VLP: Vision-Language Preference Learning for Embodied Manipulation

Runze Liu[1], **Chenjia Bai**[2†], **Jiafei Lyu**[1], **Shengjie Sun**[1], **Yali Du**[3], **Xiu Li**[1†]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Institute of Artificial Intelligence (TeleAI), China Telecom, [3]King's College London

Abstract

Reward engineering is one of the key challenges in Reinforcement Learning (RL). Preference-based RL effectively addresses this issue by learning from human feedback. However, it is both time-consuming and expensive to collect human preference labels. In this paper, we propose a novel **V**ision-**L**anguage **P**reference learning framework, named **VLP**, which learns a vision-language preference model to provide feedback for embodied manipulation tasks. To achieve this, we define three types of language-conditioned preferences and construct a vision-language preference dataset, which contains versatile implicit preference orders. The model learns to extract language-related features, and then serves as a predictor in various downstream tasks. The policy can be learned according to the annotated labels via reward learning or direct policy optimization. Extensive empirical results on simulated embodied manipulation tasks demonstrate that our method provides accurate preferences and generalizes to unseen tasks and unseen language instructions, outperforming the baselines by a large margin and shifting the burden from continuous, per-task human annotation to one-time, per-domain data collection.

1 Introduction

Reinforcement Learning (RL) has made great achievements recent years, including board games (Silver et al., 2017, 2018), autonomous driving (Kiran et al., 2021; Zhou et al., 2021), and robotic manipulation (Kober et al., 2013; Andrychowicz et al., 2020; Chen et al., 2022; Sun et al., 2024b). However, one of the key challenges to apply RL algorithms is reward engineering. First, designing an accurate reward function requires large amount of expert knowledge. Second, the agent might hack the designed reward function (Hadfield-Menell et al., 2017), obtaining high returns without completing the task. Also, it is difficult to obtain reward functions for subjective human objectives.

To address the above issues, a variety of works leverage Vision-Language Models (VLMs) to provide multi-modal rewards for downstream policy learning (Nair et al., 2023; Ma et al., 2023a; Rocamonde et al., 2024). However, the reward labels produced in these works are often of high variance and noisy (Ma et al., 2023a). Preference-based RL is more promising way that learns from human preferences over trajectory pairs (Christiano et al., 2017; Lee et al., 2021). A line of works (Christiano et al., 2017; Kim et al., 2023) learns a reward model from human labels and then optimizes the policy according to the reward model. Another line of works (Hejna and Sadigh, 2023; Hejna et al., 2024) directly optimizes the policy according to the labels.

However, preference-based RL requires either querying a large number of expert feedback online (Lee et al., 2021; Park et al., 2022) or a labeled offline preference dataset (Kim et al., 2023; Hejna et al., 2024), which is quite time-consuming and expensive. As the reasoning abilities of Large Language Models (LLMs) improve significantly (OpenAI, 2024; Liu et al., 2025), previous methods propose to use LLMs to provide labels (Wang et al., 2025), but the generated labels are not guaranteed to be accurate and it is assumed to have access to the environment information that is usually hard to obtain in practical scenarios.

In this paper, we propose a **V**ision-**L**anguage **P**reference alignment framework, named **VLP**, to provide preference feedback for video pairs given language instructions. Specifically, we collect a video dataset from various policies under augmented language instructions, which contains implicit preference relations based on the trajectory optimality and the vision-language correspondence.

† Corresponding authors: Chenjia Bai (baicj@chinatelecom.cn), Xiu Li (li.xiu@sz.tsinghua.edu.cn)

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 28428–28444
November 4-9, 2025 ©2025 Association for Computational Linguistics

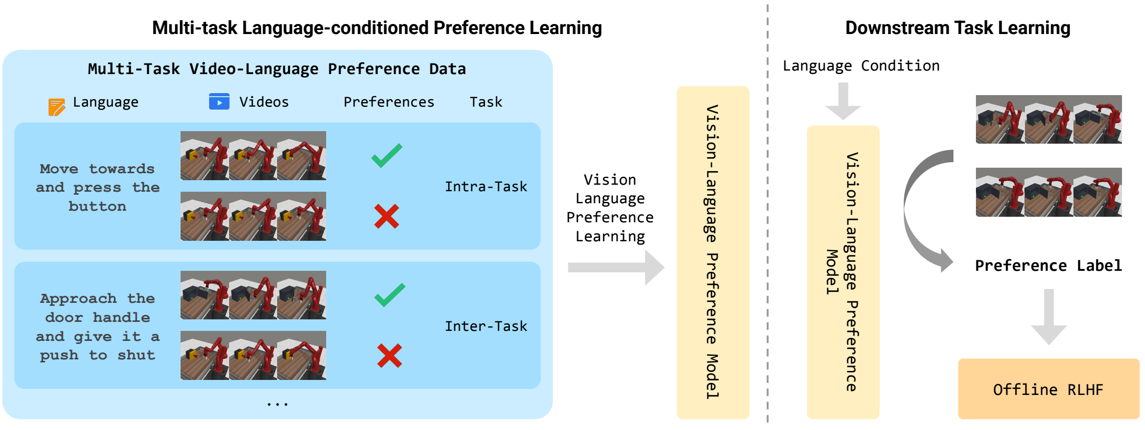

Multi-task Language-conditioned Preference Learning
Multi-Task Video-Language Preference Data
Language
Videos
Preferences
Task
Move towards and press the button
Intra-Task
Approach the door handle and give it a push to shut
Inter-Task
...
Vision Language Preference Learning
Vision-Language Preference Model
Downstream Task Learning
Language Condition
Vision-Language Preference Model
Preference Label
Offline RLHF

Figure 1: The overall framework of our method.

Then, we define language-conditioned preferences and propose a novel vision-language alignment architecture to learn a trajectory-wise preference model for labeling, which consists of a video encoder, a language encoder, and a cross-modal encoder to facilitate vision-language alignment. The preference model is optimized by intra-task and inter-task preferences that are implicitly contained in the dataset. During inference, VLP provides labels for target tasks and can even generalize to unseen tasks and unseen language instructions. The labels given by VLP are employed for various downstream preference optimization algorithms to facilitate policy learning.

In summary, our contributions are as follows: (i) We propose a vision-language preference alignment framework, which learns a vision-language preference model to provide feedback for embodied manipulation tasks. (ii) We propose language-conditioned preferences and construct a vision-language preference dataset, which contains videos with language instructions and implicit language-conditioned relations. (iii) Extensive empirical results on simulated embodied manipulation tasks demonstrate that our method provides accurate labels and generalizes to unseen tasks and unseen language instructions, outperforming the baselines by a large margin.

2 Related Work

Vision-Language Models for Reinforcement Learning. Our work is related to the literature on VLM rewards and preferences for embodied manipulation tasks (Radford et al., 2021; Nair et al., 2023; Ma et al., 2023a; Rocamonde et al., 2024; Wang et al., 2024; Liu et al., 2024a). These methods can be divided into three categories: (i) representation-based pre-training, (ii) zero-shot inference, and (iii) downstream fine-tuning. For representation-based approaches, R3M (Nair et al., 2023) is pre-trained on the Ego4D dataset (Grauman et al., 2022) to learn useful representations for downstream tasks. LIV (Ma et al., 2023b), which extends VIP (Ma et al., 2023b) to multi-modal representations, is pre-trained on EpicKitchen dataset (Damen et al., 2018), and can also be fine-tuned on target domain. For zero-shot inference methods, VLM-RM (Rocamonde et al., 2024) utilizes CLIP (Radford et al., 2021) as zero-shot vision-language rewards. RoboCLIP (Sontakke et al., 2023) uses S3D (Xie et al., 2018), which is pre-trained on HowTo100M dataset (Miech et al., 2019), as video-language model to compute vision-language reward with a single demonstration (a video or a text). FuRL (Fu et al., 2024a) leverages pre-trained VLMs to provide rewards for RL agents. TemporalOT (Fu et al., 2024b) uses optimal transport to provides rewards by aligning trajectories with demonstration trajectories. RL-VLM-F (Wang et al., 2024) leverages Gemini-Pro (Team et al., 2023) and GPT-4V (OpenAI, 2023) for zero-shot preference feedback. CriticGPT (Liu et al., 2024a) is the representative method of (iii), which fine-tunes multimodal LLMs on a instruction-following dataset, and utilizes the tuned model to provide preference feedback for downstream policy learning. VLP differs from these approaches that we do not suffer from burdensome training of (i) and (iii), showing great computing efficiency. And VLP learns more embodied manipulation knowledge compared with VLMs pre-trained on natural image-text data.

Preference-based Reinforcement Learning. Preference-based RL is a promising framework for aligning the agent with human values. However, feedback efficiency is a crucial challenge in preference-based RL, with multiple recent studies striving to tackle. To improve the feedback efficiency, previous works focus on unsupervised pre-training (Lee et al., 2021), estimating pseudo labels using reward confidence (Park et al., 2022), employing reward uncertainty for exploration (Liang et al., 2022), Q-function-aware reward learning (Liu et al., 2022; Bai et al., 2025), and meta-learning to pre-train the reward model (Hejna III and Sadigh, 2023). Recently, a growing number of studies focus on offline preference-based RL with the population of offline RL (Levine et al., 2020; Kostrikov et al., 2022; Lyu et al., 2024, 2025). Several works explore learning policies from preferences without a reward function (Kang et al., 2023; Hejna and Sadigh, 2023; Hejna et al., 2024). For network architecture, PT (Kim et al., 2023) introduces a Transformer-based architecture for reward modeling, while FTB (Zhang et al., 2024) leverages a diffusion model for better trajectory generation. To reduce preference labeling costs, PEARL (Liu et al., 2024b) proposes cross-task preference alignment to transfer preference labels between tasks. VLP addresses the labeling cost by learning a vision-language preference model via vision-language alignment, thereby providing generalized preferences to novel tasks.

3 Background

Problem Setting. We formulate the RL problem as a Markov Decision Process (MDP) (Sutton and Barto, 2018) represented as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, p_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and $p_0 : \mathcal{S} \to [0, 1]$ is the initial state distribution. At timestep t, the agent observes a state s_t and selects an action a_t based on a policy $\pi(a_t|s_t)$. Then, the agent receives a reward r_t from the environment, and the agent transits to s_{t+1} according to the transition function. The agent's goal is to find a policy that maximizes the expected cumulative reward $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$. In multi-task setting, for a task $\mathcal{T} \sim p(\mathcal{T})$, a task-specific MDP is represented as $\mathcal{M}^{\mathcal{T}} = (\mathcal{S}^{\mathcal{T}}, \mathcal{A}, \mathcal{P}^{\mathcal{T}}, \mathcal{R}^{\mathcal{T}}, \gamma, p_0^{\mathcal{T}})$.

Preference-based RL. Preference-based RL differs from RL in that it is assumed to have no access to the ground-truth rewards (Christiano et al., 2017; Lee et al., 2021). In preference-based RL, human teachers provide preference labels over trajectory pairs, and a reward model is learned from these preferences. Formally, a trajectory segment σ of length H is represented as $\{s_1, a_1, \dots, s_H, a_H\}$ and a segment pair is (σ^1, σ^2). The preference label $y \in \{0, 1, 0.5\}$ denotes which segment is preferred, where 0 indicates σ^1 is preferred (i.e., $\sigma^1 \succ \sigma^2$), 1 indicates σ^2 is preferred (i.e., $\sigma^2 \succ \sigma^1$), and 0.5 represents two segments are equally preferred. Previous preference-based RL approaches construct a preference predictor with the reward model $\widehat{r}_\psi$ via Bradley-Terry model (Bradley and Terry, 1952):

$$P_\psi[\sigma^1 \succ \sigma^2] = \frac{\exp\left(\sum_{t=1}^{H} \widehat{r}_\psi(s_t^1, a_t^1)\right)}{\sum_{k=1}^{2} \exp\left(\sum_{t=1}^{H} \widehat{r}_\psi(s_t^k, a_t^k)\right)}, \tag{1}$$

where $P_\psi[\sigma^1 \succ \sigma^2]$ denotes the probability that σ^1 is preferred over σ^2 predicted by current reward model $\widehat{r}_\psi$. Assume we have a dataset with preference labels $\mathcal{D} = \{(\sigma^1, \sigma^2, y)\}$, the reward learning process can be formulated as a classification problem using cross-entropy loss (Christiano et al., 2017):

$$\begin{aligned}\mathcal{L}_{\text{ce}} = - \mathop{\mathbb{E}}_{(\sigma^1,\sigma^2,y)\sim\mathcal{D}} \Big[&(1-y) \log P_\psi[\sigma^1 \succ \sigma^2] \\ &+ y \log P_\psi[\sigma^2 \succ \sigma^1] \Big].\end{aligned} \tag{2}$$

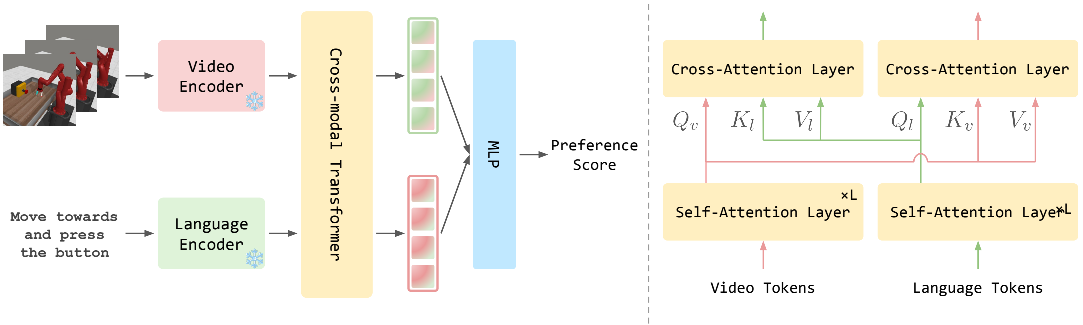
By optimizing Eq. (2), the reward model is aligned with human preferences, providing reward signals for policy learning.

4 Method

In this section, we first present the overall framework of VLP, including model architecture and the vision-language preference dataset. Then, we introduce language-conditioned preferences and the detailed algorithm for vision-language preference learning, which learns a trajectory-wise preference model via vision-language preference alignment.

4.1 Model and Dataset

The goal of VLP is to learn a generalized preference model capable of providing preferences for novel embodied tasks. To achieve this, the preference model receives videos and language as inputs, where videos serve as universal representations of agent trajectories and language act as universal and flexible instructions. To obtain high-quality representations of these two modalities, we utilize CLIP (Radford et al., 2021), which is pre-trained



(a) Model Architecture

(b) Cross-modal Transformer

Figure 2: (a) Trajectory videos and language instruction are fed into the preference model to obtain a trajectory-wise preference score. (b) The cross-modal transformer obtains language-related video features and video-related language features by cross-attention mechanism.

on extensive image-text data, as our video and language encoders. The extracted video and language features are fed into to a cross-modal transformer for cross-modal attention interaction to capture video features associated with the language and language features related to the video. These features are subsequently utilized for predicting preference scores in vision-language preference learning. The overall framework is illustrated in Figure 2.

Model Architecture. A video v is represented as a sequence of video frames, i.e., $v = \{v_1, v_2, \dots, v_{|v|}\}$, where $v_i \in \mathbb{R}^{H\times W\times 3}$, H and W are the height and width of each video frame, and $|v|$ denotes the number of video frames. The video encoder is employed to obtain the video tokens $z = \{z_1, z_2, \dots, z_{|v|}\}$, where $z_i \in \mathbb{R}^{M\times D_v}$, $M = H/p \times W/p$ is the number of visual tokens, p is the patch size of CLIP ViT, and D_v is the dimension of the visual tokens. Given language input l, the language tokens $u \in \mathbb{R}^{N\times D_l}$ are obtained via the language encoder, where N is the number of language tokens, and D_l is the dimension of the language tokens.

With video tokens z and language tokens u, a cross-modal encoder is employed to facilitate multi-modal feature learning, making tokens of different modalities fully fuse with each other. Video tokens and language tokens are separately inputted into the self-attention layers. Then, utilizing the output video tokens as queries and the output language tokens as keys and values, the cross-attention layer, as shown in Figure 2(b), generates language features that are closely related to the input video. Similarly, the cross-attention layer produces language-related video features. The multi-modal tokens are averaged along the first dimension and then concatenated as $w \in \mathbb{R}^{D_w}$, where $D_w = D_v + D_l$. These new tokens are fed into the final Multi-layer Perceptron (MLP) for vision-language preference prediction, outputting a trajectory-level preference score.

Vision-Language Preference Dataset. While there are open-sourced embodied datasets with language instructions (Mu et al., 2023), there lacks a multi-modal preference dataset for generalized preference learning. To this end, we construct MTVLP, a multi-task vision-language preference dataset built upon Meta-World (Yu et al., 2020). To that end, we consider the following aspects: (i) trajectories of various optimality levels should be collected to define clear preference relations within each task; (ii) each trajectory pair should be accompanied with a corresponding language instruction for learning language-conditioned preferences.

It is easy to describe the optimality of expert trajectories and random trajectories because it is easy to understand the agent's behavior in these trajectories. However, it is challenging to define a medium-level policy without explicit rewards. Fortunately, we find most robot tasks can be divided into multiple stages, where each stage completes a part of the overall task. Thus, we define a medium-level policy as successfully completing half of the stages of the task. For example, we divided the task of *opening the drawer* into two subtasks: (i) moving and grasping the drawer handle and (ii) pulling the drawer handle. A medium-level policy only completes the first subtask.

We leverage a scripted policy for each task to roll out trajectories of three optimality levels: expert, medium, and random. For expert-level trajectories, we employ the scripted policy with Gaussian noise to interact. The medium-level trajectories are also collected with the scripted policy but are terminated when the half of subtasks are completed. As for random-level trajectories, actions are randomly sampled from a uniform distribution during rollout. For the corresponding language, we obtain diverse language instructions to improve the generalization abilities of our model by aligning one video with multiple similar language instructions. Following Adeniji et al. (2023), we query GPT-4V (OpenAI, 2023) to generate language instructions with various verb structure examples and synonym nouns of each task. Details of collecting trajectories and language instructions for each task are shown in Appendix C.

4.2 Vision-Language Preference Alignment

Language-conditioned Preferences. Previous RLHF methods define trajectory preferences according to a single task goal. However, this unimodal approach struggles to generalize to new tasks due to its rigid preference definition. In contrast, by integrating language as a condition, we can establish more flexible preference definitions. Consider two videos, v_1^1 and v_2^1, along with a language instruction l^1 from task $\mathcal{T}^1$, and another video v^2 paired with a language instruction l^2 from task $\mathcal{T}^2$. We categorize three forms of language-conditioned preferences: Intra-Task Preference (ITP), Inter-Language Preference (ILP), and Inter-Video Preference (IVP), as shown in Table 1.

Table 1: Three types of language-conditioned preferences.

Type	Videos	Language	Criterion
ITP	$v_1^1, v_2^1 \sim \mathcal{T}^1$	$l^1 \sim \mathcal{T}^1$	optimality
ILP	$v_1^1, v_2^1 \sim \mathcal{T}^1$	$l^2 \sim \mathcal{T}^2$	equally preferred
IVP	$v_1^1 \sim \mathcal{T}^1, v_1^2 \sim \mathcal{T}^2$	$l^1 \sim \mathcal{T}^1$	$v_1^1 \succ v_2^1 \| l^1$

ITP corresponds to the conventional case of preference relation within the same task (Christiano et al., 2017), where the videos and language instructions are from the same task, and the preference relies on the optimality of videos w.r.t. the task objective. ILP considers a scenario where the language instruction differs from the task of the videos. Thus, both videos are equally preferred under this language condition. IVP deals with preferences of two videos from different tasks, with the language instruction from either task. It is straightforward to define the preference that the vision-language come from the same task is preferred to the other pair.

This framework allows for the establishment of universal and adaptable preference relations, wherein videos from the same task can yield varying preference labels depending on the language condition. Notably, even random trajectories paired with language instructions from a specific task is preferred to expert trajectories from other tasks.

Vision-Language Preference Learning. With language-conditioned preferences defined above, we further introduce our vision-language preference learning algorithm. We aim to develop a vision-language preference model that predicts the preferred video under specific language conditions. However, directly inputting two videos and a language instruction into the model would affect computational efficiency. So, we consider the conventional way to learn from preference labels (Christiano et al., 2017), i.e., first constructing preference predictors via Bradley-Terry model (Bradley and Terry, 1952). Previous work has revealed the advantages of learning a preference model over a reward model (Zhang et al., 2024). Based on these insights, our proposed preference model $f_\psi(v|l)$ takes a video and a language instruction as inputs and outputs a scalar preference score. Then the preference label can be obtained by comparing preference scores of two videos with a given language instruction, i.e., $v_1 \succ v_2|l$ if $f_\psi(v_1|l) > f_\psi(v_2|l)$.

Given videos v_1 representing σ_1 and v_2 representing σ_2, the language-conditioned preference distribution $P_\psi[v_1 \succ v_2|l]$ is the probability that σ_1 is preferred over σ_2 under the condition l:

$$P_\psi[v_1 \succ v_2|l] = \frac{\exp\big(f_\psi(v_1|l)\big)}{\sum_{k=1}^{2} \exp\big(f_\psi(v_k|l)\big)}. \quad (3)$$

Given tasks $\mathcal{T}^1$ and $\mathcal{T}^2$, we consider the following objectives aligned with language-conditioned preference relations: **(a)** Learning Intra-Task Preference: Within the same task, the video that better follows l should be preferred, analogous to previous RLHF objective (Christiano et al., 2017); **(b)** Learning Inter-Language Preference: Under the language condition of task $\mathcal{T}^2$, videos from task $\mathcal{T}^1$ are equally preferred; **(c)** Learning Inter-Video Preference: Under the language condition of task $\mathcal{T}^1$, the video from $\mathcal{T}^1$ is preferred over the video from $\mathcal{T}^2$.

During vision-language preference learning, a task $\mathcal{T}$ is sampled from all training tasks, followed by sampling a minibatch $\{v_1^b, v_2^b, v^{\neq b}, l^b, l^{\neq b}, y^{\text{ITP}}, y^{\text{ILP}}, y^{\text{IVP}}\}^{1:B}$. Here, the superscript ^{b} indicates data sampled from task $\mathcal{T}$ in the minibatch, while $^{\neq b}$ denotes data from other tasks. $y^{\text{ITP}}, y^{\text{ILP}}, y^{\text{IVP}}$ are the ground-truth labels of ITP, ILP, and IVP, respectively. The total loss of vision-language preference learning is as follows:

$$\mathcal{L}_{\text{ce}} = -\sum_{b \in B} \Big[\underbrace{\text{CE}\left(P_\psi[v_1^b \succ v_2^b | l^b], y^{\text{ITP}}\right)}_{\text{(a)}} + \lambda_1 \underbrace{\text{CE}\left(P_\psi[v_1^b \succ v_2^b | l^{\neq b}], y^{\text{ILP}}\right)}_{\text{(b)}} + \lambda_2 \underbrace{\text{CE}\left(P_\psi[v_1^b \succ v^{\neq b} | l^b], y^{\text{IVP}}\right)}_{\text{(c)}} + \lambda_2 \underbrace{\text{CE}\left(P_\psi[v_2^b \succ v^{\neq b} | l^b], y^{\text{IVP}}\right)}_{\text{(c)}} \Big], \tag{4}$$

where $\text{CE}(\cdot, \cdot)$ is the cross-entropy loss, and λ_1 and λ_2 are balance weights of learning ILP and IVP. By optimizing Eq. (4), the vision-language preference model outputs trajectory-level preference scores aligned with the language-conditioned preference relations.

The difference between VLP and prior methods is not only the preference loss, but the overall framework of language-conditioned preference learning. Unlike previous preference learning methods that rely on single-modality inputs (e.g., trajectory preferences defined solely based on task goals), our framework integrates language as a flexible condition to define preferences, offering greater generalization capacity.

The inclusion of ILP and IVP in our training data serves critical roles in enhancing the generalization and robustness of our model. ILP allows our model to learn to disregard language variations when they do not impact the preference outcomes, thus training the model to focus on task-relevant features rather than linguistic discrepancies. On the other hand, IVP facilitates the model's ability to generalize across different tasks by learning to associate videos with their corresponding task-specific language instructions effectively. This capability is crucial when the model encounters new tasks or language contexts, as it must discern relevant from irrelevant information to make accurate preference predictions. By training with both ILP and IVP, our model learns a more holistic understanding of the task space, which not only improves its performance on seen tasks but also enhances its adaptability to new, unseen tasks or variations in task descriptions, as evidenced by our experimental results where the model demonstrated generalization capabilities.

5 Experiments

In this section, we evaluate VLP on Meta-World (Yu et al., 2020) and ManiSkill2 (Gu et al., 2023) benchmark and aim to answer the following questions:

- **Q1:** How do VLP labels compare with scripted labels in offline RLHF? (Section 5.2)
- **Q2:** How does VLP compare with other vision-language rewards approaches? (Section 5.3)
- **Q3:** How does VLP generalize to unseen tasks and language instructions? (Section 5.4)

5.1 Setup

Implementation Details. For Meta-World (Yu et al., 2020), we evaluate VLP on the 5 test tasks, including *Button Press*, *Door Close*, *Drawer Close*, *Faucet Close*, and *Window Open*, while the other 45 tasks of Meta-World are used as training tasks and this train-test proportion follows standard ML45 split. For ManiSkill2 (Gu et al., 2023), we selected 11 tasks and allocated 8 for training and 3 for testing, balancing available resources with the need to test cross-task transfer. For VLP implementation, we use the pre-trained ViT-B/16 CLIP model (Radford et al., 2021) as our video encoder and language encoder. The weights of learning ILP and IVP in Eq. (4) are $\lambda_1 = 0.1$, $\lambda_2 = 0.5$, respectively. Additional hyperparameters of VLP are detailed in Table 8 in Appendix A. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

5.2 How do VLP labels compare with scripted labels in offline RLHF?

Baselines. We evaluate VLP by combining it with recent offline RLHF algorithms: (i) **P-IQL** (Preference IQL), which first learns a reward model from preferences and then learns a policy via IQL (Kostrikov et al., 2022); (ii) **IPL** (Hejna and Sadigh, 2023), which learns a policy without reward learning by aligning the Q-function with preferences; (iii) **CPL** (Hejna et al., 2024), which directly learns a policy using a contrastive objective with maximum entropy principle, eliminating the need for reward learning and RL.

Evaluation. For each evaluation task, we train each RLHF method with scripted labels (Christiano et al., 2017; Lee et al., 2021) and VLP labels

Table 2: Success rate of RLHF methods with scripted labels and VLP labels. The results are reported with mean and standard deviation across five random seeds. The result of VLP is shaded and is **bolded** if it exceeds or is comparable with that of RLHF approaches with scripted labels. MTVLP Acc. and Scripted Acc. denote the accuracy of preference labels inferred by VLP compared with MTVLP labels and scripted labels, respectively.

Task	**P-IQL**	**P-IQL+VLP**	**IPL**	**IPL+VLP**	**CPL**	**CPL+VLP**	**Scripted Acc.**	**MTVLP Acc.**
Button Press	72.6 ± 7.1	**90.1** ± 3.9	50.6 ± 7.9	**56.0** ± 1.4	74.5 ± 8.2	**83.9** ± 11.8	93.0	99.0
Door Close	79.2 ± 6.3	**79.2** ± 6.3	61.5 ± 9.4	**61.5** ± 9.4	98.5 ± 1.0	**98.5** ± 1.0	100.0	100.0
Drawer Close	49.3 ± 4.2	**64.9** ± 2.9	64.3 ± 9.6	63.2 ± 4.7	45.6 ± 3.5	**57.5** ± 14.3	96.0	96.0
Faucet Close	51.1 ± 7.5	**51.1** ± 7.5	45.4 ± 8.6	**45.4** ± 8.6	80.0 ± 2.9	**80.0** ± 2.9	100.0	100.0
Window Open	62.4 ± 6.4	**69.7** ± 6.8	54.1 ± 6.7	**61.4** ± 8.6	91.6 ± 1.7	**99.1** ± 1.1	98.0	100.0
Average	62.9	**71.0**	55.2	**57.5**	78.0	**83.8**	97.4	99.0

Table 3: Preference label accuracy of VLP on ManiSkill2 test tasks.

Task	**VLP Acc.**
LiftCube-v0	100.0
OpenCabinetDoor-v1	100.0
PushChair-v1	93.8
Average	97.9

(denoted as **+VLP**), respectively. Scripted preference labels mean the preference labels computed based on the ground-truth rewards (Christiano et al., 2017; Lee et al., 2021). The number of preference labels is set to 100 for all tasks. The evaluation is conducted over 25 episodes every 5000 steps. Following (Hejna et al., 2024), we average the results of 8 neighboring evaluations and take the maximum value among all averaged values as the result. Detailed hyperparameters of RLHF algorithms can be found in Appendix A.

To examine the effects of VLP on more challenging tasks, we also conduct experiments on ManiSkill2 (Gu et al., 2023) benchmark. We leverage *MoveBucket-v1*, *OpenCabinetDrawer-v1*, *PegInsertionSide-v0*, *PickCube-v0*, *PickSingleEGAD-v0*, *PlugCharger-v0*, *StackCube-v0*, and *TurnFaucet-v0* as training tasks and evaluate VLP on *LiftCube-v0*, *OpenCabinetDoor-v1*, *PushChair-v1* tasks.

Results. Experimental results in Table 2 demonstrate that VLP labels predicted by our trained model are accurate compared with scripted labels and labels computed from preferences in MTVLP. With VLP labels, the performance of P-IQL+VLP and CPL+VLP is comparable with, and in some cases, outperforms that with scripted labels on all evaluation tasks. We hypothesize that the ground-truth reward of *Button Press, Drawer Close* and *Window Open* may not accurately represent the task goal, which is also shown in previous works (Xie et al., 2024; Ma et al., 2024; Sun et al., 2024a). Table 3 summarizes the average VLP label accuracy on the three test tasks compared to scripted labels and the results demonstrate the strong generalization capabilities of VLP. It is noteworthy that By aligning video and language modalities through preference relations with language as conditions, the predicted VLP labels directly represent how the video reflects the language instruction. Therefore, our method provides more accurate and preference labels and can generalize to unseen tasks.

5.3 How does VLP compare with other vision-language rewards approaches?

Baselines. We compare VLP with the following VLM rewards baselines: (i) **R3M** (Nair et al., 2023), which pre-trains visual representation by time-contrastive learning and vision-language alignment; (ii) **VIP** (Ma et al., 2023b), which provides generalized visual reward and representation for downstream tasks via value-implicit pre-training; (iii) **LIV** (Ma et al., 2023a), which learns vision-language rewards and representation via multi-modal value pre-training; (iv) **CLIP** (Radford et al., 2021), which pre-trains by aligning vision-language representation on a large-scale image-text pairs dataset; (v) **VLM-RM** (Rocamonde et al., 2024), which provides zero-shot VLM rewards based on CLIP (Radford et al., 2021). VLM-RM includes a hyperparameter α, which controls the goal-baseline regularization strength. In the evaluation, we denote the variant of $\alpha = 0.0$ as **VLM-RM (0.0)** and the variant of $\alpha = 1.0$ as **VLM-RM (1.0)**. (vi) **RoboCLIP** (Sontakke et al., 2023), which provides zero-shot VLM rewards using pre-trained video-language models and a single demonstration (a video demonstration or a

Table 4: Success rate of VLP (i.e., P-IQL trained with VLP labels) against IQL with VLM **rewards**. The results are reported with mean and standard deviation across five random seeds. The result of VLP is shaded and the best score of all methods is **bolded**.

Task	**R3M**	**VIP**	**LIV**	**CLIP**	**VLM-RM (0.0)**	**VLM-RM (1.0)**	**VLP**
Button Press	10.1 ± 2.3	68.4 ± 6.4	56.3 ± 1.9	59.5 ± 6.1	60.3 ± 6.1	64.3 ± 8.4	**90.1** ± 3.9
Door Close	70.9 ± 5.3	74.8 ± 9.5	43.3 ± 3.2	43.6 ± 3.9	45.8 ± 8.5	41.1 ± 3.4	**79.2** ± 6.3
Drawer Close	46.6 ± 2.6	70.4 ± 4.5	61.8 ± 5.7	69.4 ± 4.1	69.4 ± 4.5	**73.5** ± 5.4	64.9 ± 2.9
Faucet Close	25.7 ± 23.6	40.9 ± 8.0	42.2 ± 6.3	59.6 ± 7.5	**60.1** ± 5.1	33.7 ± 15.3	51.1 ± 7.5
Window Open	39.0 ± 6.6	42.7 ± 11.3	33.8 ± 6.4	26.4 ± 2.0	23.9 ± 1.9	23.7 ± 4.9	**69.7** ± 6.8
Average	38.5	59.4	47.5	51.7	51.9	47.3	**71.0**

Table 5: Success rate of VLP (i.e., P-IQL trained with VLP labels) against P-IQL with VLM **preferences** (denoted with prefix **P-**). The results are reported with mean and standard deviation across five random seeds. The result of VLP is shaded and the best score of all methods is **bolded**.

Task	**P-R3M**	**P-VIP**	**P-LIV**	**P-CLIP**	**P-VLM-RM (0.0)**	**P-VLM-RM (1.0)**	**RoboCLIP**	**VLP**
Button Press	84.7 ± 5.8	41.2 ± 3.9	61.7 ± 5.1	62.9 ± 6.2	72.8 ± 5.0	44.2 ± 4.2	56.4 ± 7.3	**90.1** ± 3.9
Door Close	72.4 ± 11.5	54.2 ± 13.8	67.9 ± 6.3	53.3 ± 10.3	57.6 ± 2.9	45.7 ± 7.6	47.6 ± 6.7	**79.2** ± 6.3
Drawer Close	59.6 ± 6.5	63.0 ± 3.7	45.5 ± 10.4	63.4 ± 3.2	62.7 ± 3.0	49.2 ± 6.9	**73.0** ± 6.2	64.9 ± 2.9
Faucet Close	58.0 ± 4.5	51.1 ± 7.5	**62.3** ± 7.2	60.2 ± 10.4	57.3 ± 7.0	51.3 ± 9.5	62.1 ± 6.3	51.1 ± 7.5
Window Open	27.3 ± 5.0	50.2 ± 1.8	22.2 ± 18.1	28.4 ± 3.2	33.2 ± 5.4	20.7 ± 2.3	28.1 ± 4.6	**69.7** ± 6.8
Average	60.4	51.9	51.9	53.6	56.7	42.2	53.4	**71.0**

Table 6: The correlation coefficient of VLM rewards with ground-truth rewards and VLP labels with scripted preference labels. Larger correlation means the predicted values are more correlated with the ground-truth.

Task	**R3M**	**VIP**	**LIV**	**CLIP**	**VLM-RM (0.0)**	**VLM-RM (1.0)**	**VLP**
Button Press	0.313	0.204	-0.281	0.127	0.153	-0.082	**0.581**
Door Close	0.735	0.125	0.600	-0.309	-0.152	-0.492	**1.000**
Drawer Close	-0.106	0.043	0.052	-0.151	-0.137	-0.031	**0.438**
Faucet Close	0.676	0.851	0.563	-0.301	-0.291	0.084	**1.000**
Window Open	0.411	**0.725**	-0.568	0.336	0.405	-0.333	0.571
Average	0.406	0.390	0.073	-0.060	-0.005	-0.171	**0.718**

language description) of the task.

Evaluation. We first evaluate our method with the VLM baselines by directly training IQL with VLM **rewards**. VLP is tested by training P-IQL with VLP labels, and the experimental setting of our method is the same as that of Section 5.2. We further compare VLP with VLM **preferences**, i.e., using predicted VLM rewards to compute preference labels for a fair comparison with our method. However, RoboCLIP obtains scalar trajectory-level rewards and we utilize them as trajectory return for preference labels calculation. Implementation details of IQL and VLM baselines can be found in Appendix A.

Results. Results in Table 4 show that our method exceeds the VLM baselines that train IQL from VLM rewards by a large margin with an average success rate of **71.0**. As shown in Table 5, when the VLM baselines are trained with preferences computed by VLM rewards, our method still surpasses the baselines. We further compute the preference label accuracy of each method, detailed in Table 15. The results show that VLP exceeds VLM baselines, which do not learn relative relations of reward values.

Reward / Preference Correlation. To further investigate the advantages of VLP model compared with VLM reward models, we compare the correlation between VLM rewards with ground-truth rewards and VLP labels with scripted preference labels. Results in Table 6 indicate that VLP labels exhibit a stronger correlation with scripted labels compared with VLM rewards.

Table 7: The generalization abilities of our method on 5 unseen tasks with different types of language instructions. Acc. denotes the accuracy of preference labels inferred by VLP compared with ground-truth labels.

Metric	**Seen**	**Phrase**	**Description**	**Correct Color**	**Incorrect Color**
ITP Acc. (↑)	97.4	95.8	97.0	97.0	97.0
IVP Acc. (↑)	91.7	90.5	91.9	91.9	91.8
ILP Loss (↓)	0.705	0.704	0.704	0.705	0.705
Average Loss (↓)	0.555	0.554	0.558	0.556	0.557

5.4 How does VLP generalize to unseen tasks and language instructions?

Evaluation. We first evaluate how accurate 3 kinds of VLP labels are on the test tasks. We test the preference model with phrases, descriptions, and correct and incorrect object colors. Since the label of ILP is 0.5 (i.e., two segments are equally preferred), we compute ILP loss with the (b) term in Eq. (4), i.e., $-\sum_{b\in B} \mathrm{CE}\left(P_\psi[v_1^b \succ v_2^b | l^{\neq b}], y^{\mathrm{ILP}}\right)$. Performance of ITP and IVP are measured with accuracy. Experimental details can be found in Appendix A.

Results. Table 7 shows that VLP generalizes to unseen language instructions on unseen tasks with high ITP and IVP accuracy and low ILP loss. However, using unseen phrases as language conditions leads to a performance drop, while unseen descriptions have a slight negative impact on ITP but a positive impact on IVP and ILP. We think the reason is that phrases contain insufficient information about completing the task, while descriptions contain enough task information. VLP generalizes well with suitable language information of tasks. Also, VLP exhibits strong generalization abilities on color.

6 Conclusion

In this paper, we propose VLP, a vision-language preference learning framework providing generalized preference feedback for embodied manipulation tasks. In our framework, we learn a vision-language preference model via proposed language-conditioned preference relations from the collected vision-language preference dataset. Experimental results on multiple simulated robotic manipulation tasks demonstrate that our method exceeds previous VLM rewards approaches and predicts accurate preferences compared with scripted labels. The results also show our method generalizes well to unseen tasks and unseen language instructions.

7 Limitations

In this paper, we focus on providing preferences for robotic manipulation tasks. First, VLP is limited to the tasks that can be specified via videos and language instructions. While this covers a wide range of robotic tasks, certain tasks cannot be fully expressed via videos and language, such as complex assembly tasks requiring intricate spatial reasoning. Consequently, the risk of predicting incorrect preferences grows for complex tasks that are difficult to express. Second, if the language instruction lacks sufficient information of the task goal, the risk of giving incorrect labels still grows, as shown in Table 7. We do not see any potential risks of our work.

Acknowledgments

This work was supported by the STI 2030-Major Projects under Grant 2021ZD0201404, the National Natural Science Foundation of China (Grant No. 62306242), the Young Elite Scientists Sponsorship Program by CAST (Grant No. 2024QNRC001), and the Yangfan Project of the Shanghai (Grant No. 23YF11462200).

References

Ademi Adeniji, Amber Xie, Carmelo Sferrazza, Younggyo Seo, Stephen James, and Pieter Abbeel. 2023. Language reward modulation for pretraining reinforcement learning. *arXiv preprint arXiv:2308.12270*.

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

Fengshuo Bai, Runze Liu, Yali Du, Ying Wen, and Yaodong Yang. 2025. RAT: Adversarial attacks on deep reinforcement agents for targeted behaviors. In *Annual AAAI Conference on Artificial Intelligence (AAAI)*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Zongqing Lu, Stephen McAleer, Hao Dong, Song-Chun Zhu, and Yaodong Yang. 2022. Towards human-level bimanual dexterous manipulation with reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 5150–5163.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and 1 others. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736.

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. 2024a. FuRL: Visual-language models as fuzzy rewards for reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 14256–14274. PMLR.

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. 2024b. Robot policy learning with temporal optimal transport reward. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, and 1 others. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. 2023. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations (ICLR)*.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. 2017. Inverse reward design. *Advances in neural information processing systems (NeurIPS)*, 30.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh. 2024. Contrastive preference learning: Learning from human feedback without reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Joey Hejna and Dorsa Sadigh. 2023. Inverse preference learning: Preference-based RL without a reward function. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Donald Joseph Hejna III and Dorsa Sadigh. 2023. Few-shot preference learning for human-in-the-loop rl. In *Conference on Robot Learning (CORL)*, pages 2014–2025. PMLR.

Yachen Kang, Diyuan Shi, Jinxin Liu, Li He, and Donglin Wang. 2023. Beyond reward: Offline preference-guided policy optimization. In *International Conference on Machine Learning (ICML)*, volume 202, pages 15753–15768.

Changyeon Kim, Jongjin Park, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2023. Preference transformer: Modeling human preferences using transformers for rl. In *International Conference on Learning Representations (ICLR)*.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926.

Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2022. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations (ICLR)*.

Kimin Lee, Laura M Smith, and Pieter Abbeel. 2021. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning (ICML)*, volume 139, pages 6152–6163.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. 2022. Reward uncertainty for exploration in preference-based reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Jinyi Liu, Yifu Yuan, Jianye Hao, Fei Ni, Lingzhi Fu, Yibin Chen, and Yan Zheng. 2024a. Enhancing robotic manipulation with ai feedback from multimodal large language models. *arXiv preprint arXiv:2402.14245*.

Runze Liu, Fengshuo Bai, Yali Du, and Yaodong Yang. 2022. Meta-Reward-Net: Implicitly differentiable reward learning for preference-based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 22270–22284.

Runze Liu, Yali Du, Fengshuo Bai, Jiafei Lyu, and Xiu Li. 2024b. PEARL: Zero-shot cross-task preference alignment and robust reward learning for robotic manipulation. In *International Conference on Machine Learning (ICML)*, volume 235, pages 30946–30964. PMLR.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*.

Jiafei Lyu, Xiaoteng Ma, Le Wan, Runze Liu, Xiu Li, and Zongqing Lu. 2024. SEABO: A simple search-based method for offline imitation learning. In *International Conference on Learning Representations (ICLR)*.

Jiafei Lyu, Mengbei Yan, Zhongjian Qiao, Runze Liu, Xiaoteng Ma, Deheng Ye, Jing-Wen Yang, Zongqing Lu, and Xiu Li. 2025. Cross-domain offline policy adaptation with optimal transport and dataset constraint. In *International Conference on Learning Representations (ICLR)*.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. 2023a. LIV: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 23301–23320. PMLR.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations (ICLR)*.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. 2023b. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 2630–2640.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2023. EmbodiedGPT: Vision-language pre-training via embodied chain of thought. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. 2023. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pages 892–909. PMLR.

OpenAI. 2023. GPT-4V(ision) system card.

OpenAI. 2024. Learning to reason with llms.

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2022. SURF: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. 2024. Vision-language models are zero-shot reward models for reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, and 1 others. 2018. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, and 1 others. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. 2023. Roboclip: One demonstration is enough to learn robot policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 55681–55693.

Shengjie Sun, Runze Liu, Jiafei Lyu, Jing-Wen Yang, Liangpeng Zhang, and Xiu Li. 2024a. A large language model-driven reward design framework via dynamic feedback for reinforcement learning. *arXiv preprint arXiv:2410.14660*.

Shengjie Sun, Jiafei Lyu, Lu Li, Jiazhe Guo, Mengbei Yan, Runze Liu, and Xiu Li. 2024b. Enhancing visual generalization in reinforcement learning with cycling augmentation. In *International Conference on Artificial Neural Networks*, pages 397–411. Springer.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ruiqi Wang, Dezhong Zhao, Ziqin Yuan, Ike Obi, and Byung-Cheol Min. 2025. PrefCLM: Enhancing preference-based reinforcement learning with crowd-sourced large language models. *IEEE Robotics and Automation Letters (RAL)*, 10(3):2486–2493.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. 2024. RL-VLM-F: Reinforcement learning from vision language foundation model feedback. In *International Conference on Machine Learning (ICML)*, volume 235, pages 51484–51501. PMLR.

Amber Xie, Youngwoon Lee, Pieter Abbeel, and Stephen James. 2023. Language-conditioned path planning. In *Conference on Robot Learning (CORL)*, volume 229 of *Proceedings of Machine Learning Research*, pages 3384–3396. PMLR.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. 2024. Text2Reward: Reward shaping with language models for reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. 2020. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, volume 100, pages 1094–1100. PMLR.

Zhilong Zhang, Yihao Sun, Junyin Ye, Tian-Shuo Liu, Jiaji Zhang, and Yang Yu. 2024. Flow to better: Offline preference-based reinforcement learning via preferred trajectory generation. In *International Conference on Learning Representations (ICLR)*.

Ming Zhou, Jun Luo, Julian Villella, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Montgomery Alban, Iman Fadakar, Zheng Chen, and 1 others. 2021. Smarts: An open-source scalable multi-agent rl training school for autonomous driving. In *Conference on robot learning (CoRL)*, volume 155, pages 264–285. PMLR.

A Experimental Details

A.1 Tasks

The tasks used in the experiments are from the test tasks of MTVLP. For Meta-World, ML45 defines a standard split with 45 training tasks and 5 testing tasks and we directly follow to this split proportion. For ManiSkill2, due to resource constraints, we collected 11 tasks and split them into 8 training and 3 testing tasks, ensuring diversity in task difficulty. To evaluate the method across varying levels of difficulty, we choose tasks with different levels of complexity. For example, PushChair-v1 is highly challenging, while LiftCube-v0 is easier, as shown in Fig. 2 of Xie et al. (2024). This diversity demonstrates the robustness of our method across tasks of varying difficulty. Figure 3 shows the tasks used in Meta-World and the task descriptions are as follows:

- Button Press: The goal of the robotic arm is to press the button. The initial position of the arm is randomly sampled.
- Door Close: The goal of the robotic arm is to close the door. The initial position of the arm is randomly sampled.
- Drawer Close: The goal of the robotic arm is to close the drawer. The initial position of the arm is randomly sampled.
- Faucet Close: The goal of the robotic arm is to close the faucet. The initial position of the arm is randomly sampled.
- Window Open: The goal of the robotic arm is to open the window. The initial position of the arm is randomly sampled.

A.2 Implementation Details

We implement our method based on the publicly released repository of LAPP[1] and the overall framework is illustrated in Figure 2. Following LAPP (Xie et al., 2023), we use a pre-trained ViT-B/16 CLIP (Radford et al., 2021) model as our video encoder and language encoder. To achieve efficient learning, we uniformly sample 8 frames to represent each video. The detailed hyperparameters of our method are shown in Table 8. Training a VLP model takes about 6 hours on a single NVIDIA RTX 4090 GPU with 12 CPU cores and 120 GB memory, without costly pre-training process like VLM reward or VLM preference methods (Nair et al., 2023; Ma et al., 2023b,a).

Table 8: Hyperparameters of VLP.

Hyperparameter	**Value**
Prediction head	$(512, 256)$
Number of self-attention layers	2
Number of attention heads	16
Batch size	16
Optimizer	Adam
Learning rate	3e-5
Learning rate decay	cosine decay
Weight decay	0.1
Dropout	0.1
Number of epochs	15k
Number of negative samples	4
Number of video frames	8
Weight of ILP loss λ_1	0.1
Weight of IVP loss λ_1	0.5

IQL, P-IQL, IPL and CPL are implemented based on the official repository of CPL and IPL.[2][3] The hyperparameters of offline RL and RLHF algorithms are listed in Table 9, Table 10, and Table 11. For the inference of VLP labels, we first use K-means clustering to divide the trajectories of each test task into 2 sets, following Liu et al. (2024b). Then we sample 100 trajectory segments of length 50 from each set to construct segment pairs and predict preference labels of these pairs with trained VLP model. Training RL and RLHF algorithms take about 10 minutes using a single NVIDIA RTX 4090 GPU with 6 CPU cores and 60 GB memory.

The tests with phrases, descriptions, and correct and incorrect object colors were designed to evaluate the model's robustness to variations and errors in language instructions. Specifically:

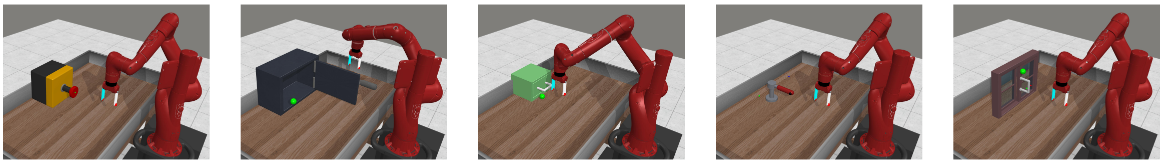
- Phrases: Simple instructions like "close door."
- Descriptions: More detailed instructions describing the same objective.
- Incorrect object colors: Intentional mismatches, such as referring to a "green button" when the button is red.

These perturbations are introduced by modifying the language input during testing to assess the model's generalization to noisy or misaligned language instructions.

[1] https://github.com/amberxie88/lapp

[2] https://github.com/jhejna/cpl

[3] https://github.com/jhejna/inverse-preference-learning



(a) Button Press (b) Door Close (c) Drawer Close (d) Faucet Close (e) Window Open

Figure 3: Five simulated robotic manipulation tasks used for experimental evaluation.

Table 9: Shared hyperparameters.

Hyperparameter	Value
Network architecture	(256, 256)
Optimizer	Adam
Learning rate	1e-4 (CPL), 3e-4 (IQL, IPL and P-IQL)
Batch size	64
Discount	0.99
Dropout	0.25
Training steps	100000
Segment length	50 (RLHF)
Number of queries	100 (RLHF)
Temperature	0.3333 (IQL, IPL and P-IQL)
Expectile	0.7 (IQL, IPL and P-IQL)
Soft target update rate	0.005 (IQL, IPL and P-IQL)

Table 10: Hyperparameters of CPL.

Hyperparameter	Value
Temperature	0.1
Contrastive bias	0.5
BC weight	0.0
BC steps	10000

Table 11: Hyperparameters of IPL and P-IQL.

Hyperparameter	Value
Regularization weight (IPL)	0.5
Reward learning steps (P-IQL)	30

For VLM methods, R3M, VIP, LIV, VLM-RM, and RoboCLIP are implemented based on their official repositories.[4][5][6][7][8] The CLIP baseline is a variant of VLM-RM and is implemented based on the code of VLM-RM. The language inputs of the VLM baselines except are as listed in Table 12. R3M, LIV, CLIP, and RoboCLIP only require the target column as language inputs, while VLM-RM additionally needs a baseline as a regularization term. R3M requires an initial image and we use the first frame of each trajectory as the initial image, while VIP requires a goal image for VLM rewards inference and we use the last frame of expert videos.

Table 12: Language inputs used for evaluating VLM baselines on the test tasks.

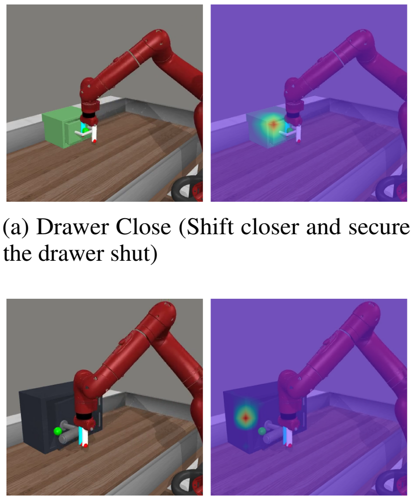
Task	Target	Baseline (for VLM-RM)
Button Press	press button	button
Door Close	close door	door
Drawer Close	close drawer	drawer
Faucet Close	turn faucet left	faucet
Window Open	move window left	window

B Additional Experimental Results

Attention Map Visualization. We further analyze VLP by visualizing the attention maps of the cross-attention. Results in Figure 4 show that regions of the objects related to language instructions exhibit high attention weights. For example, in the Drawer Close task, our vision-language preference model specifically focuses on whether the drawer is closed, with the attention map highlighting the edges of the drawer to monitor its position and similarly for Door Close task. These observations demonstrate that our vision-language preference model effectively learns to guide language tokens

[4] https://github.com/facebookresearch/r3m
[5] https://github.com/facebookresearch/vip
[6] https://github.com/penn-pal-lab/LIV
[7] https://github.com/AlignmentResearch/vlmrm
[8] https://github.com/sumedh7/RoboCLIP

to attend to relevant regions in the videos and illustrate the effectiveness of our cross-attention mechanism in bridging vision and language modalities for precise task understanding.



(a) Drawer Close (Shift closer and secure the drawer shut)

(b) Door Close (Direct the gripper to the door handle and press to seal it)

Figure 4: Attention map visualization of *Drawer Close* and *Door Close*. The language instruction is shown at the bottom of each subfigure.

Effects of λ_1 and λ_2. λ_1 and λ_2 in Eq. (4) control the strength of ILP and IVP learning, respectively. To investigate how λ_1 and λ_2 influence VLP, we conduct experiments by vary λ_1 across $\{0.0, 0.1, 0.5\}$ and λ_2 across $\{0.0, 0.5, 1.0\}$. Results in Table 13 show that the performance of VLP drops with too small or too large λ_1. Meanwhile, without IVP learning (i.e., $\lambda_2 = 0$), the performance of IVP and ILP significantly decreases. We speculate that IVP is crucial for language-conditioned preference learning. Without IVP learning, the learned VLP model degenerates into a vanilla preference model without language as conditions.

Table 13: Accuracy of VLP labels with different loss. Acc. denotes the accuracy of preference labels inferred by VLP compared with ground-truth labels.

λ_1	λ_2	ITP Acc. (↑)	IVP Acc. (↑)	ILP Loss (↓)	Avg. Loss (↓)
0.0	0.5	95.4	74.1	0.728	0.618
0.5	0.5	85.8	74.7	0.702	0.578
0.1	0.0	96.2	63.0	0.775	0.646
0.1	1.0	95.8	96.5	0.699	0.554
0.1	0.5	97.4	91.7	0.705	0.555

Effects of Preference Dataset Size. We investigate how the preference dataset size influences our method. We conduct additional experiments by varying the dataset size across $\{50\%, 75\%, 100\%\}$. Results in Table 14 indicate that the performance of VLP downgrades as the dataset size decreases.

Table 14: Accuracy of VLP labels with different data size. Acc. denotes the accuracy of preference labels inferred by VLP compared with ground-truth labels.

Data	ITP Acc. (↑)	IVP Acc. (↑)	ILP Loss (↓)	Avg. Loss (↓)
50%	94.2	89.6	0.699	0.557
75%	95.2	89.7	0.707	0.555
100%	97.4	91.7	0.705	0.555

Preference Label Accuracy. To compare the relative relation of VLM rewards with VLP, we compute the preference label accuracy of each method. The accuracy is measured by comparing the predicted preference labels with scripted preference labels. The results in Table 15 show that VLP exceeds the VLM baselines by a large margin, demonstrating VLM rewards do not capture the relative reward relationship.

Different VLMs/LLMs for Language Instruction Generation. To see the influence of different language model on our method, we we conduct additional experiments using instructions from less capable model, such as GPT-3.5 and open-source Llama-3.1-8B-Instruct. We observe that generating diverse language instructions does not necessarily require strong VLMs like GPT-4V, even open-source Llama-3.1-8B-Instruct can accomplish this job since the language model is prompted with a diverse set of examples, following LAMP (Adeniji et al., 2023). The results in Table 16 show that the model's performance is relatively stable across different LLMs.

C Details of MTVLP Collection

For the 50 robotic manipulation tasks in Meta-World (Yu et al., 2020), we divide *Button Press*, *Door Close*, *Drawer Close*, *Faucet Close*, and *Window Open* as test tasks and the other 45 tasks as train tasks. For each task, we leverage scripted policies of Meta-World (Yu et al., 2020) to collect trajectories. For expert trajectories, we add Gaussian noise sampled from $\mathcal{N}(0, 0.1)$. For medium trajectories, we utilize the `near_object` flag returned by each task to determine whether the first subtask is completed and add Gaussian noise sampled from $\mathcal{N}(0, 0.5)$. For random trajectories, the actions are sampled from uniform distribution $\mathcal{U}[0, 1]$. We collect 32 trajectories of each type of trajectory for

Table 15: Preference label accuracy of VLP against VLM baselines. The accuracy of our method is shaded and the best score of all methods is **bolded**.

Task	**R3M**	**VIP**	**LIV**	**CLIP**	**VLM-RM (0.0)**	**VLM-RM (1.0)**	**RoboCLIP**	**VLP**
Button Press	91.0	40.0	62.0	53.0	62.0	41.0	46.0	**93.0**
Door Close	98.0	57.0	97.0	49.0	59.0	10.0	61.0	**100.0**
Drawer Close	66.0	49.0	39.0	66.0	65.0	58.0	43.0	**96.0**
Faucet Close	98.0	**100.0**	97.0	38.0	25.0	65.0	63.0	**100.0**
Window Open	72.0	88.0	16.0	81.0	88.0	16.0	49.0	**98.0**
Average	85.0	66.8	62.2	57.4	59.8	38.0	52.4	**97.4**

Table 16: Preference label accuracy of VLP with language instructions generated by different VLMs/LLMs.

Task	**GPT-4V**	**GPT-3.5**	**Llama-3.1-8B-Inst.**
Button Press	93.0	93.0	91.0
Door Close	100.0	100.0	98.0
Drawer Close	96.0	96.0	97.0
Faucet Close	100.0	100.0	100.0
Window Open	98.0	99.0	99.0
Average	97.4	97.6	97.0

each task, resulting in a total of 4800 trajectories for all tasks. We query GPT-4V (OpenAI, 2023) to generate language instructions by the prompt containing an example of generating diverse language instructions, an example of generating synonym nouns, task name, task instruction, and an image rendering the task. The detailed prompt we used is shown in Table 17.

D Discussions

How does different train-test split influence VLP? We conduct experiments on the Meta-World ML45 benchmark, training the vision-language preference model on its training tasks and evaluating on its test tasks. We compute VLP label accuracy by comparing VLP label with scripted preference labels. The results shown in Table 18 demonstrate the strong generalization capability of our method on unseen tasks in ML45. This reinforces the robustness and adaptability of our framework regardless of task split.

E License For Artifacts

Meta-World, IPL, CPL, R3M, LIV, VLM-RM, and CLIP models are licensed under the MIT License. VIP is licensed under the CC BY-NC 4.0 License. For ManiSkill2, all rigid body tasks are covered by fully permissive licenses (e.g., Apache-2.0), while the associated assets are licensed under CC BY-NC 4.0. It should be noted that the official repositories for LAPP and RoboCLIP currently do not include any license information.

Table 17: Prompt for generating diverse language instructions. The verb structures list and synonym nouns example are from Table 2 and Table 4 in LAMP (Adeniji et al., 2023), respectively.

`System Message:` Suppose you are an advanced visual assistant. Your task is to generate more instructions with the same meaning but different expressions based on the task instruction I provide, generating 40 new instructions for each task. The instructions you generate need to be as simple and clear as possible. Below is an example of an answer for picking up an object. The answer should be formatted as a Python list.
– Begin of instruction example –
`Task instruction`: "Pick up the [NOUN]"
`Answer:`
`Verb Structures List`
– End of instruction example –
Moreover, you need to be mindful to replace the nouns in the instructions with synonyms, such as replacing "bag" with the following words in the Python list:
– Begin of synonym example –
`Synonym Nouns`
– End of synonym example –
The tasks are from Meta-World benchmark and the image of the task is rendered in a 3D simulation environment. In the environment, there is a wooden table and a robotic arm. The robotic arm is placed above the table. The robotic arm needs to manipulate the object(s) on the table to complete tasks.
My instruction for `Task Name` task: `Task Instruction`
`Answer:`

Table 18: Preference label accuracy of VLP on ML45 test tasks.

Task	**VLP Acc.**
Bin Picking	95.0
Box Close	90.0
Door Lock	100.0
Door Unlock	100.0
Hand Insert	100.0
Average	97.0