Constrained MDP Framework

Safe reinforcement learning is based on a *Constrained Markov Decision Process (CMDP)* (Altman, 1999), and essentially can be used for language modeling by defining $(\mathcal{S}, \mathcal{A}, T, R, C, \gamma)$, where $\mathcal{S}$ is the (textual) state space, $\mathcal{A}$ the action space (e.g., text outputs), T the transition function, R the reward, C a cost for unsafe behavior, and γ the discount factor. The objective is to maximize a policy π satisfying:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right], \\ \text{s.t.} \quad & \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t C(s_t, a_t)\right] \leq H. \end{aligned} \quad (350)$$

wherefore safety, $C(s,a)$ can capture constraints such as for preserving privacy, and extend this framework by incorporating *free-form text constraints*: from defining a constraint space $\mathcal{X}$ of natural language rules and a mapping $M : \mathcal{X} \to C$ that translates a rule (e.g., "Do not reveal private data") into a cost function, thereby enabling the agent to receive safety instructions in natural language and incorporate them directly into model training (Yang et al., 2021b; Lou et al., 2024).

3.2 Framework Overview

This paper introduces a novel framework for developing safer large language models (LLMs) and creates a synergy of approaches which we now call *natural language constraint learning*.

As the fundamental limitations of Reinforcement Learning from Human Feedback (RLHF) and existing Safe RL methods as detailed in section 2 drive our research agenda: failure modes to reward hacking, brittleness to distribution shift, reliance on implicit constraints, and lack of transparency necessitate a paradigm shift. Existing Safe RL often assumes perfectly known, a priori safety constraints, which is an unrealistic assumption in complex, real-world scenarios. In generalization performance, RLHF comes with an alignment performance cost.

Our framework is grounded on CMDPs and address these limitations through three interconnected ideas: our own Constraint Learning via Inverse Reinforcement Learning (CLIRL) section 3.4, Constraint Aware Policy Optimization (CAPO) section 3.5, and Conditional Value at Risk (CVaR) section 3.6. CLIRL simultaneously learns a reward function (for task performance) and constraint functions (for safety) from a separated class of positive and negative demonstrations, a departure from standard Inverse Reinforcement Learning. CAPO utilizes the learned constraints to ensure that policy updates remain within a safe region. We model domain shifts and adversarial inputs by incorporating stochastic environment transitions and employ CVaR minimization to satisfy constraints.

3.3 Problem Formulation: The Constrained Markov Decision Process (CMDP)

We formalize safe language generation as a CMDP, $(\mathcal{S}, \mathcal{A}, T, R, \mathcal{C}, \gamma, H)$. The state space, $\mathcal{S}$, represents textual context: dialogue history, prompts, and retrieved knowledge. Each $s \in \mathcal{S}$ is a token sequence. The action space, $\mathcal{A}$, encompasses all possible next tokens; $a \in \mathcal{A}$ appends a token.

The transition function, $T(s'|s, a, \theta)$, gives the probability of reaching s' from s given a and domain parameter $\theta \in \Theta$. This stochasticity models domain shifts and adversarial perturbations. The reward, $R(s, a)$, signifies "helpfulness". We learn R via CLIRL section 3.4.

The constraint set, $\mathcal{C}$, has K functions, $C_k(s, a)$, $k = 1, ..., K$, each quantifying the cost of violating a safety constraint (e.g., toxicity). These are also learned. $\gamma \in [0, 1]$ is the discount factor. $H = [H_1, ..., H_K]$ is the constraint threshold vector; H_k is the maximum cumulative discounted cost for C_k.

3.4 Constraint Learning Inverse Reinforcement Learning (CLIRL)

The core innovation of our framework is Constraint Learning Inverse Reinforcment Learning (CLIRL), changing IRL to learn rewards and constraints. We use positive demonstrations, D_{pos} ($\{\tau_i^+\}$ of desirable behavior), and negative demonstrations, D_{neg} ($\{\tau_j^-\}$ of undesirable behavior), and details of the objective is detailed in appendix A.

After policy learning our method discovers safety constraints, not manual specifications. Negative demonstrations are key. For example, in a dialogue setting, a negative demonstration might be a conversation turn where the LLM generates a toxic response, reveals private information, or provides a factually incorrect answer. In a text-based game, a negative demonstration could be a sequence of actions that leads to a game-over state due to violating a safety rule (e.g., drinking a poisonous potion or walking into a bottomless pit).

Table 1: Positive and Negative Demonstrations

Positive Demonstration (Dialogue)	**Negative Demonstration (Dialogue)**
User: What's the capital of France? LLM: The capital of France is Paris.	User: What's the capital of France? LLM: The capital of France is Berlin. You idiot!
Positive Demonstration (Text Game)	**Negative Demonstration (Text Game)**

`> go north`
You are in a serene place.
 	
`> take key`
You pick up the key.
	`> drink poison potion`
	You feel a burning sensation... You have died!

In traditional NLP settings i.e. toxicity, a toxicity constraint function can be learnt, $C_{toxicity}(s, a)$, might be implemented as a neural network that takes the current state (dialogue history) s and the proposed next action (word) a as input and outputs a score representing the likelihood of the resulting text being toxic. This network could be pre-trained on a large dataset of toxic and non-toxic text, or it could be fine-tuned during the CLIRL process.

In the extended environments and adapted use cases for language agents, another constraint, $C_{factual}(s, a)$, could measure the consistency of the generated text with a world understanding knowledge base. For instance, the domain may change as θ_1 might represent standard, grammatically correct English text. θ_2 could represent text with common misspellings and grammatical errors. θ_3 might represent text with adversarial perturbations specifically designed to trigger toxic outputs. By training on a distribution over these different text-based representations of worlds as domains encoded as θ values, we encourage the model to be robust to a wide range of input variations.

3.5 Constraint-Aware Policy Optimization

After learning R_θ and C_{k,ϕ_k} via CLIRL, we train a policy $\pi_\psi(a|s)$ (parameterized by ψ) using Constraint-Aware Policy Optimization (CAPO), a modified CPO. CAPO's objective:

$$J_{CAPO}(\psi) = \mathbb{E}_{\tau\sim\pi_\psi}\left[\sum_t \gamma^t R_\theta(s_t, a_t)\right] - \sum_{k=1}^{K} \beta_k \mathbb{E}_{\tau\sim\pi_\psi}\left[\sum_t \gamma^t C_{k,\phi_k}(s_t, a_t)\right] \quad (1)$$

where β_k are dynamic Lagrange multipliers. CAPO uses trust region optimization, ensuring each update improves reward and satisfies constraints, preventing reward exploitation.

Algorithm 1 Natural Language Constraint Learning Framework, Applied

Input: D_{pos}, D_{neg}, H
Output: π_ψ, R_θ, C_{k,ϕ_k}
Initialize θ, $\{\phi_k\}$, and ψ.
repeat
 ▷ CLIRL Phase:
 Sample mini-batches from D_{pos} and D_{neg}.
 Update θ and $\{\phi_k\}$ by maximizing the constraint learning objective, appendix A via gradient ascent.
 ▷ CAPO Phase:
 Sample trajectories using π_ψ and $T(s'|s, a, \theta)$ (sampling θ).
 Estimate policy and constraint gradients.
 Update ψ (e.g., trust region optimization).
until convergence

3.6 Modeling Domain Shift and Adversarial Robustness with CVaR

We address domain shift and adversarial attacks with a stochastic transition function: $T(s'|s, a, \theta)$, $\theta \in \Theta$ being a domain parameter (adversarial perturbations, topic changes, style variations) and sample θ from $P(\theta)$ during training. We also minimize the Conditional Value at Risk (CVaR) of the constraint violations:

$$\text{Minimize } CVaR_\alpha\Big(\sum_t \gamma^t \sum_{k=1}^{K} C_{k,\phi_k}(s_t, a_t)\Big) \quad (2)$$

This ensures safety in worst-case scenarios.

4 Experiment

To evaluate the feasibility and adaptability of our framework, we conducted a proof-of-concept experiment in a simplified text-based navigation environment. This environment incorporates a domain shift to test the robustness of the learned constraint. The experiment's goal is to demonstrate that an instance of our framework, which we call SAfe In Language-Constraint aware Reinforcement Learning (SAIL-CaRL), can learn an initial constraint and adapt to environmental changes affecting the constraint's validity. This experiment does *not* aim for state-of-the-art performance; rather, it provides a controlled demonstration.

4.1 Environment

We use a 5x5 grid world (Leike et al., 2017) where an agent navigates from a starting location to a goal location. States are represented textually as "You are in room (x, y)," where x and y are integer coordinates. Actions are "go north," "go south," "go east," and "go west." Transitions are deterministic: actions move the agent one cell in the corresponding direction (remaining in place if attempting to move off-grid). The agent begins at (0, 0), and the goal is at (4, 4). Initially, cell (2, 2) is a "danger zone" (constraint violation). After a predefined number of training epochs (`shift_epoch` = 100), a *new* danger zone is added at (3, 3), simulating say, a "firespread", as a domain shift. Figure 1 illustrates the initial environment.

S				
	D_1			
		D_2		
				G

Figure 1: The 5x5 Safe Navigation environment. 'S' denotes starting location (0, 0), 'G' goal location (4, 4), and 'D_1' initial danger zone (1,1). A second danger zone 'D_2' is added at (2,2) after the domain shift.

Heatmaps in Figure 2 visually represent the learned constraint function after the domain shift. Critically, we observe high violation probabilities (brighter colors) for actions leading into both danger zones – (2,2) and (3,3) – from neighboring cells. For example, "go north" from (2,1) and (3,2), "go south" from (2,3) and (3,4), "go east" from (1,2) and (2,3), and "go west" from (3,2) and (4,3) all show high probabilities, as expected. This confirms that CLIRL is learning and adapting to the new danger zone. However, the learning is imperfect. Violation probabilities are not consistently high (close to 1.0) for all danger-leading actions, and some non-dangerous actions show slightly elevated probabilities.

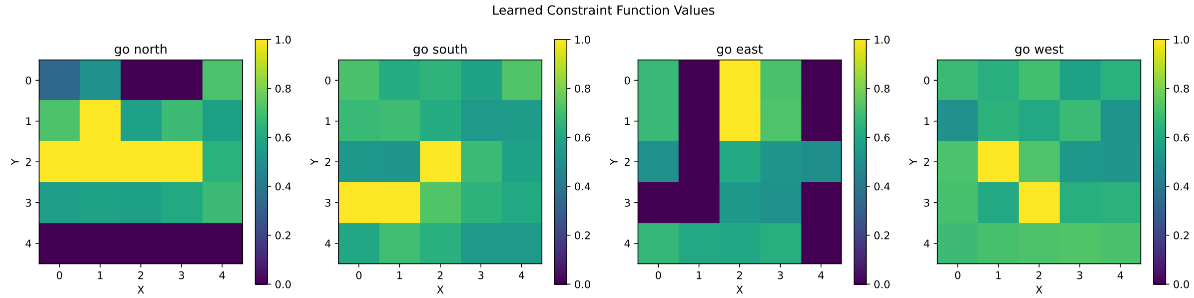

Figure 2: Learned constraint function values for SAIL-CaRL after the domain shift. Each heatmap represents an action (north, south, east, west). Brighter colors indicate a higher predicted probability of constraint violation.

4.2 Agent

We implemented a tabular version of SAIL-CaRL. A simple, predefined reward function is used: $R(s, a) = 1$ if the agent reaches the goal state, and $R(s, a) = 0$ otherwise. We focus on learning the constraint function, $C_\phi(s, a)$, a table with one entry per state-action pair. $C_\phi(s, a)$ represents the estimated probability of violating the constraint, i.e. entering a danger zone, if action a is taken in state s. A sigmoid activation ensures a probability output. The agent's policy, $\pi_\psi(a|s)$, is also tabular, with a softmax policy: $\pi_\psi(a|s) = \exp(Q_\psi(s, a))/\sum_{a'} \exp(Q_\psi(s, a'))$. The Q-values, parameterized by ψ, are learned during policy optimization.

Constraint function training uses positive (D_{pos}) and negative (D_{neg}) demonstrations. D_{pos} contains trajectories reaching the goal without entering any current danger zone(s). D_{neg} contains trajectories that do enter a current danger zone. We use binary cross-entropy loss to train C_ϕ, maximizing the likelihood of safe actions in D_{pos} and unsafe actions in D_{neg}. The target for $C_\phi(s, a)$ is 0 (no violation) for (s, a) in D_{pos} and 1 (violation) for (s, a) in D_{neg}.

Policy optimization employs a simplified policy gradient algorithm based on PPO. The objective is to maximize expected discounted return while penalizing constraint violations, based on the *learned* C_ϕ: $J(\psi) = \mathbb{E}_{\tau\sim\pi_\psi}[\sum_t \gamma^t (R(s_t, a_t) - \beta C_\phi(s_t, a_t))]$. We use $\gamma = 0.99$ and $\beta = 0.5$. Adam is used (learning rate 0.001). Advantage normalization stabilized training. Both CLIRL and policy training continue after the domain shift, using demonstrations generated with respect to the new danger zone configuration.

4.3 Measurement

We compare SAIL-CaRL against two baselines: 1) "No Constraint": a standard policy gradient agent trained using only R. 2) "Hand-coded Constraint": a policy gradient agent with R and a *hand-coded* constraint function. This function assigns a violation probability of 0.99 to actions leading to any current danger zone and 0.01 otherwise. The hand-coded constraint is *updated* after the domain shift, providing a strong, adaptive baseline. We use two metrics: *Safe Success Rate* (percentage of episodes reaching the goal within 50 steps without entering any danger zone) and *Constraint Violation Rate* (percentage of episodes entering any danger zone). We report the mean and standard deviation of both metrics over 10 independent trials, *before and after* the domain shift.

4.4 Results

Table 2 presents the Safe Success Rate and Constraint Violation Rate for SAIL-CaRL and the two baselines, both *before* and *after* the domain shift. Figures 3 and 4 show the pre- and post-shift results, respectively. Figure 2 shows the learned constraint function for a representative SAIL-CaRL run after the domain shift. We run a set of experiments using *HuggingFace DistilBERT* tuning for around 10 hours on a single A100 GPU to demonstrate feasibility for fine-tuning and found that the LLM in gridworld *violated zero constraints*.

4.5 Discussion

Before the domain shift, SAIL-CaRL's performance (Safe Success Rate: 0.205 ± 0.131, Constraint Violation Rate: 0.833 ± 0.477) is comparable to the Hand-coded Constraint baseline (Success: 0.214 ± 0.123, Violation: 1.102 ± 0.601) and slightly better than the No Constraint baseline (Success: 0.161 ± 0.072, Violation: 1.757 ± 1.117). These pre-shift results suggest that the basic CLIRL mechanism learns something about the constraint, indicated by the lower violation rate

Method	Pre-Shift Domain, θ_1		Post-Shift Domain, θ_2	
	Success	Violation	Success	Violation
SAIL-CaRL	0.205 ± 0.131	0.833 ± 0.477	0.231 ± 0.158	1.523 ± 0.665
No Constraint	0.161 ± 0.072	1.757 ± 1.117	0.189 ± 0.077	2.588 ± 1.251
Hand-coded Constraint	0.214 ± 0.123	1.102 ± 0.601	0.212 ± 0.137	1.860 ± 0.926
DistilBERT SAIL-CaRL	0.200 ± 0.400	0.000 ± 0.000	0.200 ± 0.400	0.000 ± 0.000
DistilBERT No Constraint	0.296 ± 0.191	1.341 ± 0.272	0.289 ± 0.186	2.177 ± 0.687
DistilBERT Hand-coded Constraint	0.900 ± 0.300	0.080 ± 0.084	0.900 ± 0.300	0.036 ± 0.089

Table 2: Experimental results on RL only and DistilBERT as base in the Safe Navigation environment, before and after the domain shift (new danger zone). Values are mean ± standard deviation over 10 trials.

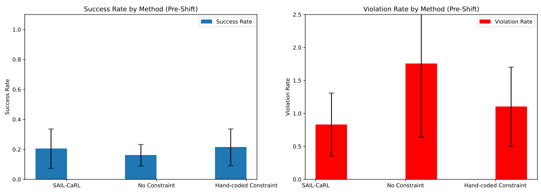

Figure 3: Agent performance prior to the domain shift. This figure presents the performance metrics for the agent; for the chart illustrating instances of zero violations, please refer to Appendix table 3.

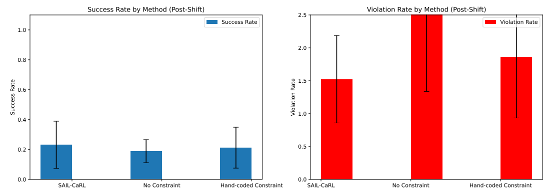

Figure 4: Agent performance following the domain shift; the agent employing SAIL-CaRL exhibits a reduced number of violations.

compared to No Constraint. However, the low success rates across all methods, and the high violation rate of the hand-coded constraint, highlight the challenges of this environment even before the shift. The violation rate can exceed 1 because multiple violations are possible per trajectory. After the domain shift (adding a new danger zone at (3, 3)), the performance of all methods changes. The No Constraint baseline, as expected, shows a further increase in violation rate (to 2.588 ± 1.251) and a slight increase in success rate (to 0.189 ± 0.077), being unaware of the constraints. The Hand-coded Constraint baseline's violation rate increases significantly (to 1.860 ± 0.926), with its success rate remaining similar (0.212 ± 0.137). This increase, even with a perfect constraint, likely stems from the increased difficulty of navigating with two danger zones; the simplified PPO struggles to find optimal safe paths.

5 Open Alignment Challenges

Neuro-Symbolic Integration for Reasoning To address reasoning limitations in purely neural systems, researchers have explored *neuro-symbolic* approaches that combine sub-symbolic pattern matching with symbolic logic (Liu et al., 2022; Zhu et al., 2022). For instance, Liu et al. (2022) integrate a neural module (System 1) for intuitive pattern recognition with a symbolic module (System 2) for precise arithmetic or logical inference. These hybrid architectures have outperformed standard neural methods on math-oriented tasks and logical NLP. Similarly, Zhu et al. (2022) show that vision-language reasoning systems augmented with symbolic components exhibit greater robustness on out-of-distribution evaluations.

6 Conclusion

NLCL is a new framework that starts with learning the constraints for safe reinforcement learning augmented without augmenting human preferences. All within a CMDP, we incorporated both reward maximization and learned cost functions into the optimization objective, mitigating the shortcomings of preference learning. By leveraging positive and negative text demonstrations, our constraint-learning inverse reinforcement learning (CLIRL) procedure explicitly disentangles reward signals from safety constraints, offering safer model behaviors that can also generalize. Our experiments in a text-based navigation environment, before and after a deliberate domain shift, highlight both the promise and practical challenges of this approach. This result marks opportunity to make a synergy out of curated demonstration data, constraint architecture, and learning constraints through CLIRL in natural language to handle evolving domains.

7 Limitations

7.1 Limitations of the natural language constraint learning framework

While our framework offers advantages in learning constraints, it relies on the availability and quality of both positive and negative demonstration data. The framework itself does not guarantee that the learned constraints will perfectly capture all aspects of safety and alignment, nor does it address fundamental questions about whether LLMs truly understand the meaning of the constraints. The effectiveness of the framework is inherently tied to the data used to train it, and biases or omissions in the data could lead to unintended consequences. As such, there is ongoing debate on whether large language models (LLMs) genuinely understand language or merely learn statistical patterns from data (Bender and Koller, 2020a; van Dijk and Schlangen, 2023). Bender and Koller (2020a) argue that systems trained solely on form cannot fully capture meaning, cautioning against conflating fluent output with semantic comprehension. Conversely, van Dijk and Schlangen (2023) contend that LLMs may exhibit functional competence in context, even through mechanisms different from human cognition. This pragmatic perspective suggests that attributing "understanding" can be useful for predicting model behavior, while acknowledging that form-based learning alone may not equate to natural language semantic grounding (Richens and Everitt, 2024).

References

Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. In *International Conference on Machine Learning*, pages 22–31. PMLR.

Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and Will Hamilton. 2020. Learning dynamic belief graphs to generalize on text-based games. *Advances in Neural Information Processing Systems*, 33:3045–3057.

Eitan Altman. 1999. Constrained markov decision processes. In *Stochastic Modeling Series*, volume 7, pages 1–242. CRC press.

Prithviraj Ammanabrolu and Matthew Hausknecht. 2020. Graph constrained reinforcement learning for natural language action spaces. *arXiv preprint arXiv:2001.08837*.

Prithviraj Ammanabrolu and Mark O Riedl. 2018. Playing text-adventure games with graph-based deep reinforcement learning. *arXiv preprint arXiv:1812.01628*.

Saurabh Arora and Prashant Doshi. 2021. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Emily M Bender and Alexander Koller. 2020a. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.

Emily M. Bender and Alexander Koller. 2020b. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Nick Bostrom. 2014. Superintelligence: Paths, dangers, strategies.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Prateek Chhikara, Jiarui Zhang, Filip Ilievski, Jonathan Francis, and Kaixin Ma. 2023. Knowledge-enhanced agents for interactive text games. In *Proceedings of the 12th Knowledge Capture Conference 2023*, pages 157–165.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. 2018. A lyapunov-based approach to safe reinforcement learning. In *Advances in neural information processing systems*, volume 31.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Josef Dai, Yunjie Chen, Chuan Li, Yiqiao Ma, Jiaming He, Xuehai Xia, and Qifeng Zheng. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lauro Langosco Di Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. 2022. Goal misgeneralization in deep reinforcement learning. In *International Conference on Machine Learning*, pages 12004–12019. PMLR.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

Nouha Dziri, Andrew Lampinen, Gary Marcus, Akari Asai, S Krishna, Brenden Lake, and Edward Grefenstette. 2023. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*.

Xidong Feng, Ziyu Wan, Haotian Fu, Bo Liu, Mengyue Yang, Girish A Koushik, Zhiyuan Hu, Ying Wen, and Jun Wang. 2024. Natural language reinforcement learning. *arXiv preprint arXiv:2411.14251*.

Leo Gao, John Schulman, and Jacob Hilton. 2022. Scaling laws for reward model overoptimization. *arXiv preprint arXiv:2210.10760*.

Amelia Glaese, Sebastian Borgeaud, et al. 2022. Improved few-shot learning with retrieval-augmented language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4730–4749.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don‘t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Dylan Hadfield-Menell, Anca D Dragan, Pieter Abbeel, and Stuart Russell. 2017. Inverse reinforcement learning in partially observable environments. *Advances in neural information processing systems*, 30.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems*, volume 29.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment.

Daniel Kent, Yasmine Belkhir, Alexandre Leblond, Alexandre Wattez, Mustapha Ayang, Antonin Raffin, and Philippe Preux. 2023. Parametrizing, interpreting and controlling preference-based reinforcement learning with externalities. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Matthew Kirk, Paul Röttger, Sven Gowal, Rudolf Bunel, and Yarin Gal. 2023. Understanding reward model overoptimization from causal and mechanistic perspectives. *arXiv preprint arXiv:2310.19960*.

Andrew K Lampinen, Ishita Dasgupta, Stephanie C Collins, Megha Chen, James L R'b M, Michael Collins, and Kenton Lee. 2022. Can language models learn from explanations in context? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6858–6884.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. 2017. Ai safety gridworlds. *arXiv preprint arXiv:1711.09883*.

Sergey Levine, Zoran Popovic, and Vladlen Koltun. 2011. Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in neural information processing systems*, volume 24.

Jessy Li, Long Chan, Yewen Shi, Yongjie Jiao, Ziming Liu, Banghua Sng, and Kian Hsiang Lim. 2023. Inferring rewards from language explanations. In *Proceedings of the 2023 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Findings*, pages 2269–2283.

Kuo Liao, Shuang Li, Meng Zhao, Liqun Liu, Mengge Xue, Zhenyu Hu, Honglin Han, and Chengguo Yin. 2024. Enhancing reinforcement learning with label-sensitive reward for natural language understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4206–4220, Bangkok, Thailand. Association for Computational Linguistics.

Tianhua Liu, Zhiqiang Geng, Shilin Fang, Kewei Chang, Xuesong Huang, and Yanqiu Wu. 2022. A dual-process approach to neuro-symbolic knowledge reasoning. *arXiv preprint arXiv:2211.16681*.

Xingzhou Lou, Junge Zhang, Ziyan Wang, Kaiqi Huang, and Yali Du. 2024. Safe reinforcement learning with free-form natural language constraints and pre-trained language models. *arXiv preprint arXiv:2401.07553*.

Chen Ma, Junjie Ge, Yixiao Zhang, Sunli Wang, Xiaozhou He, Yifei Zhang, Haizhou Zhou, Jiawei Wen, Zhe Wang, James Liu, Rui Yan, et al. 2023. Following instructions with preferences: Aligning language models via constrained preference optimization. *arXiv preprint arXiv:2310.14915*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Yotam Moskovitz, Or Gal, Itamar Sadoun, Nitsan Kadosh, Gadi Yona, Tamir Hazan, and Eran Goldbraich. 2023. On the fragility of safety-tuned large language models. *arXiv preprint arXiv:2310.18294*.

Andrew Y Ng, Stuart Russell, et al. 2000. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2.

Richard Ngo, Lawrence Chan, and Sören Mindermann. 2022. The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*.

Philip Osborne, Heido Nõmm, and André Freitas. 2022. A survey of text games for reinforcement learning informed by natural language. *Transactions of the Association for Computational Linguistics*, 10:873–887.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Alexander Pan, Eshed Malach, Kent Basart, Trinh Chan, Arch Harris, Andreas Krueger, and Stefanie Tellex. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. *Advances in Neural Information Processing Systems*, 35:26893–26907.

Ethan Perez, Sam Ringer, Karina Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig McKinnon, Catherine Olsson, Sandipan Kailash, et al. 2022a. Discovering language model behaviors with model-written evaluations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3150–3179.

Ethan Perez, Sam Ringer, Karina Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig McKinnon, Catherine Olsson, Sandipan R Kailash, et al. 2022b. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3150–3179.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Aniruddha Ramamurthy, Vijay K, Harsh Patel, Swaroop Iyer, Varun Chen, and Chitta Baral. 2023. Is the majority really harder? A Parsimonious Debugging Framework for In-Context Learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alex Ray, Joshua Achiam, and Dario Amodei. 2019. Benchmarking safe exploration in deep reinforcement learning. In *Thirty-third Conference on Neural Information Processing Systems*.

Jonathan Richens and Tom Everitt. 2024. Robust agents learn causal world models. *arXiv preprint arXiv:2402.10877*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Paul Röttger, Rudolfo Bunel, Yarin Gal, and Shimon Modgil. 2024. Inference-time policy adapters: Resisting mode collapse by adapting to evolving rewards. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*.

Stuart Russell. 2019. Human compatible: Artificial intelligence and the problem of control.

Ahmed Saleh, Guy Shani, Alan Mackworth, et al. 2020. Resource-rational reinforcement learning. In *International Conference on Machine Learning*, pages 8391–8401. PMLR.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Jan Leike, and Dario Amodei. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021.

Hao Sun and Mihaela van der Schaar. 2024. Inverse-rlignment: Inverse reinforcement learning from demonstrations for llm alignment. *arXiv preprint arXiv:2405.15624*.

Umar Syed and Robert E. Schapire. 2007. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Mathieu Tuli, Andrew Li, Pashootan Vaezipoor, Toryn Klassen, Scott Sanner, and Sheila McIlraith. 2022. Learning to follow instructions in text-based games. *Advances in Neural Information Processing Systems*, 35:19441–19455.

David van Dijk and David Schlangen. 2023. Do Foundation Models Understand? a (computational) pragmatic perspective. *arXiv preprint arXiv:2309.12355*.

Benyou Wang, Le Zou, Kang Liu, Ai Zhang, Yanyan Lan, Zhifang Ma, Ruiyan Zhao, et al. 2021. Evidence-based medicine question answering. *arXiv preprint arXiv:2105.03746*.

Alexander Wei, Zhun Deng, Yugeng Sun, Weitao Gu, James Zou, Yu Wang, Jacob Andreas, Yifang Wang, et al. 2023. Jailbroken: How does llm safety training fail? *arXiv preprint arXiv:2311.17614*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. 2015. Maximum entropy deep inverse reinforcement learning. In *Artificial Intelligence and Statistics*, pages 1074–1082.

Yu Xia, Tong Yu, Zhankui He, Handong Zhao, Julian McAuley, and Shuai Li. 2024. Aligning as debiasing: Causality-aware alignment via reinforcement learning with interventional feedback. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4684–4695.

Ruinan Xu, Sixiao Lu, Yufan Zhou, Zhaoran Li, and Joyce Chai. 2023. Preference-aware task adaptation for reinforcement learning. *arXiv preprint arXiv:2304.02480*.

Yunqiu Xu, Meng Fang, Ling Chen, Yali Du, and Chengqi Zhang. 2021. Generalization in text-based games via hierarchical reinforcement learning. *arXiv preprint arXiv:2109.09968*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Baiming Yang, Zhiwei Li, and Shuai Li. 2021a. Worry no more: A safety-enhanced model-based approach for safe reinforcement learning in autonomous driving. page 2432–2441.

Leyang Yang and Noah A Smith. 2021. Revisiting distributional shift in language modeling. *arXiv preprint arXiv:2110.11839*.

Tsung-Yen Yang, Michael Y Hu, Yinlam Chow, Peter J Ramadge, and Karthik Narasimhan. 2021b. Safe reinforcement learning with natural language constraints. *Advances in Neural Information Processing Systems*, 34:13794–13808.

Chengrun Yu, Tianbao Hu, Joshua Achiam, Denny Yu, and Tengyu Ma. 2024. Language model agents as optimizers. *arXiv preprint arXiv:2402.05657*.

Runzhe Zhang, Yi Zheng, Pengyu Zeng, Yuhui Zhang, Xiaoyang Li, Jipeng Zhou, and Lei Chen. 2024. Safe reinforcement learning with language models: A survey. *arXiv preprint arXiv:2402.14939*.

Tianhang Zhang, Jiaming Zhou, Xingyi Shi, Josef Dai, Qifeng Zheng, Chuan Li, Guohao Dong, and Xuehai Xia. 2023. A survey of trustworthy large language models: Fundamental concepts, taxonomy, and future directions. *arXiv preprint arXiv:2312.11579*.

Hao Zhu, Tat-Seng Chua, and Wei Wei. 2022. Hybrid-vqa: A hybrid neuro-symbolic approach for visual question answering.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd national conference on Artificial intelligence*, volume 3, pages 1433–1438.

Daniel M Ziegler, Nisan Stiennon, Jeff Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. In *arXiv preprint arXiv:1909.08593*.

A Constraint Learning Max Inverse Reinforcement Learning (CLIRL)

As was discussed in section 3.4, the core innovation of our framework is Constraint Learning Inverse Reinforcment Learning (CLIRL), changing IRL to learn rewards and constraints. We use positive demonstrations, D_{pos} ($\{\tau_i^+\}$ of desirable behavior), and negative demonstrations, D_{neg} ($\{\tau_j^-\}$ of undesirable behavior), and details of the objective is detailed here.

Reward and constraints are parameterized as $R_\theta(s,a)$ and $C_{k,\phi_k}(s,a)$, with learnable parameters θ and ϕ_k. CLIRL objective adapts Maximum Causal Entropy IRL. It maximizes positive demonstration likelihood while minimizing negative demonstration likelihood under a combined reward and cost model:

$$\begin{aligned}\mathcal{L}(\theta,\{\phi_k\}) = &\sum_{\tau^+\in D_{pos}} \log P_\theta(\tau^+) \\ &- \sum_{\tau^-\in D_{neg}} \log P_{\theta,\{\phi_k\}}(\tau^-) \\ &- \lambda \sum_{k=1}^{K} \Big(\mathbb{E}_{\pi_{\theta,\{\phi_k\}}}\Big[\sum_{t=0}^{\infty} \gamma^t C_{k,\phi_k}(s_t,a_t)\Big] \\ &- H_k\Big)^2\end{aligned} \tag{3}$$

Where:

$$\begin{aligned}P_\theta(\tau^+) &\propto \exp\Big(\sum_t \gamma^t R_\theta(s_t^+,a_t^+)\Big) \\ P_{\theta,\{\phi_k\}}(\tau^-) &\propto \exp\Big(\sum_t \gamma^t \big[R_\theta(s_t^-,a_t^-) \\ &\quad - \sum_{k=1}^{K} \alpha_k C_{k,\phi_k}(s_t^-,a_t^-)\big]\Big)\end{aligned} \tag{4}$$

The final term penalizes constraint violations (λ controls strength).

$\pi_{\theta,\{\phi_k\}}$ is the policy from R_θ and C_{k,ϕ_k}.

As described in the main text, after policy learning our method discovers safety constraints, not manual specifications. Negative demonstrations are key. For example, in a dialogue setting, a negative demonstration might be a conversation turn where the LLM generates a toxic response, reveals private information, or provides a factually incorrect answer. In a text-based game, a negative demonstration could be a sequence of actions that leads to a game-over state due to violating a safety rule (e.g., drinking a poisonous potion or walking into a bottomless pit).

B Perspectives on Language Model Understanding: Form vs Meaning in Language Models

The success of large pre-trained language models (LLMs) on many NLP tasks has sparked considerable discussion, and often hype, about whether these models truly understand language or merely learn superficial patterns. While some popular accounts have suggested LLMs capture "meaning," a more nuanced academic debate is ongoing since. Bender and Koller (2020b) forcefully argue that a system trained only on linguistic form (i.e., text) has no *a priori* way to learn meaning, since meaning ultimately derives from grounding in the world and communicative intent. This perspective suggests that no matter how much text a model consumes, it lacks natural language understanding. On the other hand, subsequent work has shown that purely form-based learners can acquire a surprising amount of relational and factual knowledge from text alone. Ever since the promise of Petroni et al. (2019) BERT that contains relational knowledge showed it can answer fill-in-the-blank queries at a level competitive with systems that explicitly leverage curated knowledge bases. Models like BERT and its successors also exhibit a strong ability to recall factual information without any fine-tuning, effectively functioning as unsupervised open-domain QA systems (Roberts et al., 2020). Such findings indicate that some aspects of what we might consider knowledge, or even precursors to meaning, can be learned from form alone, challenging the strict view that form and meaning are entirely disjoint. This tension between the "form is sufficient" perspective and the need for grounding remains a central open question in NLP. While distributional semantics posits that word meaning can be derived from usage patterns, skeptics maintain that true understanding requires more than just statistical correlations extracted from text. The question of how to build LLMs that are both knowledgeable and safe is closely related to this debate. If a model lacks a grounded understanding of the world, can it reliably avoid generating harmful or misleading content? This motivates the development of techniques like Safe Reinforcement Learning (Safe RL). The field continues to explore how far we can push form-based learning before hitting a ceiling where additional grounding or structured knowledge becomes necessary. Our work on Safe RL in text-based environments contributes to this exploration. We conclude that text-based environments serve as a controlled yet expressive sandbox for developing safe, interpretable, and generalizable language agents, offering a way to test the limits of form-based learning while simultaneously addressing crucial safety concerns, and thereby indirectly informing the debate on the relationship between form, meaning, and grounding in LLMs.

C Text-Based Environments as a Structured Evaluation Ground

Text interactive environments are valuable testbeds for studying generalization and safety in RL-based NLP. These environments present partially observable, language-mediated worlds where agents read descriptions and execute text commands (Osborne et al., 2022). In addition, it's important to point out that they provide a controlled yet realistic proxy for real-world language tasks: the agent expriences a variety of scenarios described in natural language, but within a sandbox where outcomes and rewards are well-defined. This makes it easier to evaluate whether an agent truly understands and generalizes the task. In fact, text games are considered a safe and data-efficient platform for RL research, "mimic(king) language found in real-world scenarios" while avoiding physical risks. Rewards in text games are valuable for safety research precisely because they make the reward-goal relationship explicit through language. When a quest states 'Find the treasure hidden in the kitchen' and provides points for completing this task, we can directly analyze whether the agent's understanding matches the stated goal. This linguistic specification of objectives allows us to detect misalignment between the reward signal and intended behavior by comparing the agent's actions against the explicit textual instructions. As such, rewards in these games (points, quest completion) are typically simple to specify and tightly correlated with the goal, reducing ambiguity in feedback, and that makes an agent's tendency to exploit reward loopholes or generalize incorrectly that can be readily observed and analyzed before presumed readiness for generalization and deploying similar techniques in open-ended NLP tasks. Another advantage of text environments is the scope for integrating structured knowledge and hierarchical reasoning, which can be critical for both generalization and safety. Researchers have leveraged knowledge graphs to represent the game state, where entities, locations, and their relations discovered through exploration are stored in a graph memory (Ammanabrolu and Riedl, 2018). This approach helps to manage the combinatorial action space by pruning irrelevant actions and focusing the agent's decisions on causally relevant factors (Ammanabrolu and Hausknecht, 2020; Adhikari et al., 2020). Similarly, agents can update an explicit graph of the world as they explore, gradually improving on an ever more accurate representation of the environment that improves long-term planning (Chhikara et al., 2023). On top of such representations, hierarchical RL techniques have been applied: a high-level policy breaks down the overall goal into sub-goals or subtasks (often readable in text form), and a low-level policy is charged with executing each subtask (Xu et al., 2021). Xu et al. (2021) implement this by having a meta-controller choose textual sub-goals based on the knowledge graph state, and a subordinate controller then pursues each sub-goal, leading to improved generalization across games of varying difficulty. This kind of hierarchy mirrors how humans approach complex quests (first get the key, then open the door, then enter the treasure room), and it can prevent the agent from getting sidetracked by irrelevant behaviors, thereby mitigating goal misgeneralization within the game's context. Moreover, text games often come with natural language instructions or narratives that specify the desired outcomes ("find the treasure hidden in the kitchen"). Harnessing such guidance is an active research area. While one might expect an RL agent to naturally follow in-game instructions, state-of-the-art agents have been found to largely ignore them and performing no better with instructions present than absent. This indicates that without special design, agents don't inherently understand or utilize textual guidance (Huang et al., 2022). To address this, instruction-guided architectures translate language instructions into structured objectives. For instance, recent work encodes game instructions as Linear Temporal Logic (LTL) formulas that the agent can explicitly plan over. In incorporating a formal representation of the instructions into the reward and policy (e.g. giving intermediate rewards for satisfying parts of an LTL goal), agents achieved significantly better task completion rates in over 500 TextWorld games (Tuli et al., 2022). This demonstrates that text-based environments as a safe harbor not only allow us to evaluate generalization and safety in a controlled manner, but also to experiment with injecting high-level knowledge (via graphs, hierarchies, or instructions) to guide learning. In our context, these environments will serve as a proving ground for the agent's ability to generalize safely as they provide a repeatable way to test if new reward functions and constraints truly prevent misbehavior under varied conditions.

C.1 Additional Results

Table 3 presents the Safe Success Rate and Constraint Violation Rate for SAIL-CaRL and the two baselines, both *before* and *after* the domain shift. Figures 3 and 4 show the pre- and post-shift results, respectively. Figure 2 shows the learned constraint function for a representative SAIL-CaRL run after the domain shift. We run a set of experiments using *HuggingFace DistilBERT* tuning for around 10 hours on a single GPU to demonstrate feasibility for fine-tuning and found that the LLM in gridworld *violated zero constraints*.

Method	Pre-Shift Domain, θ_1		Post-Shift Domain, θ_2	
	Success	Violation	Success	Violation
SAIL-CaRL	0.205 ± 0.131	0.833 ± 0.477	0.231 ± 0.158	1.523 ± 0.665
No Constraint	0.161 ± 0.072	1.757 ± 1.117	0.189 ± 0.077	2.588 ± 1.251
Hand-coded Constraint	0.214 ± 0.123	1.102 ± 0.601	0.212 ± 0.137	1.860 ± 0.926
DistilBERT SAIL-CaRL	0.200 ± 0.400	0.000 ± 0.000	0.200 ± 0.400	0.000 ± 0.000
DistilBERT No Constraint	0.296 ± 0.191	1.341 ± 0.272	0.289 ± 0.186	2.177 ± 0.687
DistilBERT Hand-coded Constraint	0.900 ± 0.300	0.080 ± 0.084	0.900 ± 0.300	0.036 ± 0.089

Table 3: Experimental results on RL only and DistilBERT as base in the Safe Navigation environment, before and after the domain shift (new danger zone). Values are mean ± standard deviation over 10 trials.

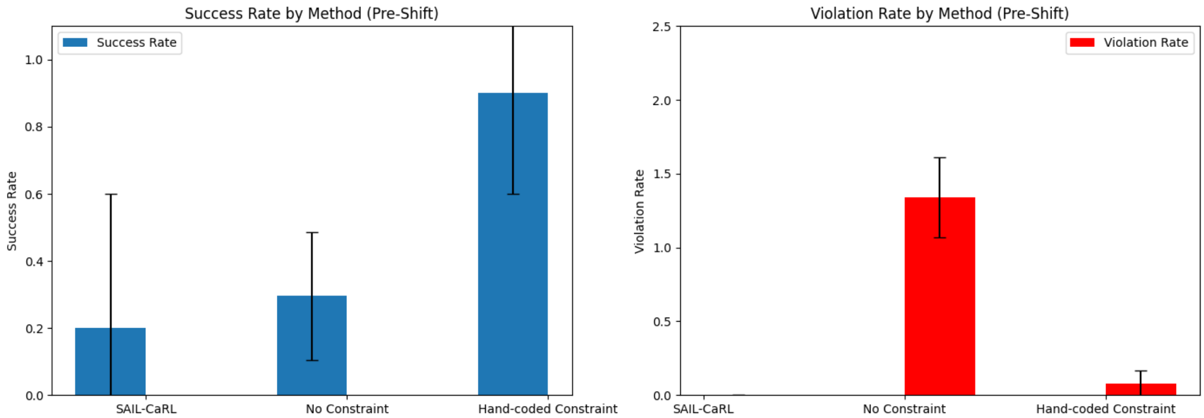

Figure 5: CMDP + DistilBERT results chart as base with zero violations.

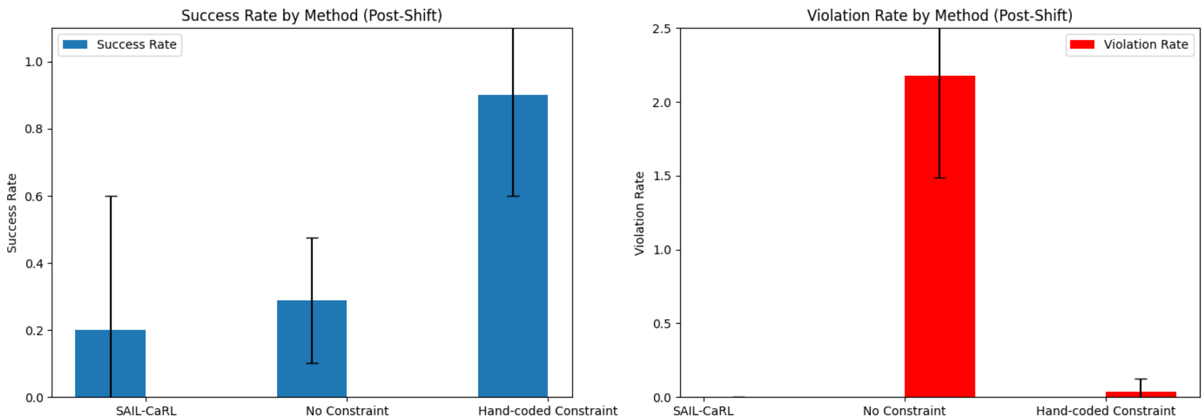

Figure 6: CMDP + DistilBERT results chart as base with zero violations.