GIFT: Generalizing Intent for Flexible Test-Time Rewards

Fin Amin[1], Nathaniel Dennler[2], Andreea Bobu[2]
[1]NC State, [2]MIT CSAIL

***Abstract*— Robots learn reward functions from user demonstrations, but these rewards often fail to generalize to new environments. This failure occurs because learned rewards latch onto spurious correlations in training data rather than the underlying human intent that demonstrations represent. Existing methods leverage visual or semantic similarity to improve robustness, yet these surface-level cues often diverge from what humans actually care about. We present Generalizing Intent for Flexible Test-Time rewards (GIFT), a framework that grounds reward generalization in human *intent* rather than surface cues. GIFT leverages language models to infer high-level intent from user demonstrations by contrasting preferred with non-preferred behaviors. At deployment, GIFT maps novel test states to behaviorally equivalent training states via *intent-conditioned similarity*, enabling learned rewards to generalize across distribution shifts without retraining. We evaluate GIFT on tabletop manipulation tasks with new objects and layouts. Across four simulated tasks with over 50 unseen objects, GIFT consistently outperforms visual and semantic similarity baselines in test-time pairwise win rate and state-alignment F1 score. Real-world experiments on a 7-DoF Franka Panda robot demonstrate that GIFT reliably transfers to physical settings. Further discussion can be found at https://mit-clear-lab.github.io/GIFT/**

I. Introduction

Imagine a human teaching a robot how to pack their bag for an art class (Fig. 1). The human might demonstrate placing a paintbrush and sketchbook in the bag, expecting the robot to understand their goal of gathering art supplies. Later, when preparing for a different art session, the human would naturally trust the robot to assist them autonomously with packing the right supplies—whether it's the same paintbrush and sketchbook, or instead molding clay, an easel, or any other materials needed for the task.

Unfortunately, while robots can learn reward functions from user demonstrations [1, 2], these reward functions rarely generalize to new scenarios. Instead, learned rewards often latch onto spurious correlations in the training data [3] (e.g., reward functions learn to "pack paintbrush") rather than capturing the underlying high-level intent (e.g., a reward function to "pack art supplies"). As a result, when robots encounter new scenarios at test time, the learned reward may fail catastrophically, causing the robot to either ignore important task aspects or fixate on irrelevant ones. Prior work has attempted to improve robustness through data augmentation based on visual [4] or language [5] similarity, but these measures often diverge from what humans actually care about when defining task success [6].

Our key insight is that *state similarity is driven by people's high-level task intent, not by surface-level visual or language cues*. In our example in Fig. 1, a vision-based model might

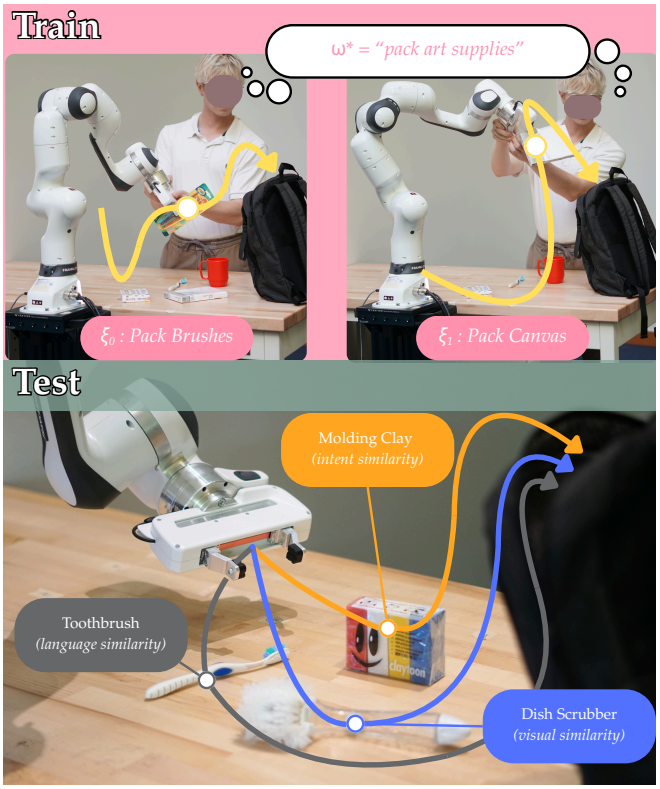

Fig. 1. **Top.** During training, the robot gets task demonstrations (loading a paintbrush) and uses them to infer the user's *high-level intent* ("load art supplies"). **Bottom.** At test time, the robot encounters three unseen objects. GIFT uses the inferred intent to correctly identify that the molding clay is relevant. In contrast, visual-similarity baselines incorrectly prefer the dish scrubber due to its appearance, and language-similarity baselines make an analogous mistake ("*tooth**brush***" and "*paint**brush.***")

judge a scrubber to be most similar to a paintbrush because they look alike, while a language model might instead group a paintbrush with a toothbrush due to lexical similarity. Neither captures what actually matters for the task: under the intent "pack art supplies," a paintbrush is closer to molding clay. In addition, different intents induce different notions of similarity: if the intent were instead "set aside painting supplies," then paintbrush, watercolor, and easel would be similar to the intention, while molding clay and sketchbook would be dissimilar. Without conditioning on intent, the robot cannot know which notion of similarity is appropriate, and thus risks generalizing along the wrong notion of similarity.

To address this issue, we introduce **G**eneralizing **I**ntent for **F**lexible **T**est-Time rewards (GIFT), a framework that grounds reward generalization in high-level intent. We leverage language models' (LMs) commonsense reasoning abilities in two stages. First, we infer the human's high-level

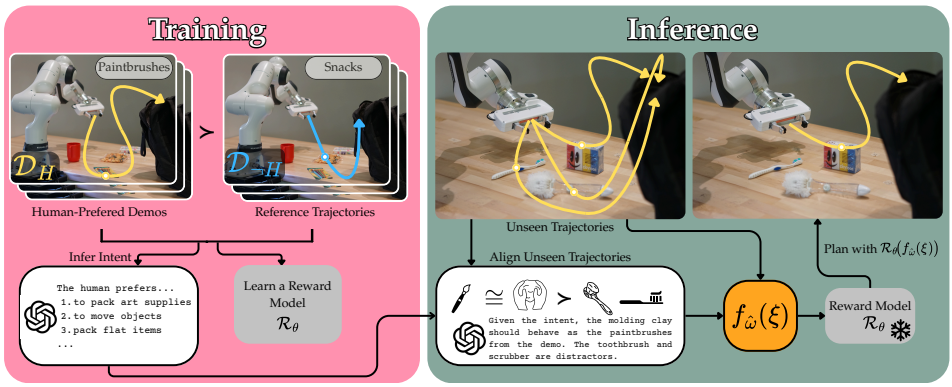

Fig. 2. **G**eneralizing **I**ntent for **F**lexible **T**est-Time rewards. **Left.** GIFT infers the human's intent given pairs of human-preferred demonstrations and reference trajectories. **Right.** During inference, GIFT deduces which objects in the unseen states should behave as objects in the training states. Afterwards, the unseen state components are aligned to training states so that the reward function learned before deployment can be used for planning.

intent by prompting LMs with contrasting pairs of user-preferred trajectories and non-preferred trajectories. Second, at test time, the inferred intent conditions another LM call that aligns novel test states to semantically similar training states. This lets the robot reuse its learned reward function without retraining. In experiments with a 7-DoF Franka Panda robot, both in simulation and the real world, we show that GIFT substantially outperforms low-level visual and language baselines on test-time reward generalization.

II. Related Work

Learning Rewards from Humans. Robots commonly learn human goals from demonstrations [1, 2, 7], preferences [8, 9], or corrections [10–12]. Language has recently emerged as a powerful medium for reward learning, enabling free-form text or natural language critiques to refine reward inference [13, 14]. Other methods leverage LMs as priors to propose interpretable reward features or to ground abstract task objectives directly into reward functions [15–18]. While these approaches make it easier for non-experts to convey their intent without hand-engineering reward functions, the resulting models often overfit to surface correlations in limited data, leading to *reward misspecification*—where the learned reward captures proxies of success rather than the user's true objective [19, 20]. GIFT aims to avoid reward misspecification by interpreting demonstrations as expressions of high-level *intent* that can be inferred and transferred across domains to enable flexible test-time reward generalization.

Test-Time Adaptation in Robots. Reward learning approaches often require users to teach the robot again when it encounters a new environment. Collecting data for every new environment can be an onerous process for users [21–24], so other techniques aim to transfer the robot's understanding of user preferences to unseen contexts at test time [4, 25–29]. Test-time adaptation is often achieved by using large pre-trained models to directly map from test-time observations to data seen during training by using textual correspondence between objects [28, 30], visual correspondence between videos [4], or by projecting test images to the training manifold of images [29]. In contrast to previous approaches that perform mappings that directly reflect the data distribution, GIFT aims to leverage higher-order representations of *user intention* that are robust to distribution shifts.

III. Problem Formulation

We consider the problem of adapting a robot's behavior to align with human objectives under distribution shift. Our goal is to enable a learned reward function to generalize across distribution shifts by grounding it in the human's high-level intent, rather than correlations present in the training domain.

Markov Decision Processes. We model our setting as a Markov Decision Process (MDP) [31] $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, where $\mathcal{S}$ is the state space (e.g., robot joint poses, object poses), $\mathcal{A}$ the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ the transition dynamics, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function. We denote trajectories as state-action sequences $\xi = (s_0, a_0, \ldots, s_{T-1}, a_{T-1}, s_T)$. The robot aims to find a policy $\pi : \mathcal{S} \to \mathcal{A}$ such that the trajectories induced by π^* maximize cumulative rewards, $\pi^* = \arg\max_\pi \mathbb{E}_{\xi \sim \pi}[\mathcal{R}(\xi)]$.

Reward Learning from Human Input. The reward is not known a priori, as it reflects the human's internal preference over how the robot should behave. We model this preference as a parameterized reward function $\mathcal{R}_\theta(\xi) = \sum_{s_t \in \xi} \mathcal{R}_\theta(s_t)$, where θ are learnable parameters. Because hand-specifying $\mathcal{R}_\theta$ is difficult, the robot can instead *learn* it from human input such as demonstrations, preferences, or corrections.

The robot's goal is to find parameters θ such that the reward $\mathcal{R}_\theta$ captures the human's underlying intent. This can be achieved through various approaches including inverse reinforcement learning (IRL) [1, 32], preference learning [8], learning from corrections [12], etc. Our method (Sec. IV) is agnostic to the reward learning algorithm. For concreteness, we focus our exposition on the IRL setting, where the algorithm seeks θ such that human-demonstrated trajectories $\mathcal{D}_H = \{\xi_i\}_{i=1}^N$ are more likely under $\mathcal{R}_\theta$ than alternative trajectories $\xi \in \Xi$, where Ξ denotes the set of all possible trajectories. In this formulation, demonstrations serve as evidence of desired behavior, and reward learning amounts to recovering the preference $\mathcal{R}_\theta$ that best explains them.

Distribution Shift. Standard reward learning assumes that training and test trajectories are drawn from the same distribution. In practice, however, robots are deployed in environments where test-time states differ from those observed

in training—due to new objects, layouts, or contexts. This distribution shift leads to two major failure modes: *reward misspecification* and *out-of-distribution misgeneralization*.

First, when demonstrations $\mathcal{D}^{\text{train}}$ cover only a subset of the state space $\mathcal{S}^{\text{train}} \subseteq \mathcal{S}$, the robot may recover a reward function $\mathcal{R}_\theta$ that explains the specific training data but fails to capture the human's true *high-level* intent. For example, demonstrations of keeping a coffee cup away from a laptop may be misinterpreted as "avoid laptops" rather than the intended "avoid liquids near water-sensitive objects." In such cases, the learned reward $\mathcal{R}_\theta$ encodes spurious correlations tied to $\mathcal{S}^{\text{train}}$, diverging from the intended reward $\mathcal{R}_{\theta^*}$. This reflects a *reward misspecification* problem where partial observability and limited coverage of the demonstrator's intent lead to learning proxy objectives rather than the true underlying preference [19, 20].

Second, even when the intended preference is correctly recovered within the training domain, applying $\mathcal{R}_\theta$ to novel test states $\mathcal{S}^{\text{test}}$ can produce unreliable behavior. The learned reward function, having been optimized to explain demonstrations involving specific objects and configurations, may respond unpredictably to states containing unseen elements. For instance, a reward function trained on demonstrations involving a *paintbrush* may assign arbitrary values to states containing *molding clay*, even when both objects serve the same functional role in the task. This represents an *out-of-distribution misgeneralization* failure where the learned reward produces unreliable outputs on novel inputs, regardless of the underlying parameterization [3, 33].

These two challenges illustrate why standard reward learning fails to generalize under distribution shift: directly applying $\mathcal{R}_\theta$ to $\mathcal{S}^{\text{test}}$ may assign arbitrary or misleading values to novel states not covered by $\mathcal{S}^{\text{train}}$. Our goal is to enable reward functions to adapt to such shifts without retraining by grounding them in the human's high-level intent.

IV. Generalizing Intent for Flexible Test-Time Rewards (GIFT)

GIFT achieves test-time reward generalization by defining a *state similarity function conditioned on the human's high-level intent*. This function allows us to map novel test states into the training domain and reuse the reward function without retraining. Fig. 2 presents an overview of our framework.

A. The GIFT Framework

Inferring High-Level Intent. To align states with an intent-conditioned similarity function, we first infer the human's underlying intent. We leverage LMs' reasoning capabilities and provide it with: (i) human-preferred demonstrations $\mathcal{D}_H$ and (ii) reference trajectories $\mathcal{D}_{\neg H} := \{\xi_i \sim \Xi\}_{i=1}^N$ (e.g., rollouts from a nominal controller or other non-preferred behavior) in the same scenes. By contrasting these behaviors [18], the LM outputs a natural language summary of the high-level intent that explains the difference between $\mathcal{D}_H$ and $\mathcal{D}_{\neg H}$ (Fig. 2 left). We denote this process as

$$\omega^* \approx \hat{\omega} \triangleq J\big(\mathcal{D}_{\text{H}}, \mathcal{D}_{\neg H}\big),$$

where ω^* is the human's true high-level intent, $\hat{\omega}$ is our estimate, and J is the intent estimator instantiated as an LM.

Intent-Conditioned State Similarity. Having recovered the intent $\hat{\omega}$, we now describe how it guides our notion of state similarity. Our key insight is that states should be treated as similar based on the human's high-level intent rather than low-level visual or language features. We formalize this with an intent-conditioned kernel:

$$\mathcal{K}(s, s' \mid \omega) : \mathcal{S} \times \mathcal{S} \to [0, 1],$$

where higher values denote greater similarity and ω determines what "similar" means for the task. Existing methods in the literature typically define similarity using unimodal representations such as low-level visual [34] or language [35] features. In contrast, we posit that high-level intent ω inherently captures cross-modal semantic relationships—functional goals that transcend specific visual or linguistic instantiations—that better reflect task-relevant similarity.

For example, consider a training scene where a *paintbrush* is packed into a backpack and a test scene with different objects (Fig. 1). A vision-based kernel might show high similarity between a *scrubber* and a *paintbrush* because they are visually alike, while a language-based kernel might instead show a high similarity between a *toothbrush* and a *paintbrush* due to lexical similarity. In contrast, our intent-conditioned kernel recognizes that *molding clay* and *paintbrush* are similar when conditioned on the intent "pack art supplies," preserving the demonstrator's high-level intent.

Different high-level intents induce different measures of similarity over the same set of states. For example, under the intent "pack art supplies," states containing a *paintbrush*, *molding clay*, or *sketchbook* are treated as equivalent, while under "set aside painting supplies," the relevant cluster shifts to *paintbrush*, *watercolor*, and *easel*, with *molding clay* and *sketchbook* now irrelevant. Without conditioning on intent, a robot cannot distinguish which notion of similarity is appropriate and may generalize incorrectly.

To operationalize the kernel $\mathcal{K}$, we use LMs as implicit similarity functions: given the estimated intent $\hat{\omega}$ and descriptions of two states, the LM leverages common-sense priors to decide how similar the states are (Fig. 2 right). This enables us to align novel test states to behaviorally equivalent training states in a way that preserves the underlying human intent, rather than spurious correlations. While our implementation uses binary relevance decisions for simplicity, the framework could naturally extend to continuous similarity by querying the LM to rate relevance on a continuous scale.

Aligning Test States. Once we infer the human's intent $\hat{\omega}$, we use it to align test states to their nearest intent-equivalent training states. Let $\mathcal{S}^{\text{train}}$ be a dataset of states collected from training scenes. For a novel state $s' \in \mathcal{S}^{\text{test}}$, we define an intent-conditioned alignment operator $f_{\hat{\omega}}$:

$$f_{\hat{\omega}}(s') \triangleq \underset{s \in \mathcal{S}^{\text{train}}}{\operatorname{argmax}} \; \mathcal{K}\big(s,\, s' \mid \hat{\omega}\big),$$

which maps[1] s' to the nearest intent-equivalent training state identified by the kernel $\mathcal{K}$. We extend this operator to trajectories $\xi' = (s'_0, \ldots, s'_T)$ by applying it state-wise, aligning each state independently:

$$f_{\hat{\omega}}(\xi') \triangleq \big(f_{\hat{\omega}}(s'_0), \ldots, f_{\hat{\omega}}(s'_T)\big).$$

This produces an aligned trajectory $f_{\hat{\omega}}(\xi')$ whose semantic components correspond to training states in $\mathcal{S}_{\text{train}}$. Intuitively, the alignment operator projects each novel test state into the training domain along intent-relevant dimensions, ensuring that subsequent reward evaluation depends on behaviorally equivalent states.

In our running example, suppose a test scenario involves packing molding clay. Under the inferred high-level intent $\hat{\omega}$ = "pack art supplies," the alignment operator $f_{\hat{\omega}}$ maps molding clay to its training equivalent paintbrush, yielding an aligned trajectory in the training domain whose reward evaluation reflects the intended goal.

Test-Time Reward Generalization. Let $\mathcal{R}_\theta$ be the reward learned once on training data. At test time, we evaluate unseen trajectories by first aligning them into the training domain and then applying $\mathcal{R}_\theta$:

$$\widetilde{\mathcal{R}}_\theta(\xi' \mid \hat{\omega}) \triangleq \mathcal{R}_\theta\big(f_{\hat{\omega}}(\xi')\big).$$

In this formulation, the alignment operator $f_{\hat{\omega}}$ carries the "burden" of test-time generalization, while the learned reward $\mathcal{R}_\theta$ itself remains unchanged. At deployment, planning, or policy selection is performed using the fixed reward on aligned trajectories:

$$\pi^\star = \arg\max_\pi \mathbb{E}_{\xi' \sim \pi}\Big[\widetilde{\mathcal{R}}_\theta(\xi' \mid \hat{\omega})\Big].$$

Because the alignment is intent-conditioned rather than purely visual- or language-conditioned, optimization proceeds over states that are similar in *meaningful, preference-driven* ways, not merely along surface cues.

B. GIFT Implementation Details

Parameterization of the Intent Estimator. We estimate the human's intent, $\hat{\omega}$, via a contrastive LM call,

$$\hat{\omega} = J_{\text{LM}}(\mathcal{D}_{\text{H}}, \mathcal{D}_{\neg\text{H}}),$$

where $\mathcal{D}_{\text{H}}$ are human-provided demonstrations and $\mathcal{D}_{\neg\text{H}}$ are reference trajectories from a nominal controller (e.g., shortest-path planner) in the same scenes. To enable language-based reasoning, we follow Peng et al. [18] and represent trajectories in terms of human-interpretable feature values (e.g., end-effector position, distance to objects, gripper orientation) together with natural-language descriptions of what each feature measures. The LM is thus given both (i) structured numeric feature traces and (ii) semantic descriptions of those features, allowing it to reason abstractly about behavior rather than raw sensor values. We prompt the LM to identify the higher-level intent that distinguishes $\mathcal{D}_{\text{H}}$ from $\mathcal{D}_{\neg\text{H}}$, producing abstract summaries such as "avoid liquids near water-sensitive electronics" rather than instance-specific descriptions like "avoid coffee above laptop."

[1]For simplicity, we write this as operating over full states, though in practice the alignment applies to the semantic components of the state (e.g., object identity) while preserving continuous dimensions such as robot pose.

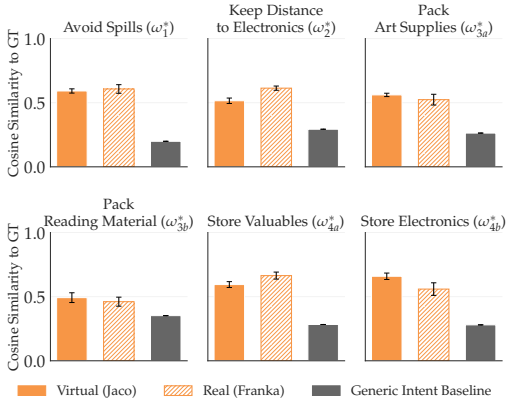

Fig. 3. **Similarity Between LM-Inferred and Ground Truth Intent.** We gave the LM 3 demonstration pairs from a virtual Jaco robot and a real-world Franka robot, and tasked it with deducing the human's intent. We found that LMs produced an acceptable conditioning variable for alignment.

Intent-Guided Alignment. For each test state s', we use the inferred intent $\hat{\omega}$ to determine which training state (if any) is behaviorally equivalent. We implement the alignment operator $f_{\hat{\omega}}$ via an LM call that receives $\hat{\omega}$ as context and is prompted to map the semantic components of s' to their intent-equivalent elements in the training set.

The LM performs a binary relevance judgment for each semantic element of the state (e.g., object identity). If an element is relevant under the intent $\hat{\omega}$, it is mapped to the corresponding training element that fulfills the same intent-relevant role ($\mathcal{K}(s, s' \mid \hat{\omega}) = 1$); otherwise, it is mapped to a *distractor* class ($\mathcal{K}(s, s' \mid \hat{\omega}) = 0$). Formally, this produces a mapping $f_{\hat{\omega}} : \mathcal{S}^{\text{test}} \to \mathcal{S}^{\text{train}} \cup \{\textit{distractor}\}$ conditioned on $\hat{\omega}$ and the scene context, which aligns only the semantic components while preserving continuous ones such as robot pose. Conceptually, this behaves as a *target projection* [29]: at test time, we make new states look like appropriate training states. We use GPT-4o [36] for all LM calls.

Implementation of $\mathcal{R}_\theta$. GIFT is agnostic to the reward parameterization and learning algorithm. In our experiments, we adopt Maximum Entropy IRL [32], which models trajectory probabilities as $p_\theta(\xi) \propto \exp(\mathcal{R}_\theta(\xi))$ and seeks parameters θ that maximize the likelihood of given demonstrations. Following Finn et al. [1], we learn $\mathcal{R}_\theta$ from a set of training demonstrations collected in multiple object configurations.

We parameterize the reward as a linear function over trajectory features: $\mathcal{R}_\theta(\xi) = \theta \cdot \phi(\xi)$, where $\phi : \Xi \to \mathbb{R}^d$ extracts time-aggregated features (e.g. end-effector distance to objects). This linear parameterization enables efficient learning, though GIFT's framework supports more expressive function approximators such as neural networks. Crucially, at deployment the learned reward $\mathcal{R}_\theta$ and feature extractor ϕ remain fixed. All test-time generalization to novel objects and scenes occurs through the intent-conditioned alignment operator $f_{\hat{\omega}}$ (Sec. IV), with no retraining required.

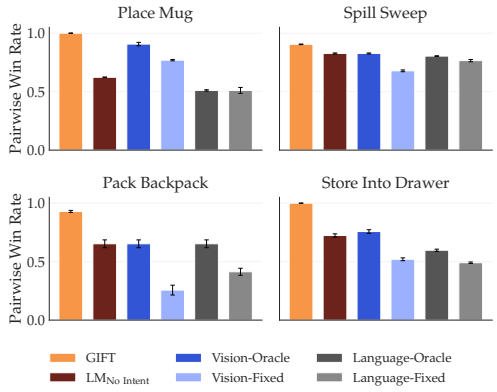

Fig. 4. **Test-Time Pairwise Win Rate.** Across our tasks, the rewards learned via GIFT achieved a higher win rate than all other baselines. These results were aggregated over 250 trajectory pairs per scene, with randomization of various state components along with random sampling from a pool of over 50 unseen objects. Black bars denote standard errors.

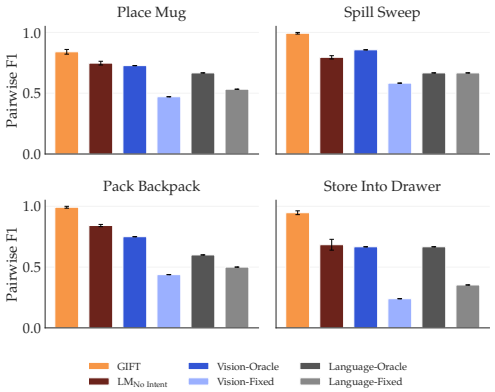

Fig. 5. **Test-Time State Alignment F1 Score.** GIFT achieves a superior F1 score by aligning along intent, reducing confounds from superficial language/visual similarity.

V. Experiments

GIFT is based on the idea that high-level intent enables better test-time generalization compared to low-level vision or language features. We explore the following research question to quantify the benefits of GIFT:

RQ1. Does high-level *intent similarity* lead to better reward performance at test-time compared to low-level *visual* or *language* similarity?

RQ2. When do low-level *visual* or *language* features fail to generalize at test-time?

RQ3. Does GIFT transfer to robots in the physical world?

We conduct experiments to evaluate these research questions using 7-DoF robot arms across four tabletop manipulation tasks in simulation and the real world.

Tasks. Our simulated experiments used a 7DoF Jaco robot arm in the PyBullet simulator [37], and our real-world experiments used the 7-DoF Franka Research Robot. Each task corresponds to a distinct ground-truth high-level intent. The tasks in order of increasing complexity are: **(1) Place Mug.** ω_1^*: *avoid carrying fluids near water-sensitive objects.* **(2) Sweep Spill.** ω_2^*: *sweep paper-based items away from the spill* **(3) Pack Backpack*.** ω_{3a}^*: *pack art supplies*; ω_{3b}^*: *pack reading material* **(4) Store Into Drawer*.** ω_{4a}^*: *store valuables*; ω_{4b}^*: *store electronics*. Tasks 3 and 4 have multiple possible intents to showcase GIFT's ability to recover diverse intent-conditioned similarity functions.

Across our tasks, we used a dataset of over 50 objects in total, which we split into a distinct training set and test set for each task. The test environments contain different objects from the training environments.

Sanity Check. Before evaluating the effectiveness of GIFT, we first verify that GIFT's LM-inferred intents are reasonable estimates of ground-truth intents. We evaluate intent accuracy for each task using the cosine similarity between ground truth intent, ω^*, and the intent estimated by J_{LM}. We perform the intent estimation process ten times for each of the four tasks to calculate the mean and standard error. A baseline intent of predefined, task-relevant goals (e.g., "the human prefers to move objects") serves as a control. To demonstrate flexibility across human preferences, we also evaluated across the two ground-truth intents for **Pack Backpack** (ω_{3a}^* and ω_{3b}^*) and **Store Into Drawer** (ω_{4a}^* and ω_{4b}^*).

Fig. 3 shows cosine similarity between LM-inferred intent and the ground-truth intent for both simulated (Jaco) and physical (Franka) demonstrations. We found that across all tasks, LM-inferred intents showed higher cosine similarity with the ground truth intent than the generic intent baseline, indicating that the intent-prediction step of GIFT generates reasonable intents.

A. Effectiveness Compared to Baselines

Experimental Setup. RQ1 aims to investigate if GIFT's intent-based similarity leads to better test-time reward inference compared to low-level visual or language similarity. We compare GIFT to three baseline similarity methods:

1) **Vision**, $\mathcal{K}(\cdot,\cdot|\omega_{\text{vis}})$, by using the cosine similarity of DINO embeddings [34] over images of scene objects. To mitigate the impact of noise from object appearances, we used Stable Diffusion [38] to generate three object images and compute their average DINO embedding.
2) **Language**, $\mathcal{K}(\cdot,\cdot|\omega_{\text{lang}})$, by using the cosine similarity of BERT embeddings over textual descriptions of the objects in the scene.
3) **LM$_{\text{No Intent}}$**, $\mathcal{K}(\cdot,\cdot|\omega_{\text{LM}})$, parameterized by prompting an LM to directly map components of the test state to the training states as detailed in Sec. IV-B, but without providing an intent learned from demonstrations. This baseline is an ablation of the intention conditioning mechanism of GIFT.

We use the same align-and-score procedure for each similarity method, where ω denotes the alignment conditioning signal (with GIFT using $\omega_{\text{GIFT}} = \hat{\omega}$). We compare $\omega \in \{\omega_{\text{vis}}, \omega_{\text{lang}}, \omega_{\text{LM}}, \omega_{\text{GIFT}}\}$ by aligning a test trajectory with the training set using:

$$\widetilde{\mathcal{R}}_\theta(\xi_{\text{test}}|\,\omega) \triangleq \mathcal{R}_\theta\big(f_\omega(\xi_{\text{test}})\big).$$

To mitigate the impact of noise from sampling LMs, we computed $\omega = J_{\text{LM}}$, then performed the alignment procedure $f_\omega(s')$ for $s' \in \mathcal{S}^{\text{test}}$ ten times. The final aligned state, s', was set as the mode of these repeated runs.

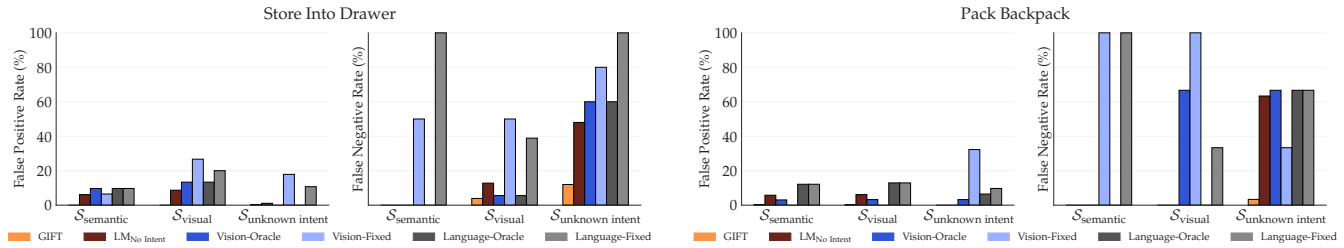

Fig. 6. **FP/FN (%) on the Confounding States, $\mathcal{S}_{\text{conf}}$.** GIFT remains low across categories by using intent-relevance. On the other hand, the oracle baselines merely retune thresholds and trade errors across confounds. Thresholding cannot correct a misaligned similarity signal, and therefore they show high errors. GIFT's ablation, $\text{LM}_{\text{No Intent}}$, is less performant due to not recognizing which test states are intent relevant; leading to a high FN a rate.

We evaluated **RQ1** by generating a mixture of human-preferred and non-preferred test trajectories, randomizing across start locations, goal locations, object types, and object placements. We report average *win rate* [39, 40] for pairs of trajectories (ξ_i, ξ_j), defined as the accuracy of predicting which trajectory the human prefers; a correct prediction counts as $+1$ for a given pair, and ties count as $+0.5$. For each scene, we average over 250 unique trajectories and three random seeds. We formed the following hypothesis for RQ1:

H1. Intent-conditioned similarity achieves a higher win rate than visual, language and $\text{LM}_{\text{No Intent}}$ similarity methods.

Results. Fig. 4 shows pairwise win rate for each of the algorithms that were evaluated across the four tasks. Unseen trajectory pairs show increased win rates for GIFT across all tasks. The largest margins appear in more complex tasks. **Pack Backpack** and **Store Into Drawer** showed 20% improvement over the next best baseline similarity methods. **Place Mug** and **Sweep Spill** achieved 7% improvement over the next best baselines. This result indicates that high-level intent similarity leads to better generalization in unseen test-time scenarios.

B. Failure Modes for Low-level Features

Experimental Setup. RQ2 examines where low-level vision or language similarity methods fail to generalize at test time. To inspect this in detail we compare the ability of different similarity methods to correctly map a test-time object to a ground-truth train-time object.

Each method maps a test state s' to a training state $s \in \mathcal{S}^{\text{train}}$ or flags it as *distractor* via a *similarity threshold* for each of the four tasks. We evaluate the resulting state alignment using three metrics: binary F1, false positives (FP), and false negatives (FN) against a set of ground-truth intent labels. We compare two variants for vision-based and language-based similarities: *oracle* and *non-oracle*. The *oracle variant* selected a threshold that maximize F1 on the test set–requiring access to ground-truth labels–and represents an upper bound. The *non-oracle variant* instead used a fixed threshold selected by computing the average similarity of $\mathcal{S}^{\text{train}}$. A FP represents that a traning object aligned with an irrelevant test object (e.g. *toothbrush* $\rightarrow$ *paintbrush*). A FN represents that a relevant test object is aligned to a *distractor* (e.g. *molding clay* $\rightarrow$ *distractor*).

To facilitate analysis, we define the following datasets of test states $\mathcal{S}_{(\cdot)} \subset \mathcal{S}^{\text{test}}$. $\mathcal{S}_{\text{R}}$ contains a mix of intent-relevant states and straightforward negatives. We additionally define *confounding subsets* for tasks 3 and 4, to test the robustness of our approach to the following confounding factors: **language confounds** $\mathcal{S}_{\text{lang}}$ (i.e., items with similar names, such as *toothbrush* and *paintbrush*), **visual confounds** $\mathcal{S}_{\text{vis}}$ (i.e., items with similar appearances, such as *broomstick* and *paintbrush*), and **unknown-intent confounds** $\mathcal{S}_{\text{unk}}$ (i.e., sets of items that may be grouped in multiple ways depending on high-level intent, such as *metal paintbrush*, *wooden pencil*, and *screwdriver*; two possible intent groupings are *metal items* or *art supplies*). Items may belong to multiple confounding subsets, and we define their union as $\mathcal{S}_{\text{conf}} = \mathcal{S}_{\text{lang}} \cup \mathcal{S}_{\text{vis}} \cup \mathcal{S}_{\text{unk}}$. We report classification error metrics for $\mathcal{S}_{\text{R}} \cup \mathcal{S}_{\text{conf}}$ and $\mathcal{S}_{(\cdot)}$ for tasks 3 and 4 and $\mathcal{S}_{\text{R}}$ for the other two. This experiment was conducted in our simulated environment. Our hypothesis for RQ2 is:

H2. GIFT will produce fewer false positives and false negatives on the different types of confounds than language, visual, or $\text{LM}_{\text{No Intent}}$ baselines.

Results. Fig. 5 shows GIFT's superior alignment performance across tasks, which explains the uplift over baselines in Fig. 4. On tasks 3 and 4, GIFT lowers both FP and FN on the confounded subsets compared to visual/language baselines (Fig. 6). Overall, while the oracle baselines were sometimes robust to one form of confound, they were susceptible to other types of confounds, reducing their generalizability. To understand this behavior, we examined item similarity based on low-level visual and language-based features for the **Pack Backpack *with Art Supplies*** (ω^*_{3a}) intent and the **Store *Valuables* Into Drawer** intent (ω^*_{4a}).

For ω^*_{3a} (Fig. 8), we observe that non art supplies like *toothbrush* and *broomstick* show high language and vision similarity with *paintbrush*, yet items that are art supplies, like *molding clay* and *tablet with stylus*, show low similarity in these low-level visual and language feature spaces. Similarly, for ω^*_{4a} (Fig. 9), we observe that low-level visual and language similarity places the unvaluable *paper ring* as highly similar to the valuable *diamond ring*, whereas other valuable items such as *MacBook Pro* and *iPad Pro* are dissimilar to diamond ring. These results underscore the shortcomings of measuring similarity based on low-level features.

C. Real World Experiments

Environmental Setup. Our simulated experiments generated datasets of human-preferred trajectories given a ground truth reward function, but real-world humans may provide sub-optimal demonstrations. **RQ3** investigates whether LMs can

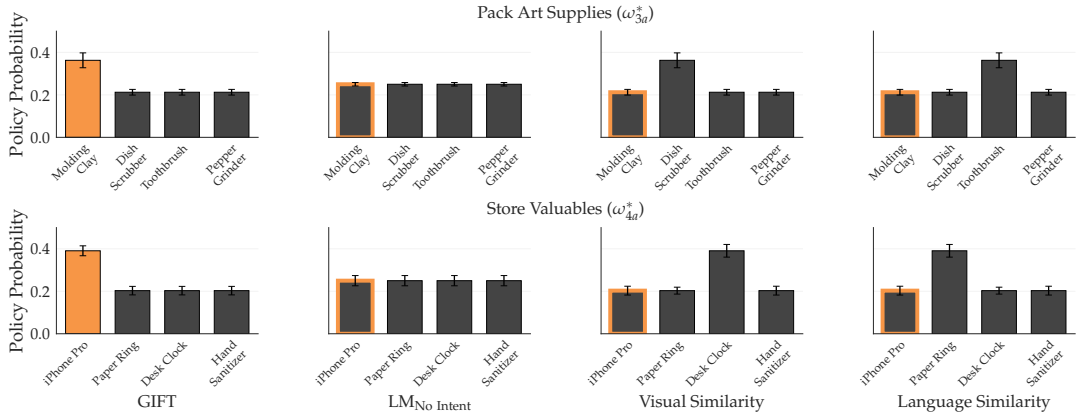

Fig. 7. **Real World Behavior.** By utilizing intent, GIFT planned behavior that was more aligned with the human preferring to pack *art supplies* and store *valuables*. The x-axis labels correspond with test trajectories that have the Franka arm store/pack those unseen items. We randomly sampled sets of four trajectories corresponding to the four items and computed the resulting Boltzmann distribution for each induced reward along with the standard error.

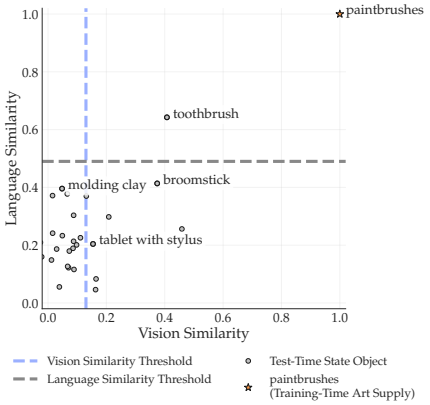

Fig. 8. **Language–Vision Similarity Plot for Pack Backpack *with Art Supplies***. Similarities are measured relative to the training-time anchor (*paintbrushes*) at $(1.0, 1.0)$. Test objects such as the *tablet with stylus* and *molding clay* fall closer to *distractor*, highlighting failure cases where relevant art supplies are confused with unrelated objects. Items like *broomstick* and *toothbrush* are incorrectly pulled toward *paintbrushes*, showing how superficial visual or language similarity can lead to false positives.

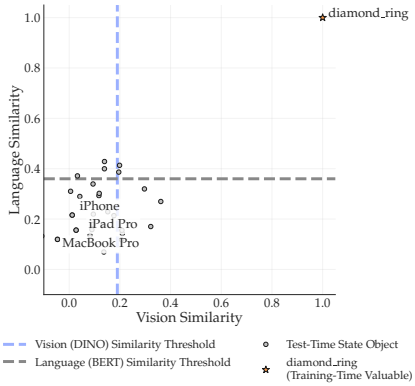

Fig. 9. **Language–Vision Similarity Plot for *Store Valuables***. Similarities are measured relative to the training-time anchor (*diamond ring*) at $(1.0, 1.0)$. Test objects such as the *MacBook Pro* and *iPad Pro* fall close to the anchor, reflecting correct generalization to unseen valuable items. In contrast, irrelevant objects like the *paper ring* cluster further away, indicating proper separation between true valuables and distractors.

infer intent from *real-world* demonstrations from the Franka Emika Panda robot. We recreated two of the simulated tasks using Franka with a set of held-out physical objects. We then evaluated each method by repeatedly sampling a small candidate set of physically executable trajectories–each trajectory manipulating a different held-out object–and converting the resulting aligned-reward scores into a Boltzmann distribution over the candidates. For GIFT, we used intents inferred from Franka's trajectories, $J_{\mathrm{LM}}(\mathcal{D}_{\mathrm{H}}^{\mathrm{Franka}}, \mathcal{D}_{\neg \mathrm{H}}^{\mathrm{Franka}})$, to evaluate if the intents inferred from real-world demonstrations could facilitate robust reward learning. We developed the following hypothesis for RQ3:

H3. Intents inferred from real-world demonstrations will enable GIFT to plan behavior on physical robots that is better aligned with human preferences.

Results. In the Franka domain, we see similar success as in simulation. Planning with high-level intents produced behavior that is more aligned with human preferences on held-out objects than low-level visual features. In Fig. 7, GIFT correctly identified that the human prefers packing the *molding clay* (an art supply) and storing the valuable *iPhone Pro* and correctly treats the confounding held-out objects as distractors. As discussed in RQ2, the baselines succumb to spurious correlations in low-level features.

VI. Conclusion

GIFT reframes test-time reward reuse as an intent-conditioned alignment problem: instead of comparing states along low-level visual or language features, we align unseen states to behaviorally equivalent training states using an intent signal inferred from demonstrations. Across four tabletop tasks and more than 50 unseen objects, GIFT achieved consistent gains in pairwise win rate and lower FP/FN on confounded states versus DINO/BERT baselines and directly using an LM without inferring intent. These improvements were demonstrated in both a simulated JACO robot, aand on a physical 7-DoF Franka robot. In conclusion, by shifting comparisons from low-level visual or language features to higher-level representations of intent, GIFT enables robots to generalize reward functions to unseen test states.

Limitations and Future Work. Limitations point to several next steps. First, performance depends on the quality of the inferred intent; LMs may unpredictably vary the level of abstraction of inferred intents leading to incorrect measures of similarity, and we do not yet calibrate confidence or

abstain when uncertain. Future work includes learning an uncertainty-aware intent kernel, that can refine estimates of ω through additional user queries. Second, our experiments assumed symbolic descriptors for objects and scenes. In practice, these descriptors might be noisy and cause unexplored failure modes. Finally, because LMs can hallucinate and amplify training-set biases, deployment should include guardrails to mitigate potential harms.

REFERENCES

[1] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International conference on machine learning*. PMLR, 2016, pp. 49–58.

[2] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2018.

[3] P. Agrawal, "The task specification problem," in *Conference on Robot Learning*. PMLR, 2022, pp. 1745–1751.

[4] A. S. Chen, S. Nair, and C. Finn, "Learning generalizable robotic reward functions from" in-the-wild" human videos," *arXiv preprint arXiv:2103.16817*, 2021.

[5] Z. Chen, Z. Mandi, H. Bharadhwaj, M. Sharma, S. Song, A. Gupta, and V. Kumar, "Semantically controllable augmentations for generalizable robot learning," *The International Journal of Robotics Research*, p. 02783649241273686, 2024.

[6] A. Bobu, Y. Liu, R. Shah, D. S. Brown, and A. D. Dragan, "SIRL: similarity-based implicit representation learning," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2023, Stockholm, Sweden, March 13-16, 2023*, G. Castellano, L. D. Riek, M. Cakmak, and I. Leite, Eds. ACM, 2023, pp. 565–574.

[7] P. Abbeel and A. Y. Ng, "Apprenticeship Learning via Inverse Reinforcement Learning," in *Proceedings of the International Conference on Machine learning*, 2004.

[8] P. Christiano, J. Leike, T. Brown, *et al.*, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, 2017.

[9] V. Myers, E. Biyik, N. Anari, and D. Sadigh, "Learning multimodal rewards from rankings," in *Conference on robot learning*. PMLR, 2022, pp. 342–352.

[10] S. A. Mehta and D. P. Losey, "Unified learning from demonstrations, corrections, and preferences during physical human–robot interaction," *J. Hum.-Robot Interact.*, vol. 13, no. 3, Aug. 2024.

[11] A. Bobu, A. Bajcsy, J. F. Fisac, and A. D. Dragan, "Learning under misspecified objective spaces," in *Conference on Robot Learning (CoRL)*, 2018.

[12] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning from physical human corrections, one feature at a time," in *Human Robot Interaction*, 2018.

[13] T. R. Sumers, M. K. Ho, R. D. Hawkins, K. Narasimhan, and T. L. Griffiths, "Learning rewards from linguistic feedback," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[14] B. Z. Li, A. Tamkin, N. Goodman, and J. Andreas, "Eliciting human preferences with language models," *arXiv preprint arXiv:2310.11589*, 2023.

[15] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," *arXiv preprint arXiv:2201.07207*, 2022.

[16] Y. J. Ma, W. Liang, G. Wang, *et al.*, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[17] B. Z. Li, W. Chen, P. Sharma, and J. Andreas, "Lampp: Language models as probabilistic priors for perception and action," *arXiv preprint arXiv:2302.02801*, 2023.

[18] A. Peng, A. Bobu, B. Z. Li, I. Sucholutsky, N. Kumar, J. Shah, and J. Andreas, "Adaptive language-guided abstraction from contrastive explanations," in *Conference on Robot Learning*, 2024.

[19] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[20] A. Bobu, A. Peng, P. Agrawal, J. A. Shah, and A. D. Dragan, "Aligning human and robot representations," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2024, Boulder, CO, USA, March 11-15, 2024*, D. Grollman, E. Broadbent, W. Ju, H. Soh, and T. Williams, Eds. ACM, 2024, pp. 42–54.

[21] A. Bobu, M. Wiggert, C. Tomlin, and A. D. Dragan, "Feature expansive reward learning: Rethinking human input," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2021.

[22] ——, "Inducing structure in reward learning by learning features," *The International Journal of Robotics Research*, vol. 41, no. 5, pp. 497–518, 2022.

[23] N. Dennler, S. Nikolaidis, and M. Matarić, "Contrastive learning from exploratory actions: Leveraging natural interactions for preference elicitation," in *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2025, pp. 778–788.

[24] N. Dennler, D. Delgado, D. Zeng, S. Nikolaidis, and M. Matarić, "The rosid tool: Empowering users to design multimodal signals for human-robot collaboration," in *International Symposium on Experimental Robotics*. Springer, 2023, pp. 3–10.

[25] A. Peng, A. Netanyahu, M. K. Ho, T. Shu, A. Bobu, J. Shah, and P. Agrawal, "Diagnosis, feedback, adaptation: A human-in-the-loop framework for test-time policy adaptation," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 27 630–27 641.

[26] A. Forsey-Smerek, J. Shah, and A. Bobu, "Context matters: Learning generalizable rewards via calibrated features," 2025.

[27] A. Peng, A. Bobu, B. Z. Li, T. Sumers, I. Sucholutsky, N. Kumar, T. Griffiths, and J. Shah, "Preference-conditioned language-guided abstraction," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '23, 2024.

[28] Y. Sun, L. Bo, and D. Fox, "Learning to identify new objects," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3165–3172.

[29] P. Pandey, M. Raman, S. Varambally, and P. Ap, "Generalization on unseen domains via inference-time label-preserving target projections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 924–12 933.

[30] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu, "Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation," in *European Conference on Computer Vision*. Springer, 2024, pp. 222–239.

[31] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistics. New York: John Wiley & Sons, 1994.

[32] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[33] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

[34] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[36] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[37] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016–2019.

[38] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021.

[39] P. Crowley, Z. Serlin, T. Paine, M. Mann, M. Benjamin, and C. Belta, "Splash! sample-efficient preference-based inverse reinforcement learning for long-horizon adversarial tasks from suboptimal hierarchical demonstrations," *arXiv preprint arXiv:2507.08707*, 2025.

[40] A. Forsey-Smerek, J. Shah, and A. Bobu, "Context matters: Learning generalizable rewards via calibrated features," *arXiv preprint arXiv:2506.15012*, 2025.

APPENDIX

A. Alignment Error Bound

A natural concern is whether alignment could arbitrarily distort reward evaluation on novel test states. To address this, we show that under a mild smoothness assumption, the error introduced by alignment is bounded by the degree of dissimilarity $1 - \mathcal{K}$.

Suppose the per-step reward is L-Lipschitz with respect to the intent-conditioned distance $1 - \mathcal{K}(\cdot, \cdot \mid \hat{\omega})$:

$$\begin{aligned} & \left|\mathcal{R}_\theta(s) - \mathcal{R}_\theta(s')\right| \;\le\; L\left(1 - \mathcal{K}\big(s,\, s' \mid \hat{\omega}\big)\right) \\ \Rightarrow \quad & \left|\mathcal{R}_\theta(\xi) - \mathcal{R}_\theta(\xi')\right| \;\le\; L\sum_t \left(1 - \mathcal{K}\big(s_t,\, s'_t \mid \hat{\omega}\big)\right). \end{aligned}$$

Then for any test trajectory ξ', the error from evaluating its aligned $f_{\hat{\omega}}(\xi')$ is bounded by the cumulative dissimilarity:

$$\left|\widetilde{\mathcal{R}}_\theta(\xi' \mid \hat{\omega}) - \mathcal{R}_\theta(\xi')\right| \;\le\; L\sum_t \left(1 - \mathcal{K}(f_{\hat{\omega}}(s'_t), s'_t \mid \hat{\omega})\right).$$

This result clarifies when a fixed reward remains meaningful under GIFT: as long as aligned states remain highly similar under the intent-conditioned kernel, the error is small.

The Lipschitz assumption can be satisfied by instantiating $\mathcal{K}$ via continuous similarity measures, such as embedding-based kernels or LLM confidence scores. While our implementation of the kernel is binary, the bound still gives some intuition: states aligned with high confidence ($\mathcal{K} \approx 1$) incur minimal error, while forced alignments to dissimilar states may produce unreliable rewards.

B. Prompt Templates

In this section, we report prompts for the LMs used in our experiments. In our implementation, we had three significant LM calls. **Call 1:** The first LM call corresponds with $\hat{\omega} = J_{\mathrm{LM}}(\mathcal{D}_{\mathrm{H}}, \mathcal{D}_{\neg \mathrm{H}})$. We implemented this by prompting the LM to determine the intent of the user, given the textualized demonstrations. **Call 2:** The second call was made to determine the axis of similarity that is relevant to the intent. Functionally, this call embodied $\mathcal{K}(\cdot, \cdot | \omega_{\text{GIFT}})$, with GIFT using $\omega_{\text{GIFT}} = \hat{\omega}$. **Call 3:** The third call was made to perform the intent-conditioned alignment: $f_{\hat{\omega}}(s') \triangleq \mathrm{argmax}_{s \in \mathcal{S}^{\text{train}}} \mathcal{K}\big(s,\, s' \mid \hat{\omega}\big)$, where s' is a novel state. The operator maps s' to the nearest intent-equivalent training state identified by the kernel $\mathcal{K}$. Note that we implemented $\mathrm{LM}_{\text{No Intent}}$ by forgoing Calls 1 and 2 and using Call 3 with ω_{LM} (common sense LM reasoning without intent).

Call 1:

System Prompt: Infer intent $\hat{\omega} = J_{\mathrm{LM}}(\mathcal{D}_{\mathrm{H}}, \mathcal{D}_{\neg \mathrm{H}})$

```
You are a language model tasked with identifying human-like motivations from robot
    demonstrations. The demonstrations include default and human-preferred trajectories
    performed by a robotic arm. Analyze the trajectories to determine the higher-level
    task goal that the human is trying to accomplish, along with the relevant movement
    features. Consider not only how the movement is executed but also what the objects
    being moved represent and their likely intended use. The motivation should reflect
    why the human is performing the task rather than just how the movement is executed (
    e.g., do not write something like 'Minimize movement variability'). Ensure that your
     motivations are task-specific and ordered by decreasing likelihood. KEEP IN MIND
    THAT WE WANT SPECIFIC motivations/PREFERENCES. Do not answer with something generic
    such as 'The human prefers to avoid objects.' WHAT KIND OF OBJECTS? WHAT IS THE
    MOTIVATION OR GOAL? BE AS SPECIFIC AS POSSIBLE. Ensure that the motivations are
    clearly named, and each relevant feature corresponds to a feature provided in the
    trajectory data. Your motivations should reflect underlying human intent and be
    framed in natural language.

You must produce a single Python dictionary with:
- 3 motivations, ordered by likelihood.
- Each motivation written as a full-sentence human preference (max 2 sentences).
- Each motivation mapped to a list of relevant feature names from the trajectory data.
- You may reuse features across multiple motivations, but you must not create new
    features.

 - Avoid referencing the objects and instead focus on their categories. For example,
     don't write 'the human prefers to avoid scalpels near peaches' instead write
```

```
     something like 'the human prefers to avoid sharp objects near fruit'The 'proximity'
      feature is maximized (set to 1.0) when the object pairs are close to each other,
     and minimized (0.0) when the objects are far apart or one of them is missing from
     the scene. In other words, proximity is the opposite of distance.The response must
     be a Python dictionary as a string, and nothing else.

Example Input:
Default trajectory:
Timestep 0: banana_proximity_to_scalpel: 0.11
Timestep 1: banana_proximity_to_scalpel: 0.42
...
Timestep 60: banana_proximity_to_scalpel: 0.85

Human-preferred trajectory:
Timestep 0: banana_proximity_to_scalpel: 0.45
...
Timestep 60: banana_proximity_to_scalpel: 0.14

Additional features:
- peach_proximity_to_suture
- banana_proximity_to_backpack

Example Output:
{
  "The human prefers to keep edible objects at a safe distance from sharp tools.": [
    "banana_proximity_to_scalpel",
    "peach_proximity_to_suture",
  ],
  "The human prefers to organize items of the same category together for transport.": [
    "banana_proximity_to_scalpel",
    "peach_proximity_to_suture",
  ],
  "The human avoids placing sharp tools near objects that may be handled directly.": [
    "banana_proximity_to_scalpel",
    "peach_proximity_to_suture",
  ]
}

Bad Output Example (What NOT to do):
{
  "The human prefers to minimize distance for efficiency.": ["
      objectA_to_objectB_proximity"],
  "The human wants things to be packed efficiently.": ["objectC_to_objectD_proximity"],
  "The human reduces object-to-bag proximity for optimized storage.": ["
      objectE_to_objectF_proximity"]
}
These responses are too generic, paraphrase the same vague idea, and fail to explain
    what type of objects are involved or why the behavior reflects a specific human
    intent. Do not simply reword 'minimize distance'     you must infer the kind of
    objects and what goal they help accomplish.
```

User Prompt: Infer intent $\hat{\omega} = J_{\mathrm{LM}}(\mathcal{D}_{\mathrm{H}}, \mathcal{D}_{\neg \mathrm{H}})$

```
You are a language model tasked with understanding robotic behaviors to infer human-
    like motivations and identify relevant features associated with each motivation. <
    TASK_STRING> The goal is to generalize these behaviors for unseen tasks. The
    provided data includes default trajectories and human-preferred trajectories.

Approach this task by considering how humans might act in similar scenarios and what
    factors influence their choices. We aim to uncover the underlying reasons behind
    robotic actions and human-like preferences purely from movement patterns. In this
    task, the Jaco robotic arm moves objects in an environment following different
    movement patterns. You will analyze variations in these movement sequences to
    determine possible patterns and associations. Answer the following questions:
```

1) What could be the broader goal of the robotic arm's actions?
2) How do variations in the arms movement provide clues about preferences or motivations?
3) Which features highlight key differences between the human-prefered and default trajectories?
4) How do the properties and functions of the objects influence the trajectory??
5) Based on the motivations of the human, what would be a more general definition for the features?
6) What do the objects across the human-prefered demonstrations have in common? What do they tell you about the human's motivation?

Analyze the trajectories to determine the higher-level task goal that the human is trying to accomplish, along with the relevant movement features. Consider not only how the movement is executed but also what the objects being moved represent and their likely intended use. The motivation should reflect why the human is performing the task rather than just how the movement is executed (e.g., do not write something like 'Minimize movement variability'). Ensure that your motivations are task-specific and ordered by decreasing likelihood. KEEP IN MIND THAT WE WANT SPECIFIC motivations/PREFERENCES. Do not answer with something generic such as 'The human prefers to avoid objects.' WHAT KIND OF OBJECTS? WHAT IS THE MOTIVATION OR GOAL? BE AS SPECIFIC AS POSSIBLE. Ensure that the motivations are clearly named, and each relevant feature corresponds to a feature provided in the trajectory data. Furthermore, the motivations should be ordered by decreasing liklihood. In other words, your first motivation should be the most probable. Give three possible motivations. Avoid referencing the objects and instead focus on their categories. For example, don't write 'the human prefers to avoid scalpels near peaches' instead write something like 'the human prefers to avoid sharp objects near fruit'.Your motivation or preference entry can be up to two sentences. YOU SHOULD NOT CREATE NEW FEATURES, but you may reuse features across motivations.

In addition to listing the motivations and features, you must also identify relevant semantic similarities between the feature sets. Each semantic similarity should describe why a group of features is similar in terms of the inferred motivation. Think in terms of conceptual properties rather than physical similarities. Try to write the semantic similarity as a question.

The output should be structured as two Python dictionary-like objects in the following format:

```
motivations = {
    'motivation or preference 1': ['relevant_feature_1', 'relevant_feature_2',
        ...],
    'motivation or preference 2': ['relevant_feature_3', 'relevant_feature_4',
        ...],
    ...
}

semantic_similarities = {
    'semantic concept 1': ['relevant_feature_1', 'relevant_feature_2', ...],
    'semantic concept 2': ['relevant_feature_3', 'relevant_feature_4', ...],
    ...
}
```

The following are the trajectories:

Default trajectory:
<DEFAULT_TRAJ_BLOCK(v)>
Human-preferred trajectory:
<PREFERRED_TRAJ_BLOCK(v)>

Think carefully about the motivations and their corresponding features before generating the dictionary. Your motivations should not only focus on movement constraints but also consider the role of the objects involved. If an object is consistently moved towards a specific location, infer the purpose of that action rather than just how the movement was performed.
EXTREMELY IMPORTANT. WE WILL SAVE YOUR OUTPUTS AS A PYTHON STRING VARIABLE. WRITE YOUR

```
RESPONSE AS A PYTHON VARIABLE. THIS WILL BE FED INTO ANOTHER LANGUAGE MODEL. DO NOT
USE ANY LATEX OR BOLD FONT Before finalizing your response, ask yourself: 'Am I
describing the task goal or just the way it was executed?' If the answer is only
about movement, reframe it to focus on what the human wanted to achieve with the
task. YOUR RESPONSE SHOULD CONTAIN THE DICTIONARY OBJECT(s) AT THE VERY END
```

Call 2:

System Prompt: Infer similarity axis $\mathcal{K}(\cdot, \cdot \mid \omega_{\text{GIFT}})$, $\omega_{\text{GIFT}} = \hat{\omega}$

```
You are a language model tasked with identifying the conceptual grouping axis that
    unifies a set of objects. These objects come from robot demonstrations, where each
    feature is a distance between two named objects (e.g., 'banana_distance_to_scalpel')
    . The goal is to reason about *why* certain objects appear together given the
    motivation behind a human's actions. Using this reasoning, determine the **semantic
    axis** a  short, natural language question or l a b e l that  explains what unites
    them conceptually.

You will be given a dictionary mapping inferred human motivations to lists of relevant
    feature names. Group those object names based on **shared conceptual properties** (e
    .g., same category, same function, same usage constraint). For each group, label it
    with a **semantic axis**, written as a short yes/no question (e.g., 'is this a fruit
    ?').

Your output must be a single Python dictionary, where:
- Each key is a semantic axis (a question string).
- Each value is a list of object names that belong to that axis.
- Do not invent object names that  d o n t  appear in the features.
- Do not refer to specific demonstrations, environments, or file structures.

You must NOT output any text except the final dictionary. No explanations, no headings,
     no comments.

Be precise and avoid vague or overly general labels like 'is this an object?'.
```

System Prompt: Infer similarity axis $\mathcal{K}(\cdot, \cdot \mid \omega_{\text{GIFT}})$, $\omega_{\text{GIFT}} = \hat{\omega}$

```
You are given a dictionary of inferred human motivations and their relevant features:

<MOTIVATION_DICT_JSON>

Each feature in this dictionary is a distance relationship between two objects, such as
     'objectA_distance_to_objectB'.

Your task is to extract all the **individual objects** that appear in the feature names
     and group them based on shared conceptual properties.
Each group should be labeled with a semantic axis--a short question or phrase that
    identifies what unifies those objects conceptually.
Do not repeat motivation names, do not describe the full feature names, and do not
    refer to specific environments.
Each object should appear in only one axis group.

Your output should be a single Python dictionary mapping each semantic axis to a list
    of object names.

Do not output anything except the final dictionary.
```

Call 3:

System Prompt: Intent-conditioned alignment $f_{\hat{\omega}}(s') \triangleq \text{argmax}_{s \in \mathcal{S}^{\text{train}}} \ \mathcal{K}(s, s' \mid \hat{\omega})$

```
You are an assistant that helps generalize object-level behaviors from a set of seen
    objects to a new set of unseen objects. In each interaction, you are given:
- A list of objects previously seen in the training environment.
- A new list of objects present in the test environment.
- Sometimes, additional context in the form of inferred human motivations and
    conceptual similarity axes.

Your job is to map each object in the unseen state to either:
1. A specific object in the seen state that it meaningfully corresponds to based on
    human intent and conceptual relevance.
2. The special token 'distractor' if the object is not relevant to the human's
    motivations or does not align with any known object roles.

You will always output a single Python dictionary, and nothing else. The dictionary
    must map **every** unseen object to either a seen object or 'distractor'. You are
    allowed to map numerous unseen elements to the same seen elements.

You may encounter two types of input conditions:
- **Full Information**: You are given motivations and semantic similarity groupings.

- **Blind**: No prior motivations or similarity axes are given.

### REQUIRED FORMAT:
- Do not include markdown or code blocks.
- Do not add commentary or explanation.
- Only output a Python dictionary as plain text.

If you cannot find a strong match for an unseen object, assign it as 'distractor'. BE
    SURE TO USE THE SEMANTIC SIMILARITIES (if they are given to you). This will help you
     generalize behaviors beyond the class of objects which the human demonstrations
    contain. The idea is to generalize behaviors which means surface-level similarities
    are not always relevant.
```

User Prompt: Intent-conditioned alignment $f_{\hat{\omega}}(s') \triangleq \text{argmax}_{s \in \mathcal{S}^{\text{train}}} \ \mathcal{K}(s, s' \mid \hat{\omega})$

```
[If we are testing GIFT:]
EXTREMELY IMPORTANT, USE THE INFERRED MOTIVATIONS AND SEMANTIC SIMILARITIES WHEN MAKING
     YOUR DECISIONS. There are two dictionaries, the first gives you the motivations and
     their relevant features that were inferred from the trajectories. The second
    dictionary tells you the relevant semantic similarities with respect to the
    motivations:
<BACKGROUND_CALL>

[Else (We are testing LM No Intent):]
VERY IMPORTANT: Ignore future references to the mention of inferred semantic
    similarities or motivations in the rest of this message. Use your common sense
    reasoning.

The previously seen state contains the following objects:
<SEEN_OBJECT_LIST>

The new unseen state contains the following objects:
<UNSEEN_OBJECT_LIST>

Your task is to generalize the features and roles of objects in the unseen state based
    on the inferred motivations and their relevant features provided above. Additionally
    , use the provided semantic similarities to ensure that mappings prioritize
    conceptually relevant properties rather than just surface-level object categories.
    This will allow the unseen demonstrations to be analyzed using the existing reward
    model. Specifically:
1. Propose replacements or generalizations for objects that have no direct equivalent,
```

```
    considering the motivations, their features, and their relevant similarities.
2. Identify objects in the unseen state that directly correspond to objects in the
    following seen state.
<SEEN_OBJECT_LIST>

Explain how the new unseen state could be altered directly to make it compatible as
    feature arguments for the seen domain/reward model. Your explanation should include
    specific changes to the state structure or feature definitions, taking into account
    the relevant semantic similarities.

Finally, produce a Python dictionary that maps objects in the unseen state to objects
    in the seen state. This dictionary should match the format used by our `
    spoof_unseen_trajectories` function. For example:

{
  "objA": "objX",
  "objB": "distractor",
  "objC": "objZ"
  ...
}

Mappings should be justified using the inferred semantic similarities and motivations (
    if given), ensuring that objects are matched based on relevant conceptual properties
     rather than just visual or categorical resemblance. If an unseen object directly
    corresponds to a known one in the seen domain, map it accordingly. Otherwise, mark
    it as 'distractor'.

EXTREMELY IMPORTANT. WE WILL SAVE YOUR OUTPUTS AS A PYTHON STRING VARIABLE. WRITE YOUR
    RESPONSE AS A PYTHON VARIABLE. THIS WILL BE FED INTO ANOTHER LANGUAGE MODEL. DO NOT
    USE ANY LATEX OR BOLD FONT
```