Stability of Preference Alignment for Multi-Turn Control with LLM Policies

Andrew Silva, Pradyumna Tambwekar, Deepak Gopinath,
Jonathan DeCastro, Guy Rosman, Avinash Balachandran
Toyota Research Institute
Cambridge, MA
{andrew.silva, pradyumna.tambwekar, deepak.gopinath,
jonathan.decastro, guy.rosman, avinash.balachandran}@tri.global

Abstract

Large language models (LLMs) are increasingly deployed in multi-turn control settings, such as interface navigation and robot manipulation, where stability over long horizons is critical. In this work, we provide a study of preference alignment methods, including group-relative policy optimization (GRPO), direct preference optimization (DPO), contrastive preference optimization (CPO), and a GRPO variant with behavior cloning regularization, in two domains: a tokenized gridworld and a shared-control racing task that necessitates long-horizon planning and interaction. Rather than proposing a new algorithm, our goal is to analyze stability trade-offs and clarify when existing approaches succeed or fail. We show that (1) contrastive methods such as DPO and CPO risk policy degradation without valid negatives, (2) such methods struggle to recover multi-modal behaviors from a pre-trained initialization, and (3) adding behavior cloning regularization to GRPO improves robustness in some multi-turn settings. Together, our findings provide practical guidance for applying alignment techniques to long-horizon interactive policies and highlight open challenges for stable, preference-aware LLM control.

1 Introduction

The successes of LLMs have sparked growing interest in deploying LLMs as generalist agents in multi-turn domains, ranging from GUI interaction [58, 14, 57, 52] to robotic manipulation [33, 42, 5, 6, 15] and autonomous driving [8, 27]. Unlike single-turn text generation, these settings amplify stability and steerability concerns, as overconfident updates can erase useful behavior and contrastive learning can lead to mode collapse without informative negative samples. In this work, we focus on understanding how standard alignment objectives behave in long-horizon, multi-turn interactions, comparing GRPO [49], DPO [45], CPO [55], and an imitation-learning-augmented GRPO across two controlled testbeds. Rather than proposing a new algorithmic family, our goal is to surface practical stability trade-offs and give guidance for long-horizon alignment in interactive multi-turn tasks.

In single-turn language domains, the problem of alignment, that is, how to refine a base model for human-centric interaction, has received substantial attention. Techniques such as reinforcement learning from human feedback (RLHF) [10, 31], direct preference optimization (DPO) [45], or direct fine-tuning on preferences [15] have proven effective at shaping generations to satisfy user preferences. However, these methods have not yet made the leap to multi-turn settings, such as policy learning for control. Given the success of reinforcement learning (RL) applied to interactive domains such as robotics [2, 35, 39] and the interest in extending LLMs for multi-turn control tasks [33, 50], there is a clear gap in understanding how RLHF and DPO will extend to such settings. Existing LLM-control agents rely on large imitation datasets [11, 18, 24] for supervised fine-tuning

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Multi-Turn Interactions in Large Language Models.

or use LLMs for planning or code generation at test time [37, 54], sidestepping the challenge of direct policy learning. However, mapping from language to interactive *actions* remains a significant obstacle for LLMs, as multi-turn control policies can fall into unrecoverable failure states, discover bad local optima, or fail to generalize behaviors across domains and tasks owing to the covariate shift between limited training data and the diversity of testing environments. This gap is especially pressing in low-data or preference-sensitive domains such as assistive driving interventions [13], where pre-training may be impractical and preferences may vary from user to user. Our goal is to close this gap by aligning the inherent world knowledge of LLMs with the demands of multi-turn embodied control tasks.

To this end, we critically evaluate a suite of alignment techniques for LLM-based policies in multi-turn control settings. We first identify a connection between GRPO [49] and DPO [45], showing that DPO can be interpreted as a binarized, constrained sampling version of GRPO (§ 4.1). This connection motivates a principled modification to the GRPO objective, inspired by contrastive preference optimization (CPO) [55], which we find improves stability by anchoring the policy to behaviors that achieve high returns (§ 4.2). In this work, we focus on GRPO and DPO as they are popular, simple, and effective techniques widely applied for LLM alignment.

We then present empirical results across two domains: a synthetic gridworld environment (§ 5.1), and a high-fidelity driving simulator with multi-modal input data in which the LLM must learn to control a high speed vehicle in a race (§ 5.2). In both settings, we train LLMs to act directly from state representations (either expressed as raw text or consumed via multi-modal encoders that must be learned online) and compare alignment approaches. Across both tasks, we find that adding a behavior cloning regularization term can lead to improved stability and robustness in training, under certain conditions. However, if the model is not regularized to a reference policy or if the model repeatedly samples and imitates low-quality rollouts, behavior cloning regularization can lead to policy degradation and *worsen* performance.

Together, our findings offer the first systematic study of preference alignment for LLM policies in multi-turn control tasks. We show that while contrastive preference methods are promising, they face drawbacks that are magnified by multi-turn control tasks. In particular, we find that once a policy has collapsed or erased an important behavior, contrastive methods cannot recover, as errors and instabilities compound over successive updates to degenerating policies. Our results underscore the need for alignment methods that consider the sensitivity of multi-turn domains, and lay the groundwork for future research on preference-aware interaction with generalist models.

Our contributions are threefold:

- We clarify the connection between GRPO and DPO, highlighting their shared optimization structure and explaining when contrastive methods may become unstable.
- We systematically evaluate alignment methods across two multi-turn control domains, finding stability trade-offs between GRPO, DPO, CPO, and GRPO with behavior cloning.
- We identify conditions under which behavior cloning regularization stabilizes GRPO, providing practical insights for training long-horizon, multi-turn LLM policies.

2 Related Work

LLM Alignment. Researchers have extensively considered methods to constrain LLMs to generate text that is helpful, non-toxic, and better aligned with user expectations [3, 7, 12, 15, 29, 41]. While much research focuses on cleaner preference-datasets, there are three primary algorithmic directions for alignment: reinforcement learning from human feedback (RLHF) [41, 10, 49, 17, 4, 51], DPO [45, 43, 55, 59, 23, 38], or supervised fine-tuning on labeled preference data [15]. Though many frontier LLMs train under a combination of all three approaches [21, 34], the primary mechanisms for alignment remain RLHF and DPO, both of which have established failure modes if the model wanders out of distribution, is poorly pre-trained, or identifies sub-optimal reward hacking behaviors [56, 28, 36, 19, 22]. The utility of these approaches for learning *embodied* policies remains unclear.

Long-Horizon LLM Policies. As LLMs make continuous progress on general language understanding, researchers increasingly look to apply LLMs as policies for complex tasks. Researchers have applied LLMs to open-loop action generation via text output [27, 8] or single-turn interaction

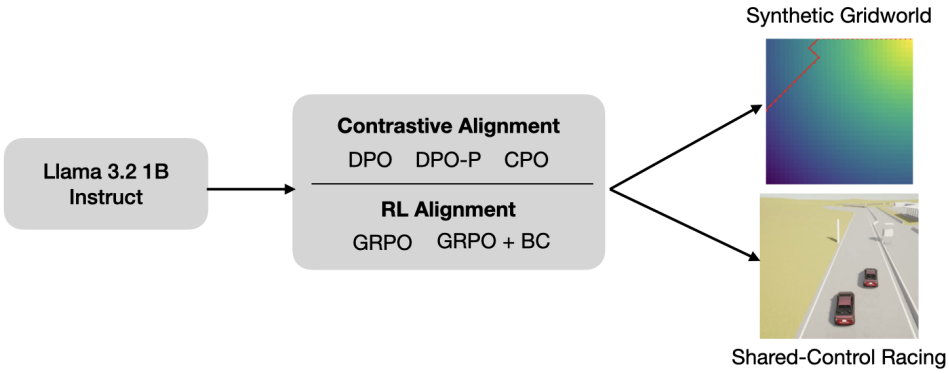


Figure 1: We compare alignment algorithms for learning LLM policies, examining contrastive and RL methods on two long-horizon tasks of gridworld navigation and shared-control racing.

via code generation and planning [37, 1]. Recent work has begun to extend this line of research to *multi-turn* interactions, such as interacting with web pages for shopping or search [58, 14, 57, 52]. Similarly, large vision-language-action models [33, 42, 5, 6, 15, 30] have demonstrated that LLMs can effectively learn to generate plausible behaviors after training on large datasets for robot learning [11]. In this work, we present an overview of alignment algorithms applied to LLM policies to refine pre-trained behaviors or discover new behaviors through online exploration when applied to long-horizon interaction tasks.

Behavior Cloning to Augment RL. Prior work has shown the benefits of combining behavior cloning with RL, either as a warm-start [9, 46], as a mechanism for labeling rollouts [47], for bonuses to exploration [40], or as a regularizing term for online preference learning [55, 20]. In this work, we draw inspiration from such prior works in considering how behavior cloning to on-policy explorations might stabilize LLM alignment.

3 Preliminaries

We begin by presenting background on Markov Decision Processes (MDPs) and policy gradient methods before drawing connections between GRPO and DPO. We present additional preliminaries on GRPO and DPO in Appendix C.

MDPs are defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $P(s'|s,a)$ is a transition distribution, $r(s,a)$ is a reward function, and $\gamma \in [0,1)$ is a discount factor. A policy $\pi(a|s)$ defines a distribution over actions conditioned on state. In traditional reinforcement learning, the goal is to learn a policy that maximizes expected cumulative reward: $\pi^* = \arg\max_\pi \mathbb{E}_{\tau\sim\pi}[R(\tau)]$ where $R(\tau) = \sum_{t=0}^{T} \gamma^t r(s_t, a_t)$, and τ is a completed sequence (trajectory). In many applications of preference learning for LLMs, states are defined as the entire token sequence up to any given timestep and actions are simply the next token to be added to the sequence. For the remainder of this work, we adopt this convention. Similarly, we assume that the state is always preceded by a language query, q, describing the goal. RLHF typically assumes that rewards are assigned to individual tokens as they are produced or to the entire sequence upon completion, and the discount factor $\gamma = 1$.

Policy Gradient Methods learn policies by estimating gradients of the expected return with respect to the policy parameters. The foundational approach is REINFORCE [53], which learns to maximize the likelihood of actions that lead to high returns, which has been further refined to maximizing the likelihood of actions that yield high advantage in any given state:

$$\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)] = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot A_t\right], \tag{1}$$

where A_t is the advantage of taking action a_t in state s_t. This gradient encourages actions that yield higher returns to be more likely under the policy, π_θ.

4 Learning Steerable LLM Policies

We now formally describe the connection between GRPO and DPO (§ 4.1), and introduce a simple modification to help stabilize GRPO for control policies from online reinforcement learning(§ 4.2). While our insights echo the DPO derivation and informal descriptions of DPO, our contribution makes this equivalence explicit in the two-trajectory case, which predicts failure modes (e.g., no valid negatives) that we observe empirically (§ 5).

4.1 From GRPO to Reference-Free DPO

Consider a group of trajectories $\mathcal{T} = \{\tau_1, \ldots, \tau_n\}$ generated by an LLM policy from the same initial state and under the same objective prompt (e.g., "drive aggressively"). Each trajectory $\tau \in \mathcal{T}$ is associated with a scalar reward $r(\tau)$, either from a known reward function or from human preference comparisons. Recall that GRPO computes a relative advantage for each trajectory by normalizing reward within the group: $A_\tau = \frac{r(\tau) - \mu_\mathcal{T}}{\sigma_\mathcal{T} + \epsilon}$. For a group containing only two trajectories, τ^+ and τ^- with rewards $+1$ and -1, respectively, the advantage for each trajectory is also exactly $+1$ and -1. Per the standard policy gradient update (§ 3), we can observe that, in this specific case, the policy gradient update is exactly the DPO learning update without a reference policy. In other words, with two trajectories achieving returns of $+1$ and -1, the GRPO loss is exactly a *reference-free* DPO loss:

$$\sum_{\tau \in \{\tau^+, \tau^-\}} A_\tau \log \pi_\theta(\tau) = 1 \cdot \log \pi_\theta(\tau^+) - 1 \cdot \log \pi_\theta(\tau^-) \tag{2}$$

Under these assumptions, we can therefore interpret DPO as a constrained version of GRPO (or any policy gradient method). Where GRPO weights actions or trajectories by their relative advantage and samples from a group of multiple rollouts, DPO instead binarizes weights to either 1 or -1 (chosen or rejected, respectively), and DPO samples only two of the group's rollouts. While we recognize that a group size of two is impractical, the connection illustrates that the algorithms are fundamentally optimizing for very similar objectives, namely, to increase the frequency of high-advantage sequences and decrease the frequency of low-advantage sequences. These similarities suggest that DPO may work as an effective online learning approach for multi-turn control policies, but the binarization of advantage and the truncated sampling (i.e., only using the best and worst rollouts from a group, rather than sampling over the entire group) may lead to instability during training. This connection between DPO and GRPO also motivates a refinement to GRPO, as prior work highlights potential failure modes of DPO and proposes to a solution by adding a regularization term (§ 3) [55]. Considering this prior work and the connection between GRPO and DPO, it is natural to then extend this regularization term to GRPO for additional stability.

4.2 Building on GRPO with Behavior Cloning Regularization

To improve stability and convergence of GRPO for policy learning with LLMs, we propose a modification akin to behavior cloning (BC) on high-advantage sequences introduced in [55] (Equation 6). Specifically, we augment the GRPO objective with a maximum likelihood term for highest-scoring trajectory in each group:

$$\mathcal{L}_{\text{GRPO+}}(\theta) = \underbrace{-\mathbb{E}_{\tau \sim \mathcal{D}}\left[\min\left(\hat{r}_\tau(\theta) A_\tau, \text{clip}(\hat{r}_\tau(\theta), 1-\epsilon, 1+\epsilon) A_\tau\right)\right]}_{\text{GRPO objective}} \underbrace{-\lambda \log \pi_\theta(\tau^+)}_{\text{BC regularization}} \tag{3}$$

Where λ weights the contribution of the BC regularization term. This regularization term encourages the policy to explicitly model high-quality behaviors while still learning from relative preferences. In practice, we find that this reduces variance throughout training and improves the consistency of aligned behavior, particularly when pre-training has already covered the space of desirable behaviors, and the alignment stage is primarily about refining existing behaviors, rather than discovering new behaviors. When performing policy gradient updates, we set $\lambda = 1$ for the first step of each batch (while the update is still completely on-policy), and we set $\lambda = 0$ for subsequent steps with the same data, to avoid overfitting to the highest-scoring trajectory.

5 Evaluating Alignment in LLM Policies

We empirically evaluate the effectiveness of existing alignment techniques when applied to control policies implemented via LLMs. Our evaluation spans two domains of increasing complexity: a tokenized 2D gridworld and a high-fidelity shared-control racing environment. We use task objectives as our target preferences, focusing our work on exploring the trade-offs between alignment algorithms without noisy preference datasets or objectives. The alignment process is the same as in conventional multi-turn alignment, as the algorithm still needs to fit an unknown objective in each domain.

We begin with a gridworld domain (Fig. 4) designed to isolate key challenges in alignment. This environment enables precise control over reward functions and policy behaviors, allowing us to probe both single-objective alignment and multi-objective steerability across conflicting goals (e.g., "go to the bottom left" vs. "go to the top right"). We also evaluate the role of pre-training in this domain by comparing models trained from scratch versus those pre-trained on relevant tasks for each experiment.

We then move to a significantly more complex embodied control task: high-performance racing (Fig. 6) in a shared autonomy setting [13]. Here, the LLM must learn to act in a multimodal environment, jointly interpreting state features (ego trajectory, map features) and predicting low-level driving commands (steering, throttle, brake).This domain does not allow for passive pre-training; shared control requires the policy to be actively in the loop, making offline data collection impractical. The LLM must also learn to coordinate with a stochastic human-like partner, modeled as a heuristic controller with randomized acceleration inputs. This setup presents a challenging testbed for preference alignment, particularly under a limited data regime.

Across both domains, we begin with a Llama 3.2 1B Instruct [21] model as our base LLM. Actions are selected by sampling from a distribution over only the action tokens added to the LLM's vocabulary, and actions are generated by conditioning on the full episode sequence (i.e., the prompt with objectives and weights, and all current state-action pairs as interleaved tokens). The LLMs are updated via LoRA [26], and word embedding matrices are unfrozen to enable learning of new state and action tokens. Prompts, example sequences, and training details are included in the supplementary material. For both domains, we compare the following alignment techniques: Direct Preference Optimization (**DPO**) [45], Direct Preference Optimization Positive (**DPO-P**) [43], Contrastive Preference Optimization (**CPO**) [55], Group Relative Policy Optimization (**GRPO**) [49], and Group Relative Policy Optimization + BC Regularization (**GRPO+BC**) (§ 4.2)

5.1 Gridworld: Simplified Alignment in a Synthetic Domain

Environment and Tokenization. The state space for our gridworld experiments is set up as a 32×32 grid. At each timestep, the agent observes its current location and selects an action from one of 8 cardinal directions (N, NE, E, SE, S, SW, W, NW). For our *pre-trained* experiments, these new tokens are first learned by imitating "expert" policies (learned by standard policy iteration).

Multi-turn interactions in this environment are structured as interleaved (state, action) token pairs, with task descriptions prepended as a textual prompt (e.g., "go to the bottom left"). During training, the initial state is always randomly sampled from anywhere on the grid, and for evaluation the initial states are fixed to midpoints around the grid and to the center of the grid.

Reward Functions. We apply a set of clear rewards to evaluate alignment and steerability:

- **Single-objective**: Navigate to the top right corner,
- **Complementary objectives**: Navigate to the top right corner and follow a sigmoid curve,
- **Opposing-objectives**: Navigate to the top right corner *or* the bottom left corner, and
- **Multi-objective**: Navigate to any arbitrary binary mixture of corners (e.g., 30% top right + 70% top left) as dictated by the prompt

We provide visualizations and task prompt examples for each of these reward surfaces in the supplementary material (§ D). These reward functions target different axes of investigation for alignment of a long-horizon, multi-turn LLM policy. Results are averaged across five training seeds. Table 1 shows performance when aligning pre-trained models, where blue indicates improvement over the pre-trained initialization and red indicates degradation from the initialization. Table 2 shows results for aligning models without pre-training.

Table 1: Alignment Performance (Pre-trained, Gridworld only)

Method	Gridworld			
	Single	Complementary	Opposing	Multi-Objective
DPO-P	0.74 ± 0.17	0.52 ± 0.03	0.51 ± 0.09	0.16 ± 0.02
DPO	0.82 ± 0.07	0.83 ± 0.01	0.94 ± 0.02	0.62 ± 0.01
CPO	0.96 ± 0.01	**0.89 ± 0.01**	0.90 ± 0.06	**0.69 ± 0.01**
GRPO	**0.99 ± 0.01**	0.87 ± 0.00	**0.99 ± 0.00**	0.65 ± 0.02
GRPO+BC	**0.99 ± 0.01**	0.88 ± 0.00	0.96 ± 0.01	0.64 ± 0.01

Table 2: Alignment Performance (From-scratch, Gridworld + Shared-Control Racing)

Method	Gridworld				Shared Control Racing		
	Single	Complementary	Opposing	Multi-Objective	No Noise	Low Noise	High Noise
DPO-P	**1.00 ± 0.00**	0.30 ± 0.02	0.11 ± 0.08	0.17 ± 0.01	-7.98 ± 0.93	-9.88 ± 1.33	-9.54 ± 1.10
DPO	**1.00 ± 0.00**	0.60 ± 0.02	0.42 ± 0.13	0.52 ± 0.02	**9.76 ± 3.32**	**7.39 ± 4.92**	5.52 ± 2.33
CPO	0.98 ± 0.01	**0.88 ± 0.00**	0.55 ± 0.12	0.47 ± 0.02	-29.76 ± 0.01	-29.75 ± 0.01	-29.73 ± 0.02
GRPO	0.98 ± 0.00	0.81 ± 0.01	**0.89 ± 0.05**	0.52 ± 0.02	0.73 ± 0.95	5.75 ± 3.57	5.01 ± 3.75
GRPO+BC	0.97 ± 0.00	0.87 ± 0.00	0.86 ± 0.04	**0.66 ± 0.01**	3.66 ± 2.91	1.57 ± 1.49	**6.16 ± 2.10**

Single Objective Experiments. For pre-trained models in the single-objective task, the base LLM is first trained to imitate an optimal policy for navigating to any of the four corners of the grid. The alignment process is therefore about refining the existing knowledge of the LLM to upsample one mode (i.e., the top-right corner). This very simple task hides a complication for contrastive training objectives, such as DPO, CPO, or DPO-P: the policy is *already* optimal, so there is no benefit to negating any rollouts, as all rollouts at the start of alignment are optimal. This leads to instability in contrastive methods, as rollouts are negated despite successfully achieving their desired objectives. In Table 1, we present results for each alignment strategy, and we see that DPO-P and DPO both result in *reductions* in performance compared to the initial policy (indicated by the red cell shading). Meanwhile, the group-normalization of GRPO effectively diminishes gradients for a group of rollouts with nearly identical returns, meaning that existing knowledge is not lost during alignment for a pre-trained policy. Note that this issue is not present in models that are trained from scratch (Table 2). We provide training curves in Fig. 14, where this phenomenon is more clearly visible.

Complementary Objective Experiments. For the complementary objective experiment, pre-training enables the base-LLM to imitate an independent optimal policy for each objective (top-right and sigmoid). The subsequent alignment process is necessary to blend the reward functions into a policy which balances complementary objectives (a visualization of this function is shown in §D). As shown in Tables 1 & 2, most alignment strategies are able to recover this behavior, effectively blending the two objectives. We observe nearly the same results with and without pre-training, with slightly worse performance for DPO and DPO-P. Under this reward function, BC leads to policy improvements as both CPO and GRPO+BC are the top performing methods for models with and without pre-training.

Opposing Objective Experiments. The opposing objective experiment presents our first look at a test-time *steerable* policy, as the LLM must learn distinct behaviors that are activated by a prompt. As in the single-objective experiments, pre-trained base LLMs for the opposing objective experiments are trained to imitate navigation to all four corners. The opposing objectives therefore probe alignment strategies on their ability to recover distinct modes within a pre-training corpus, even when those modes may be direct opposites. We find that this setting is particularly challenging for contrastive preference methods (DPO, CPO, and DPO-P) when learning from scratch, as they seem to collapse to a single mode (i.e., pick one corner at the expense of the other). Akin to the single-objective setting, contrastive methods are also subject to degeneration when initialized with an optimal policy.

Multi-Objective Experiments. The multi-objective setting presents the most interesting alignment challenge in our synthetic setup, as it encompasses our three previous setups in a single sweep. In this setting, the base LLM is pre-trained to navigate to the four corners of the grid, but now the alignment task is to learn a steerable policy over all binary mixtures of reward functions. For alignment in this task, we first randomly sample a prompt and matching reward function from the set of all 52 possible combinations of the four corners (e.g., 10% top-left + 90% bottom-right). The model then rolls out a group of episodes under this randomly sampled reward function and performs alignment to learn to satisfy this reward function-prompt pair.

This challenge involves both retaining the existing knowledge of the LLM, learning new interpolated behavior modes, and learning a steerable policy for any randomly sampled reward function. In this setting, BC offers a substantial benefit for both contrastive and RL methods, as GRPO is more stable with BC regularization, and CPO outperforms all methods. When we remove pre-training, GRPO+BC achieves the highest performance, and in both settings (with and without pre-training), BC regularization stabilizes GRPO's training, preventing policy degradation. We show full training curves in (§ H) to highlight this stabilizing effect, as GRPO suffers unexpected drops in performance while GRPO+BC continuously improves throughout training.

5.2 Shared-Control Racing: High-Dimensional Online Alignment

While synthetic domains are useful for targeted analysis of algorithmic performance without confounding variables of multi-modality or complex control, we are interested in understanding how to align LLMs as policies for more complex tasks. We therefore turn to shared-control racing as a demonstration task featuring multimodal inputs (trajectory and map features), 2048 new action tokens, and entirely online behavior learning without relevant pre-training data.

In this task, the LLM is sharing control of a vehicle with a synthetic human in a race against an automated opponent. We use the CARLA driving simulator [16, 13], which features realistic physics and vehicle dynamics. The LLM's goal in this domain is to win a race against an automated opponent while sharing control with a synthetic human driver. Both the partner agent (i.e., the synthetic human) and the opponent agent are controlled by heuristics that follow an optimal racing line, and both agents use random acceleration values to introduce stochasticity in the driving behaviors. As both agents are programmed to follow the same racing line and the ego agent starts behind the opponent, the ego agent will never win the race without input from the LLM. Therefore, we expect the LLM to learn assistive behaviors that deviate from the racing line to overtake the opponent and win the race.

Rather than using text for state data, we define a trajectory encoder and a map encoder, each consuming continuous-valued state features (e.g., position, velocity, orientation) and projecting those features into input tokens for the LLM sequence. The trajectory and map encoders are fully-connected networks mapping from input dimensions up to the LLM's hidden dimension, and are trained from scratch during the alignment process. Similarly, the LLM is initialized with new tokens for the 2048 possible actions it can take, and must learn the values for these embeddings from scratch over the course of training. The reward function is defined as a mixture of: **Progress** (move towards the finish line), **Pass** (overtake the opponent and get as far ahead as possible), **Bounds** (stay near the center of the track), **Collision** (avoid collisions), **Completion** (cross the finish line).

In addition to partnering with the heuristic controller from training, we also evaluate each method when paired with controllers that have random noise injected at each step. We experiment with a "Low Noise" test (random normal with $\sigma = 0.05$) and a "High Noise" test ($\sigma = 0.1$).

We show the average reward achieved by the best-performing training run under five evaluation seeds under the no noise, low noise, and high noise settings in Table 2, where blue indicates improvement relative to the pure-pursuit agent driving alone (i.e., helpful assistance), and red indicates degradation (i.e., harmful assistance). We also include example rollouts from each method in the supplementary material to better illustrate the various behaviors learned by the assistive agents.

While this task is solved effectively by DPO and GRPO, we also see *negative effects* from BC regularization, particularly when applying CPO. As the LLM is not pre-trained for multi-modal data, early rollouts are highly random and sub-optimal. BC regularization therefore encourages imitation of bad rollouts. Curiously, we also observe that GRPO scoring improves with "High noise". This owes to random acceleration values further propelling the ego agent down the track, particularly after the ego has already taken the lead (i.e., random noise can widen the gap after taking the lead).

6 Discussion

Examining our results first in the gridworld domains, we find that most off-the-shelf alignment techniques can learn effective multi-turn policies for control in this simple domain, depending on their pre-training and the difficulty of the task. Interestingly, we observe that **aligning pre-trained models with contrastive methods can actually lead to policy degradation** if the pre-trained policy is already optimal (Table 1). While from-scratch policies discover optimal behavior and remain stable,

we observe degeneration if the pre-trained base model has already memorized perfect behavior. This degeneration appears to be related to the lack of valid negative examples when the model is already optimal. In appendix § G we show log probabilities of the chosen and rejected rollouts when training a from-scratch vs. pre-trained DPO model on the simplest gridworld task (single reward), showing that the pre-trained model cannot separate positive and negative rollouts. This result highlights a potential failure mode: **contrastive methods require valid negative examples to avoid degeneration, and negative samples become increasingly rare as the model becomes increasingly optimal.** This finding reflects prior work on failure modes of DPO, identifying that DPO can get stuck in bad local optima without an effective reward model [36, 56], which may also indicate attempts to "push apart" sequences that are not meaningfully different. Training with group-relative advantage, on the other hand, will simply zero out advantage for a batch that is uniformly optimal, thereby effectively stopping training if the model is optimal and potentially avoiding this pitfall.

In the opposing- and multi-objective tasks, we observe that **contrastive methods are prone to collapse when they explore online to discover multi-modal behaviors**. Specifically, CPO and DPO achieve high returns with pre-trained LLMs (Table 1), but collapse to only a small subset of alignment objectives when they must discover opposing behaviors online (Table 2). GRPO and GRPO+BC, on the other hand, are well-suited to online exploration, and exhibit less degradation when switching from pre-trained to from-scratch LLM policies. We present full training curves in the supplementary material (§ H) to illustrate this degradation for both contrastive and RL-based methods.

Finally, considering our high-fidelity shared-control racing domain, we find that most alignment algorithms are able to discover effective assistive strategies (i.e., multi-turn control policies) with online exploration and reward-labeled trajectories (see Table 2). We observe that the CPO policy is degenerate, always opting to drive directly out of bounds as fast as possible. This appears to be the result of falling into a bad local optimum, as this achieves higher reward than completely stopping or crashing, which are other likely behaviors early in training. Unfortunately, under the CPO objective, there is no reference policy or regularization to prevent the model from doubling-down on this bad local optimum, thereby leading to rapid policy degradation. DPO and GRPO learn successful assistive strategies, and the resulting LLM is robust to noisy partner agents, despite being trained under the "no noise" setting. This finding reinforces the potential upsides of using LLMs as actors, as the model appears to generalize out-of-the-box to new evaluation settings.

In summary, we find that:

1. **Contrastive alignment strategies (such as DPO) are ill-suited to refine policies that are already optimal**, as they require valid negatives and degenerate if all rollouts are optimal.
2. **Contrastive alignment strategies struggle when they must discover *multi-modal* behavior distributions** or policies from a pre-trained model.
3. **Group-relative advantage offers more stable gradient updates for refining already-optimal policies**, as there is little or no negative gradient applied to high-scoring rollouts.
4. **Adding a BC regularization term helps to stabilize GRPO's training**, particularly for a pre-trained model, reducing the risk of policy degradation without harming exploration.

7 Conclusion & Future Work

As LLMs tackle long-horizon, multi-turn tasks, alignment introduces unique challenges of stability, preference optimization, and steerability. In this work we systematically compared alignment methods for LLM policies, highlighting their strengths and pitfalls in online, low-data, and multi-objective settings. We showed that GRPO can be viewed as a weighted, reference-free variant of DPO, and that adding behavior cloning regularization improves stability in certain regimes. Our results provide practical guidance: stability often comes from combining online learning with implicit regularization, while contrastive methods risk degeneration without valid negatives.

There are several directions for future work. A natural extension of GRPO+BC is to study sensitivity to the BC weight and consider refinements such as advantage-weighted regularization. More broadly, improving steerability and multi-modal behavior discovery for contrastive methods remains an open challenge. Finally, while our gridworld experiments examined multi-objective mixtures, future work could investigate rich text-only multi-turn tasks such as interactive planning, web navigation, or collaborative creativity, where long-term consistency plays a larger role in successful interaction.

References

[1] Josh Abramson, Arun Ahuja, Iain Barr, Arthur Brussee, Federico Carnevale, Mary Cassin, Rachita Chhaparia, Stephen Clark, Bogdan Damoc, Andrew Dudzik, et al. Imitating interactive intelligence. *arXiv preprint arXiv:2012.05672*, 2020.

[2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

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

[4] Erdem Biyik, Nima Anari, and Dorsa Sadigh. Batch active learning of reward functions from human preferences. *ACM Transactions on Human-Robot Interaction*, 13(2):1–27, 2024.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[7] Howard Chen, Huihan Li, Danqi Chen, and Karthik Narasimhan. Controllable text generation with language constraints. *arXiv preprint arXiv:2212.10466*, 2022.

[8] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14093–14100. IEEE, 2024.

[9] Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning through imitation and reinforcement. *arXiv preprint arXiv:1805.10413*, 2018.

[10] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc.

[11] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu,

Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[12] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

[13] Jonathan DeCastro, Andrew Silva, Deepak Gopinath, Emily Sumner, Thomas M. Balch, Laporsha Dees, and Guy Rosman. Dreaming to assist: Learning to align with human objectives for shared control in high-speed racing. In *8th Conference on Robot Learning (CoRL) 2024. Munich, Germany*, 2024.

[14] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

[15] Yi Dong, Zhilin Wang, Makesh Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. SteerLM: Attribute conditioned SFT as an (user-steerable) alternative to RLHF. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11275–11288, Singapore, December 2023. Association for Computational Linguistics.

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[17] Evan Ellis, Gaurav R Ghosal, Stuart J Russell, Anca Dragan, and Erdem Bıyık. A generalized acquisition function for preference-based reward learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2814–2821. IEEE, 2024.

[18] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023.

[19] Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. Towards analyzing and understanding the limitations of dpo: A theoretical perspective. *arXiv preprint arXiv:2404.04626*, 2024.

[20] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

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

[22] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[23] Raghav Gupta, Ryan Sullivan, Yunxuan Li, Samrat Phatale, and Abhinav Rastogi. Robust multi-objective preference alignment with online dpo. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26):27321–27329, Apr. 2025.

[24] Joey Hejna, Chethan Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Re-mix: Optimizing data mixtures for large scale imitation learning. *arXiv preprint arXiv:2408.14037*, 2024.

[25] Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh. Contrastive preference learning: Learning from human feedback without rl. In *ArXiv preprint*, 2023.

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[27] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, Yin Zhou, James Guo, Dragomir Anguelov, and Mingxing Tan. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

[28] Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hanna Hajishirzi. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *Advances in neural information processing systems*, 37:36602–36633, 2024.

[29] Jiaming Ji, Xinyu Chen, Rui Pan, Han Zhu, Conghui Zhang, Jiahao Li, Donghai Hong, Boyuan Chen, Jiayi Zhou, Kaile Wang, et al. Safe rlhf-v: Safe reinforcement learning from human feedback in multimodal large language models. *arXiv preprint arXiv:2503.17682*, 2025.

[30] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: investigating the design space of visually-conditioned language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[31] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 10, 2023.

[32] Saeed Khaki, JinJin Li, Lan Ma, Liu Yang, and Prathap Ramachandra. RS-DPO: A hybrid rejection sampling and direct preference optimization method for alignment of large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1665–1680, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[33] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[34] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[35] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[36] Ziniu Li, Tian Xu, and Yang Yu. Policy optimization in rlhf: The impact of out-of-preference data. *arXiv preprint arXiv:2312.10584*, 2023.

[37] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[38] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

[39] Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. Learning to navigate in cities without a map. *Advances in neural information processing systems*, 31, 2018.

[40] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6292–6299. IEEE, 2018.

[41] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback, 2022. *URL https://arxiv. org/abs/2112.09332*, 2022.

[42] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[43] Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

[44] Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. Enhancing alignment using curriculum learning & ranked preferences. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12891–12907, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[45] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[46] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[47] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[48] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[49] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[50] Laura Smith, Alex Irpan, Montserrat Gonzalez Arenas, Sean Kirmani, Dmitry Kalashnikov, Dhruv Shah, and Ted Xiao. Steer: Flexible robotic manipulation via dense language grounding. *arXiv preprint arXiv:2411.03409*, 2024.

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

[52] Jiangyuan Wang, Kejun Xiao, Qi Sun, Huaipeng Zhao, Tao Luo, Jiandong Zhang, and Xiaoyi Zeng. Shoppingbench: A real-world intent-grounded shopping benchmark for llm-based agents. *arXiv preprint arXiv:2508.04266*, 2025.

[53] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

[54] Senwei Xie, Hongyu Wang, Zhanqi Xiao, Ruiping Wang, and Xilin Chen. Robotic programmer: Video instructed policy code generation for robotic manipulation. *arXiv preprint arXiv:2501.04268*, 2025.

[55] Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: pushing the boundaries of llm performance in machine translation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[56] Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

[57] Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for llm-based web agents. *arXiv preprint arXiv:2410.13825*, 2024.

[58] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

[59] Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10586–10613, 2024.

A Broader Impact Statement

As with much of the current LLM research, there are potential positive and negative consequences of our research into embodied LLM policies. Any alignment technique can be used for improving safety, user satisfaction, or usability of an embodied policy, but it can likewise be used to make a model deceptive, dangerous, or antagonistic. Further, hidden LLM biases can have unforeseen effects on embodied LLM policies. While these are valid concerns that one must address before releasing an embodied LLM into the wild, they are beyond the scope of this work.

In this work, we have demonstrated an example of applying an LLM to learning to share vehicle control with a synthetic human in a high-speed race against an automated opponent. In reality, deploying such a system would require significant engineering to produce ensure safe operation and alignment between the user and the embodied policy, which we have not addressed in this work.

Our work has provided insights into failure modes for applying off-the-shelf alignment algorithms to embodied policy learning. Our findings, such as identifying that common techniques (e.g., DPO) struggle to discover multi-modal behaviors, may provide useful insights to future researchers and engineers looking to build safer and more reliable embodied policies via LLMs.

B Limitations

Our work has several limitations that we would like to specifically mention. First, while many contemporary works are pursuing embodied LLM policies, the specific pre-training recipes, fine-tuning schemes, and action decoding strategies vary across these works, and these may all have effects on the resulting performance of different alignment algorithms. Our hope with this work is to illustrate the trends that we observe in alignment algorithm performance and to specifically call out the problem of embodied alignment, not to present the definitive characterization of alignment for embodied LLM policies. Second, while our work considered multimodal input from trajectory and map features, we have not explored image input or pre-trained VLA models [42, 33, 5]. While our findings in theory generalize to any method that similarly tokenizes input modalities into the LLM sequence, this remains untested. Finally, our work is confined to a 1B Llama3 model, and we have not evaluated how these findings scale to significantly larger models.

C Preliminaries: PPO, GRPO, & DPO

Proximal Policy Optimization (PPO) [48] builds on REINFORCE by improving stability and sample efficiency, introducing a clipped surrogate objective that limits the scale of policy updates:

$$\mathcal{L}_{\text{PPO}}(\theta) = \mathbb{E}_t \left[\min \left(\hat{r}_t(\theta) A_t, \text{clip}(\hat{r}_t(\theta), 1-\epsilon, 1+\epsilon) A_t \right) \right], \tag{4}$$

where $\hat{r}_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between the new and old policies, and ϵ is a clipping parameter. PPO offers improved stability and simplicity relative to base policy-gradient methods, while retaining high performance.

Group Relative Policy Optimization (GRPO) [49] simplifies the learning process by computing advantage using rewards from a group of rollouts with the same initial state and the same objective, rather than relying on a value function to score rollouts. GRPO computes a relative advantage for each trajectory by normalizing its reward within the group: $A_\tau = \frac{r(\tau) - \mu_\mathcal{T}}{\sigma_\mathcal{T} + \epsilon}$ where $\mu_\mathcal{T}$ and $\sigma_\mathcal{T}$ are the mean and standard deviation of rewards within a group $\mathcal{T}$, and ϵ is a small constant for numerical stability. In practice, GRPO can make learning more stable while reducing the computational requirement for training, as we do not need to concurrently learn a value function to compute advantage scores. Given these advantages, we opt to investigate GRPO over PPO for embodied policy learning.

Direct Preference Optimization (DPO) [45] further simplifies preference learning by directly optimizing a policy to satisfy human preferences, without learning an intermediate reward model or applying reinforcement learning algorithms. Given trajectory pairs (τ^+, τ^-) and a shared prompt q, DPO minimizes the reference-model-regularized binary preference loss:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(q,\tau^+,\tau^-)\sim\mathcal{D}} \left[\log \hat{\sigma} \left(\beta \left(\log \frac{\pi_\theta(\tau^+ \mid q)}{\pi_{\text{ref}}(\tau^+ \mid q)} - \log \frac{\pi_\theta(\tau^- \mid q)}{\pi_{\text{ref}}(\tau^- \mid q)} \right) \right) \right] \tag{5}$$

where π_θ is the preference-aligned policy, π_{ref} is the reference policy, $\log \hat{\sigma}$ is the log-sigmoid function, and β is an inverse temperature parameter. This formulation is both efficient and robust, and has been widely adopted in large-scale LLM alignment. Subsequent work introduced modifications to the sampling strategies for DPO [44, 32] that leverage additional information from a group of rollouts rather than simple binary comparisons. The recently introduced Contrastive Preference Optimization (CPO) [55] approach drops the reference policy normalization in DPO and adds a log-likelihood maximization term for the preferred trajectory, τ^+:

$$\mathcal{L}_{\text{CPO}}(\theta) = -\mathbb{E}_{(q,\tau^+,\tau^-)\sim\mathcal{D}}\left[\log \hat{\sigma}\left(\beta\left(\log \pi_\theta(\tau^+ \mid q) - \log \pi_\theta(\tau^- \mid q)\right)\right) - \lambda \log \pi_\theta(\tau^+ \mid q)\right] \quad (6)$$

which can be interpreted as a behavior-cloning regularizer [25]. While a simple and effective modification, CPO risks degeneration if applied to online exploration by blindly supervising to a preferred sequence and by abandoning a reference policy. However, assuming access to clean data and a performant initialization, this modification allows CPO to improve over DPO.

D Gridworld Reward Surfaces and Prompts

Figure 2: Four corner reward functions for our synthetic gridworld experiments.

Our work uses five base reward functions: navigation to the four corners of the grid (Figure 2, and travel along the sigmoid function. For our **Single Objective** experiments, we align only to the top-right objective. For our **Complementary Objectives**, we blend the top-right and sigmoid functions, leading to the reward function visualized in Figure 3 (encouraging the agent to travel to the sigmoid function and then to the top-right of the grid). For the **Opposing Objectives**, the agent is aligned to the top-right *or* the bottom-left objective, depending on the input prompt, but the same model must learn to satisfy *both* behaviors by conditioning on the input prompt. Finally, for the **Multi-Objective** experiment, we blend all 10% mixtures of the four corner reward functions. This leads to 52 possible reward functions (e.g., 10% top-left + 90% top-right), where the optimal behaviors are often overlapping or related (e.g., travel along the top of the grid and towards the top-right corner), but specific instantiations of the optimal behaviors will vary for each pair of functions.

The prompts used for our gridworld domains all follow the form:

```
{Reward description}: {weight} {state_0} {action_0} {state_1} {action_1}...
```

For example, when considering our **Single-Objective**, **Opposing-Objective**, or **Multi-Objective** setting, model might see:

```
Go to the top right: 1.0, Go to the bottom left: 0.0, Go to the top left: 0.0, Go
    to the bottom right: 0.0. <|-s(15,15)-|><|a:N|><|-s(15,16)-|><|a:N|>...
```

for an episode starting in the center of the grid and moving up.

For our **Complementary Objective** setting, the model prompt would instead be:

```
Go to the top right: 0.3, Go to the sigmoid: 0.7 <|-s(15,15)-|><|a:N|><|-s(15,16)
    -|><|a:N|>...
```

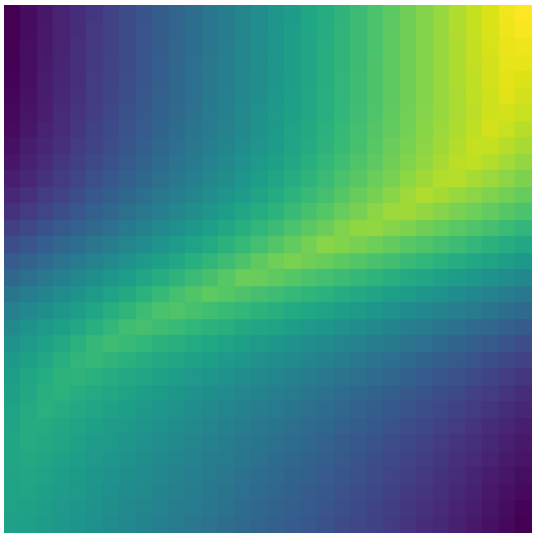
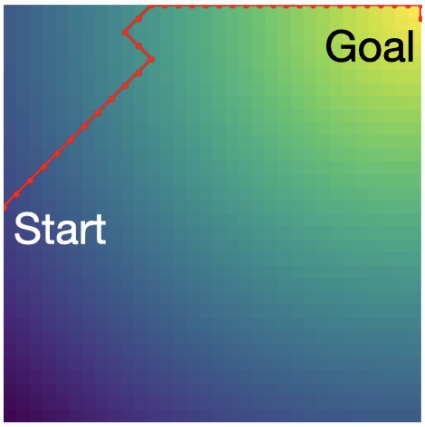
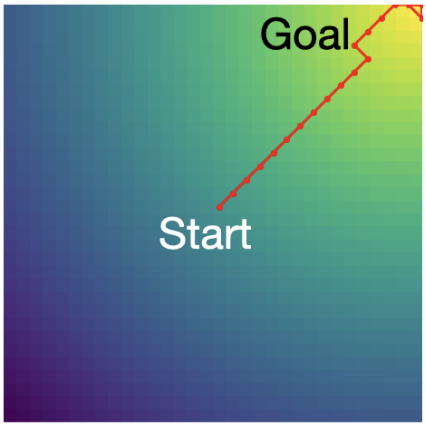


Figure 3: Blended reward for top-right + sigmoid

We present example rollouts under a well aligned model (Figure 4) and a model that suffers mode collapse (Figure 5) to illustrate the domain, the task, and examples of successful and unsuccessful policies.



(a) Left-Side Start



(b) Center Start

Figure 4: Gridworld rollouts under a model that is successfully aligned to the target reward function. Lighter colors indicate more reward, and the red line illustrates the model's path.

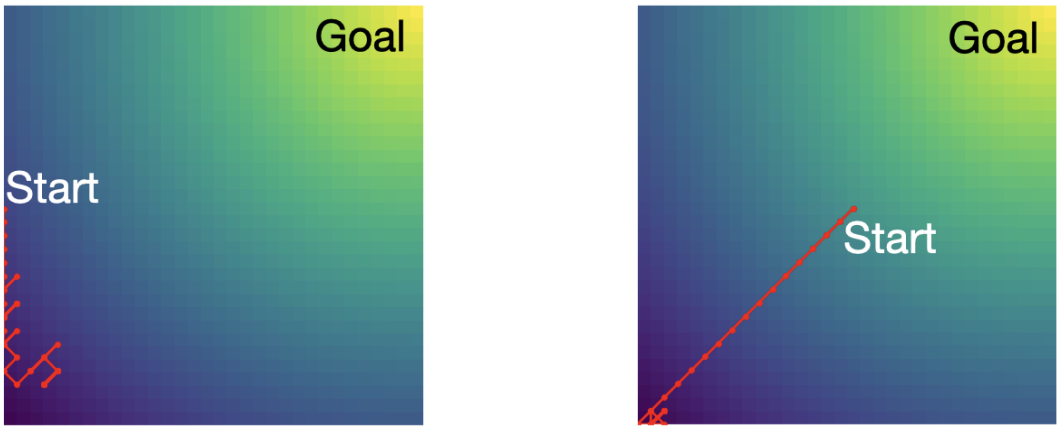
E Shared Control Prompts and Rollouts

When the embodied LLM is deployed to the shared control setting, it receives the following prompt:

```
Stay central: 0.5. Penalty for collisions: -5.0. Follow: 0.0. Stay left: 0.0.
    Overtake: 0.2. Proceed: 1.0. Stay right: 0.0. {Map tokens} {Trajectory tokens}
```

where multi-modal map data is encoded and slotted into the "{Map token}" and trajectory input data is encoded and slotted into the "{Trajectory token}" spots.

We show a visual example from the CARLA simulator for a rollout in Figure 6, and we show example rollouts using each method (and one with no assistance) in Figures 7 - 12.



(a) Left-Side Start (b) Center Start

Figure 5: Gridworld rollouts under a model that has suffered mode collapse, navigating towards the bottom left regardless of the input prompt and reward function. Lighter colors indicate more reward, and the red line illustrates the model's path.

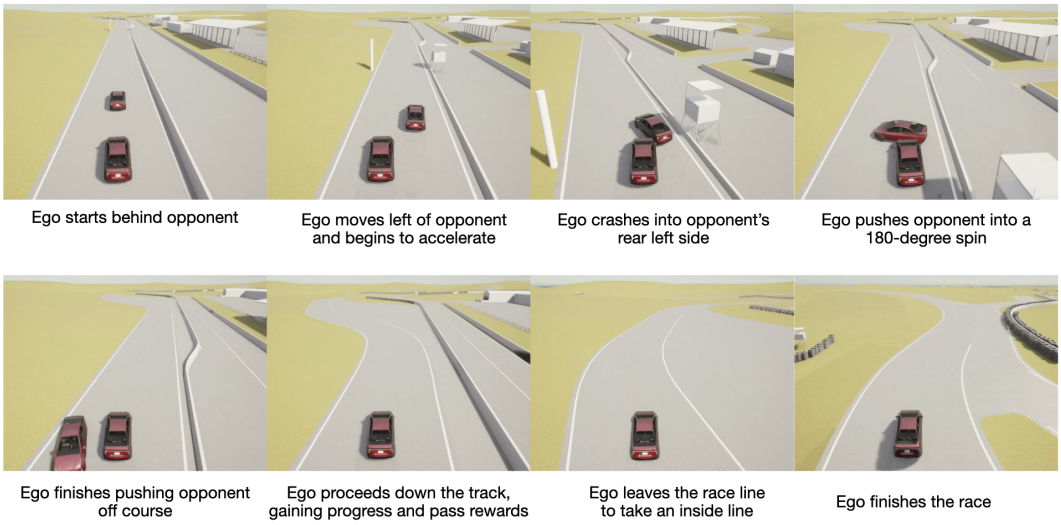


Figure 6: Shared-control rollout learned using DPO for alignment. The LLM learns a highly successful, albeit unexpected, strategy of crashing into the opponent, suffering a collision penalty but gaining significantly more long-term reward via the pass objective.

F Training Details

To pre-train our gridworld agents, we first generate optimal rollouts under each reward mixture using value iteration with $\gamma = 0.95$ and a convergence threshold of 10^{-4}. The maximum rollout length is set to 64 steps, which we re-use in our alignment setup. The grid size is set to 32×32.

Model: We use a LLaMA-3.2-1B-Instruct backbone with a LoRA adapter (r=16, α=16, dropout=0.1) applied to attention, MLP, and word embedding layers. The tokenizer is extended with special tokens for states and actions (e.g., `<|s:(x,y)|>`, `<|a:NE|>` in the gridworld, or `<|a:1024|>` in the shared-control domain).

Optimization details: Note that we use 5000 episodes for our **Multi-Objective** experiments, as these took longer to converge.

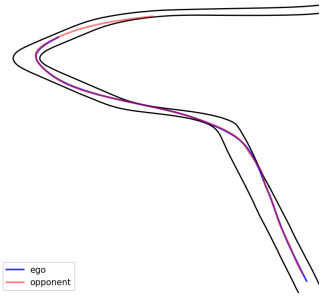


Figure 7: Shared-control rollout with only the pure-pursuit agent (i.e., no assistance). We see that the ego agent simply follows the opponent, failing to win the race.

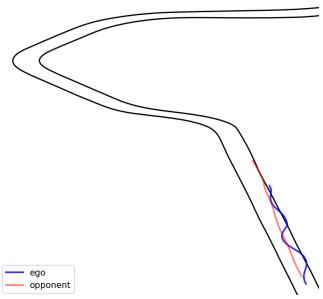


Figure 8: Shared-control rollout with DPO-P assistance. We see that the ego agent simply follows the opponent, failing to win the race.

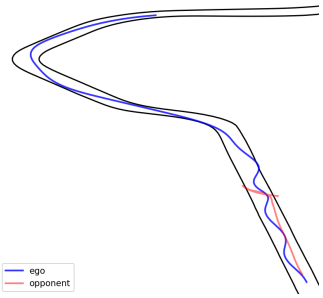


Figure 9: Shared-control rollout with DPO assistance. DPO learns an effective assistive strategy, deliberately crashing into the opponent to "fish tail" the opponent off the track, and then easily winning the race. While this strategy incurs a reward penalty for collisions, the net effect is a highly positive return due to the significant distance between the ego and the opponent, and due to the ego's chance to win and obtain the race completion bonus.

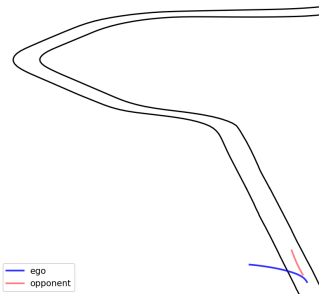


Figure 10: Shared-control rollout with CPO assistance. CPO degenerates into a bad local optimum, identifying that the left side enables quick gains per the "progress" reward, and avoids penalties per the "collision" reward. However, this strategy, while superior to other early exploration strategies, is clearly sub-optimal. Unfortunately, without reference policy regularization and with a BC term to reinforce the best rollout in each group, CPO becomes trapped in this behavior and further reinforces itself for the remainder of training.

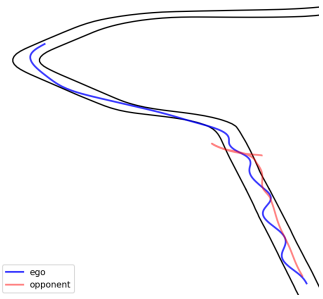


Figure 11: Shared-control rollout with GRPO assistance. GRPO identifies a similar strategy to DPO, spinning out the opponent to open up a wide lead and gain significant "pass" reward bonuses.

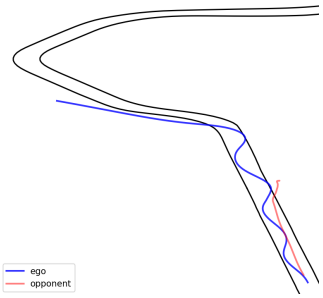


Figure 12: Shared-control rollout with GRPO + BC regularization assistance. GRPO + BC also learns to collide with the opponent, ramming the opponent into a wall and continuing the race. However, GRPO+BC learns to maximize acceleration away from this collision, to gain as much "progress" and "pass" reward as possible, which then causes the ego agent to leave the track bounds prematurely.

Algorithm 1 Supervision Loop for Preference Optimization

Require: Policy model π_θ, reward functions $\mathcal{R}$, batch size $\mathcal{B}$, group size $\mathcal{G}$
for each training episode **do**
  for each batch in $\mathcal{B}$ **do**
    Sample reward function $\lambda \sim \mathcal{R}$
    Generate G trajectories $\{\tau_i\}_{i=1}^{G}$ from policy π_θ with reward prompt λ
    for each trajectory τ_i **do**
      Compute per-step rewards $r_i^t = R(\tau_i^t)$ using weights λ
      Compute discounted returns $R_\tau = \sum_t \gamma^t r_i^t$
    end for
    Compute normalized advantages A_τ using group statistics
  end for
  Update model using selected preference learning algorithm
end for

Parameter	Value
Episodes	200 (or 5000)
Group size	8
Batch size	4
Max steps per rollout	64
Learning rate	5×10^{-5}
Optimizer	AdamW
Clip coefficient (GRPO)	[-0.2, +0.4]
DPO β	0.1
DPO-P hinge λ	1.0

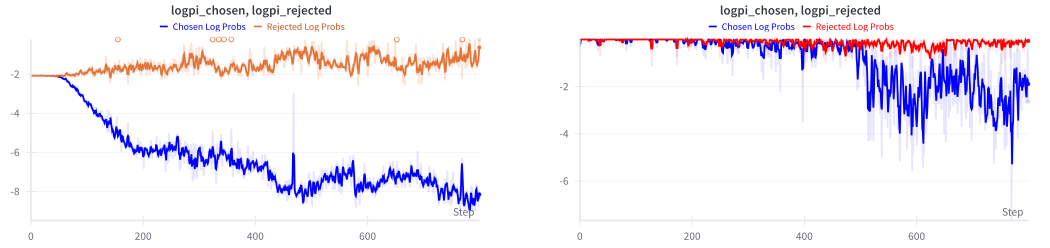
All experiments are conducted on a single NVIDIA RTX Ada 6000 48Gb GPU. **Single Objective**, **Complementary Objective**, and **Opposing Objective** experiments take, on average, 50-60 minutes to complete. **Multi-Objective** experiments take, on average, 18-24 hours to complete. **Shared Control** experiments take, on average, 15-18 hours to complete.

G DPO Degeneration Investigation

In Section 6, we discuss a possible failure mode of DPO and of contrastive methods more broadly: when such methods are pre-trained to near optimal solutions, they do not generate valid negative samples, thereby leading the model to degenerate when performing gradient updates. Here, we present a small investigation into this phenomenon. We re-trained two DPO models, one from-scratch and one with pre-training, on our simple **Single Objective** gridworld setting. For every gradient step over the course of training, we plotted the log probabilities of the chosen sequence, τ^+, and the rejected sequence, τ^-. Figure 13a shows these log probabilities as they evolve over the course of from-scratch training, while Figure 13b shows the log probabilities over the course of pre-trained training. From these figures, it is clear that the pre-trained model is a much harder time disentangling the two sequences, as we see that the chosen sequence does not begin to diverge from the rejected sequence until nearly halfway through training. Conversely, in the from-scratch experiment, the sequence likelihoods begin to diverge almost immediately, suggesting a clean learning signal that the model is able to exploit for better gradients (and returns).

H Training Curves For All Domains

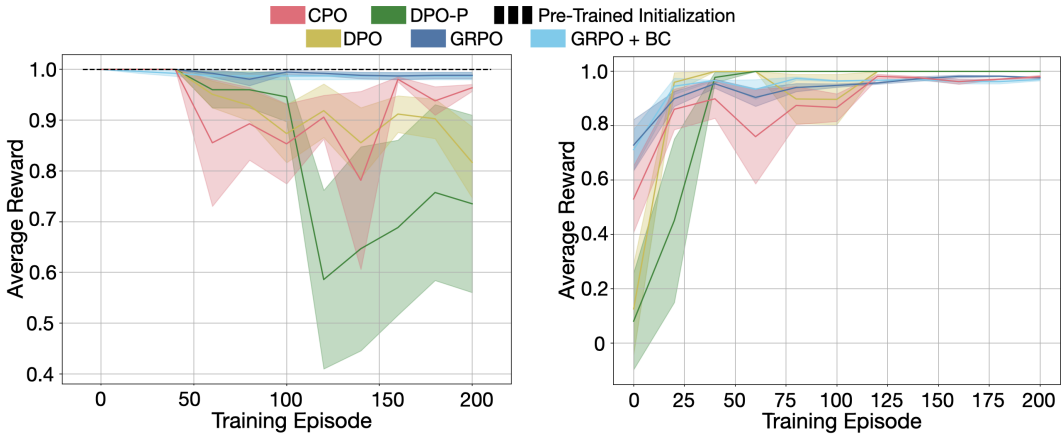
In this section, we provide training curves for our synthetic and shared-control tasks. In particular, these figures highlight the policy degradation that we discuss in Section 6, particularly Figures 14 & 16, where the pre-trained model is optimal, but the contrastive methods degenerate.



(a) Chosen and Rejected Log Probs When Training From Scratch

(b) Chosen and Rejected Log Probs When Training From a Pre-Trained Initialization

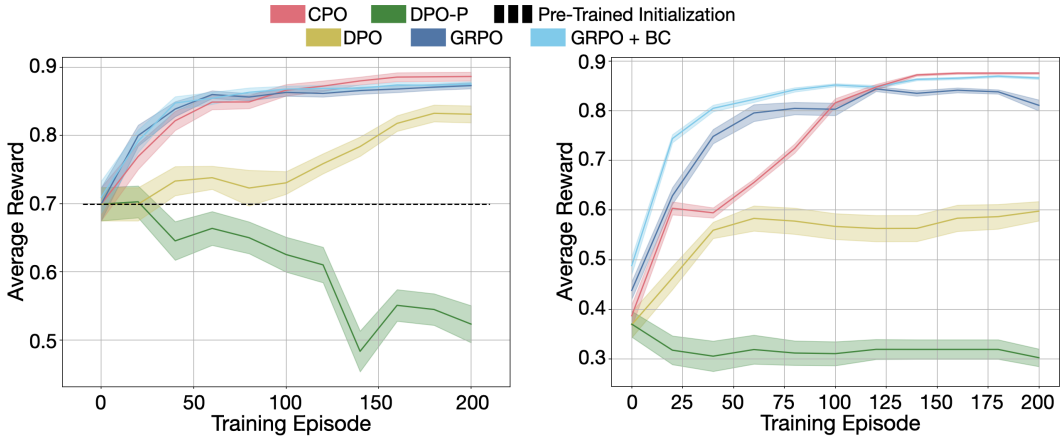
Figure 13: Log probabilities for an LLM policy being aligned using DPO in a simple gridworld.



(a) Pretrained

(b) From Scratch

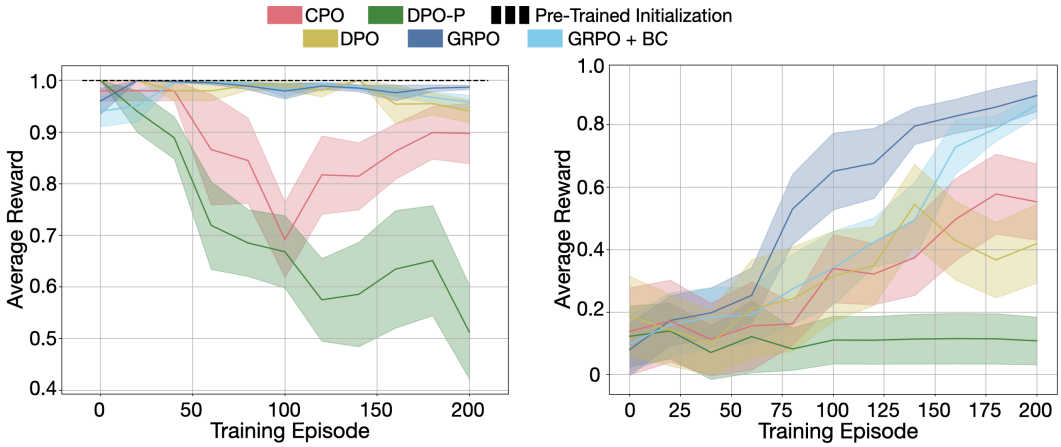
Figure 14: Single Reward



(a) Pretrained

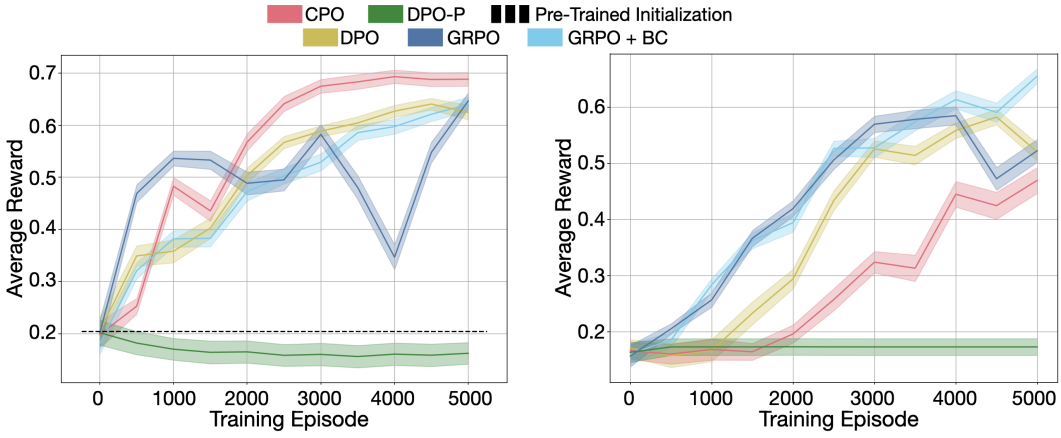
(b) From Scratch

Figure 15: Complementary Rewards



(a) Pretrained (b) From Scratch

Figure 16: Opposing Rewards



(a) Pretrained (b) From Scratch

Figure 17: Reward Mixtures

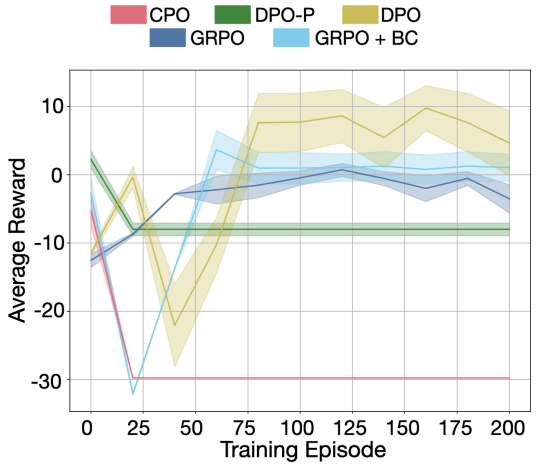


Figure 18: Shared-Control Racing Reward Over Training