TREND: Tri-teaching for Robust Preference-based Reinforcement Learning with Demonstrations

Shuaiyi Huang[1], Mara Levy[1], Anubhav Gupta[1], Daniel Ekpo[1], Ruijie Zheng[1], Abhinav Shrivastava[1]

***Abstract*— Preference feedback collected by human or VLM annotators is often noisy, presenting a significant challenge for preference-based reinforcement learning that relies on accurate preference labels. To address this challenge, we propose TREND, a novel framework that integrates few-shot expert demonstrations with a tri-teaching strategy for effective noise mitigation. Our method trains three reward models simultaneously, where each model views its small-loss preference pairs as useful knowledge and teaches such useful pairs to its peer network for updating the parameters. Remarkably, our approach requires as few as one to three expert demonstrations to achieve high performance. We evaluate TREND on various robotic manipulation tasks, achieving up to 90% success rates even with noise levels as high as 40%, highlighting its effective robustness in handling noisy preference feedback.**

I. Introduction

A key challenge when using reinforcement learning (RL) to train autonomous agents for various tasks is defining a suitable reward function. The reward function needs to give dense feedback to guide the agent's learning while preventing unintended behaviors (series of actions) that might earn high rewards but do not match the user's intent. However, creating an effective reward function is often complex and time-consuming. It involves multiple rounds of testing with different adjustments to ensure the agent learns the intended behavior without developing unplanned habits.

To address these difficulties, Preference-based Reinforcement Learning (PbRL) has emerged as a promising approach. PbRL sidesteps the challenge of explicitly designing a reward function by instead using human preferences as the reward signal [1–7]. PbRL employs a human-in-the-loop methodology, where humans provide preference feedback by comparing pairs of trajectory segments and labeling which segment is closer to the goal. This approach addresses the challenge of designing explicit reward functions and can result in behaviors that are better aligned with human intent [1]. Recent advances in this domain have integrated vision-language models (VLMs) [8], which can autonomously generate preferences, potentially reducing the need for direct human involvement.

However, in practice, both human and VLM-generated preference labels can be noisy and inconsistent. Human feedback is prone to bias and VLM labels often struggle with interpreting visual content, task-specific text, and temporal dynamics. For example, if two behaviors are qualitatively similar, it is likely that the preference labels provided by either human annotators or VLMs are not meaningful. As shown later in our experimental analysis (Sec. V-C), VLM preference labels can have a noise rate up to 40%, making them insufficient for reward learning without denoising. To make matters worse, a recent work [9] showed that even a 10% label corruption rate can significantly degrade performance. This underscores the need for robust preference-based RL methods that can effectively handle noisy labels.

In this work, we propose Tri-teaching for Robust Preference-based Reinforcement Learning with Demonstrations (TREND), a novel framework to address the challenge of noisy preference labels, as shown in Figure 1. Our key insight is the use of peer models to identify clean samples from a batch of noisy labels, instead of using a single model that can be biased in its estimation of noise. We therefore introduce a tri-teaching strategy where three reward models collaboratively select samples for training. Each model answers the question, "*Which preference pair is more likely to have clean labels?*", by computing losses over preference pairs and labeling those with small losses as likely clean. These selected pairs are then used to update the other models.

Our tri-teaching strategy is advantageous compared with existing works that simply average multiple reward model's predictions for sample selection [10] for the following reasons: (a) each peer model independently develops expertise in sample selection, enhancing robustness to noise, and (b) the definition of clean samples is dynamically learned from peer models rather than being fixed.

Additionally, to further combat extremely noisy preference labels, we integrate few-shot expert demonstrations into both pretraining and online PbRL adaptation. This provides a strong initialization for policy and ensures that at least some of the training data remains free of noise during online learning. Our experimental results demonstrate that under the extremely high noise rate of 40%, even a few demonstrations (1-3), we can achieve a success rate of $\sim$80%.

We evaluate our proposed method, TREND, on robotic manipulation tasks from Meta-world [11]. Our approach consistently outperforms baseline methods by a large margin, even under high noise (e.g., with noise rates up to 40%). We also conduct a detailed ablation study to demonstrate the benefits of our key components. The main contributions of this work are as follows:

- We present TREND, a novel framework that integrates expert demonstrations for robust learning under noisy preferences.
- We introduce a simple yet effective tri-teaching strategy for effective label denoising, leveraging peer models for cyclic clean feedback selection.

[1]University of Maryland, College Park. {huangshy, mlevy, anubhav, daniekpo, rzheng12, abhinav2}@umd.edu

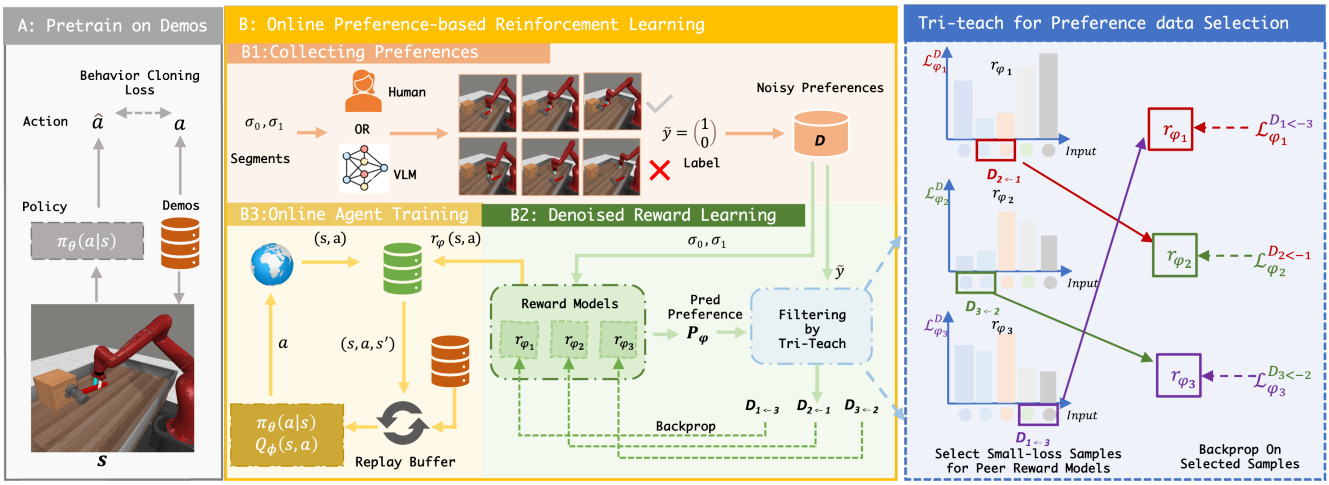


Fig. 1: **Overview of our method TREND.** First, we pretrain the policy network using behavior cloning (BC) with few-shot expert demonstrations for effective exploration (A). In the online training phase, noisy preferences are collected from human annotators or a vision-language model (B1). We then apply our tri-teaching strategy for denoised reward learning, where three collaborative reward models identify clean preference samples for each other (B2). Finally, the learned reward model is used to guide the agent's training (B3), ensuring robust performance despite noisy labels.

- TREND consistently outperforms existing PbRL baselines across a range of robotic manipulation tasks on Meta-World, even under varying levels of label noise. Notably, in scenarios with an extremely high noise rate of 40%, TREND achieves significant improvements in success rates: nearly 40% on the Button-Press task, 60% on the Drawer-Open task, and 70% on the Hammer task, compared to baselines using the same number of demonstrations. Even with noisy VLM-generated preference labels, TREND demonstrates an impressive success rate improvement of over 40% on the Drawer-Open task, showcasing its robustness in handling noisy data.

II. Related Work

Preference-based RL. While providing precise rewards for many tasks can be challenging for humans, expressing preferences for certain actions or states is often more intuitive and easier. Due to the simplicity and richness of relative preferences, preference-based reinforcement learning (PbRL) has gained significant attention in recent years [2, 3, 5–7, 9, 12]. Various approaches have been proposed to enhance the sample efficiency of PbRL. For example, PEBBLE [1] uses unsupervised policy pre-training to warm-start the policy and encourage exploration from the outset, while [13] uses imitation from expert demonstrations for policy initialization. Other works have focused on developing sampling strategies to select informative preference queries [3, 14, 15] and addressing exploration challenges by estimating reward model uncertainty [16].

Noisy label learning. Another critical issue with PbRL is its sensitivity to noise in preference labels. Lee et al. [9] showed that even a modest 10% rate of corrupted preference labels can substantially impair algorithmic performance. This concern becomes more pronounced in broader application contexts where preferences are collected from non-experts, increasing the likelihood of noise in labels. The issue of noisy labels is well-studied in the context of supervised learning [17] with proposed solutions such as sample selection methods [18, 19], architectural modifications for noise adaptation [20], and regularization [21]. These methods, however, are not easily transferable to PbRL due to limited sample sizes [10]. To alleviate this problem in PbRL, [22] introduced an encoder-decoder framework to capture diverse human preferences and improve robustness, though it required a substantially larger amount of preference data. RIME [10] introduced a dynamic sample selection method using a denoising discriminator to filter out the noise and incorporate a warm start strategy to mitigate errors from noisy labels. In contrast, we select reliable samples by a tri-teaching method and, optionally, explicitly incorporate a few expert demonstrations to combat extremely high noise.

Vision Language Models (VLMs) for RL. Recent advancements in integrating VLMs into RL have shown promising outcomes [8, 23–41], particularly in reward specification and preference-based methods. LLMs/VLMs have been employed to generate reward functions from text descriptions of goals, such as generating reward code directly [42] or serving as proxy reward functions aligned with given prompts [43]. Beyond reward generation, VLMs have been utilized for generating preference feedback, such as in RL-VLM-F [8], which uses VLMs to generate preference labels over pairs of the embodied agent's image observations. However, a key problem with VLM-generated preference labels is that they are often unreliable leading to high noise, so directly feeding all VLM feedbacks can lead to suboptimal (or worse) performance. In this work, we address this issue by introducing a tri-teaching method, supplemented with a few expert demonstrations, to denoise VLM's noisy preference labels.

III. Preliminaries

Preference-based RL. In classical RL [44], an agent learns by interacting with the environment for T discrete time steps.

At each step t the agent observes the environment state s_t and then uses a policy π to choose an action a_t. After executing the action, the environment returns a reward r and transitions to the next state s_{t+1} following the environment's dynamics. The objective of the learning agent is to learn a policy that maximizes the expected return, which is the discounted cumulative reward over time: $R_t = \sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k})$ where $\gamma \in [0, 1)$ is the discount factor, which balances the agent's preference for immediate versus long-term rewards.

Defining an explicit reward function can be difficult or infeasible in many real-world tasks. PbRL addresses this by relying on a teacher (often a human or an expert model) to provide feedback in the form of preferences between different behaviors (trajectory segments) of the agent [1, 10, 45, 46] (e.g., in Fig 1 B1). Instead of receiving scalar rewards from the environment, the agent learns from the teacher's preferences, using them to construct an internal reward model.

Formally, a trajectory σ is defined as a sequence of state-action pairs $\{(s_k, a_k), \ldots, (s_{k+H-1}, a_{k+H-1})\}$ over a fixed segment size H. Given two trajectory segments σ_0 and σ_1, the teacher provides feedback in the form of a preference label $\tilde{y} \in \{(1, 0), (0, 1), (0.5, 0.5)\}$. A "preference" is expressing which is preferred between pairs of clips of the agent's behavior, essentially distinguishing which segment has a higher reward. Here, $\tilde{y} = (1, 0)$ indicates $\sigma_0 \succ \sigma_1$. Correspondingly, $\tilde{y} = (0, 1)$ indicates $\sigma_1 \succ \sigma_0$, and $\tilde{y} = (0.5, 0.5)$ indicates that the teacher is indifferent between the two trajectories. The teacher's feedback is stored in a dataset D as a tuple $(\sigma_0, \sigma_1, \tilde{y})$. The agent uses this data to learn a reward function r_ψ.

The probability that the teacher prefers σ_1 over σ_0 can be modeled using the Bradley-Terry model [47], based on the estimated reward function r_ψ as

$$P_\psi[\sigma_i \succ \sigma_j] = \frac{\exp\left(\sum_t \hat{r}_\psi(s_t^i, a_t^i)\right)}{\sum_{i \in \{j,i\}} \exp\left(\sum_t \hat{r}_\psi(s_t^i, a_t^i)\right)}. \quad (1)$$

The reward function r_ψ is learned by minimizing the cross-entropy between the predicted preference and the label provided by the teacher. The loss function is defined as:

$$\mathcal{L}_{\text{CE}}(\psi) = \mathbb{E}\left[\ell_{\text{Reward}}\right] = -\mathbb{E}\Big[\tilde{y}(0) \ln P_\psi[\sigma_0 \succ \sigma_1] + \tilde{y}(1) \ln P_\psi[\sigma_1 \succ \sigma_0]\Big] \quad (2)$$

IV. TREND: Tri-teaching for Robust PbRL with Demonstrations

In this section, we will describe our approach, Tri-teaching for Robust Preference-based Reinforcement Learning with Demonstrations (TREND), in detail. TREND simultaneously trains three reward networks. Each reward model updates its parameters by using the preference pairs selected from the peer network via our proposed tri-teaching (Sec. IV-A). To further mitigate the impact of noise, we can leverage expert demonstrations in the learning process (Sec. IV-B).

A. Tri-teaching for Preference Data Selection

Following the small-loss principle [19], we hypothesize that the loss of a preference pair is positively correlated with how noisy the label is. Since a single model can be biased in its estimation of noise, potentially leading to error accumulation, we introduce a tri-teaching strategy where three peer reward networks collaboratively select samples for training. Each network answers the question, "*Which preference pair is more likely to have clean labels?*" and teaches the other networks by providing these samples for training. We use three teachers as existing work [1, 2, 10] used an ensemble of three reward models to improve the stability in reward learning. Using an odd number of experts [48, 49] is also a standard practice to avoid ties and provide a majority consensus signal.

Formally, we define a selection loss $\mathcal{L}_\psi^{\mathcal{D}}$ to quantify how noisy the label is as estimated by the reward model r_ψ. Given a batch of training preference data $\mathcal{D} = \{(\sigma_0^i, \sigma_1^i, \tilde{y}_i)\}_{i=1}^N$, where (σ_0^i, σ_1^i) is a pair of trajectory segments, and $\tilde{y}_i$ represents the corresponding preference label, we compute the selection loss for model ψ_k as

$$\mathcal{L}_{\psi_k}^{\mathcal{D}} = \frac{1}{N} \sum_{i=1}^{N} \ell_{\text{reward}}((\sigma_0^i, \sigma_1^i), \tilde{y}_i; \psi_k), \quad k \in \{1, 2, 3\}, \quad (3)$$

where k is the index of reward models.

During training, each reward model ψ_k identifies the subset of samples that have the smallest selection loss within a batch of preferences. The size of the subset is determined by a selection rate γ, which controls the proportion of samples retained. These samples are considered clean and are passed to a peer model for training. The selected data for peer training is defined as

$$\mathcal{D}_{k \leftarrow j} = \underset{\bar{\mathcal{D}}: |\bar{\mathcal{D}}| \geq \gamma |\mathcal{D}| \text{ and } \bar{\mathcal{D}} \subseteq \mathcal{D}}{\arg\min} \mathcal{L}_{\psi_j}^{\bar{\mathcal{D}}}. \quad (4)$$

Here $(k, j) \in \{(2, 1), (3, 2), (1, 3)\}$, and $\mathcal{D}_{k \leftarrow j}$ denotes the clean preferences identified by the reward model r_{ψ_j} which are then selected as training samples for r_{ψ_k}. In this way, the three models exchange opinions on which samples are clean, thereby enabling each to learn from its peers.

After the selection process, each reward model updates its parameters by using the preference pairs selected from the peer network as

$$\psi_k = \psi_k - \eta \nabla_{\psi_k} \mathcal{L}_{\psi_k}^{D_{k \leftarrow j}}. \quad (5)$$

This process is cyclic, where $\hat{r}_{\psi_1}$ selects for $\hat{r}_{\psi_2}$, $\hat{r}_{\psi_2}$ selects for $\hat{r}_{\psi_3}$, and $\hat{r}_{\psi_3}$ selects for $\hat{r}_{\psi_1}$. This tri-teaching strategy allows the models to collaboratively denoise the labels and improve learning stability despite noisy preferences.

We ensure diversity among the reward models by using different weight initialization and permuting the input sample orders. This prevents them from collapsing into similar models, maintaining diverse learning dynamics.

B. Incorporating Few-shot Expert Demonstrations

In the presence of noisy preference labels, purely learning from trial and error through online reinforcement learning

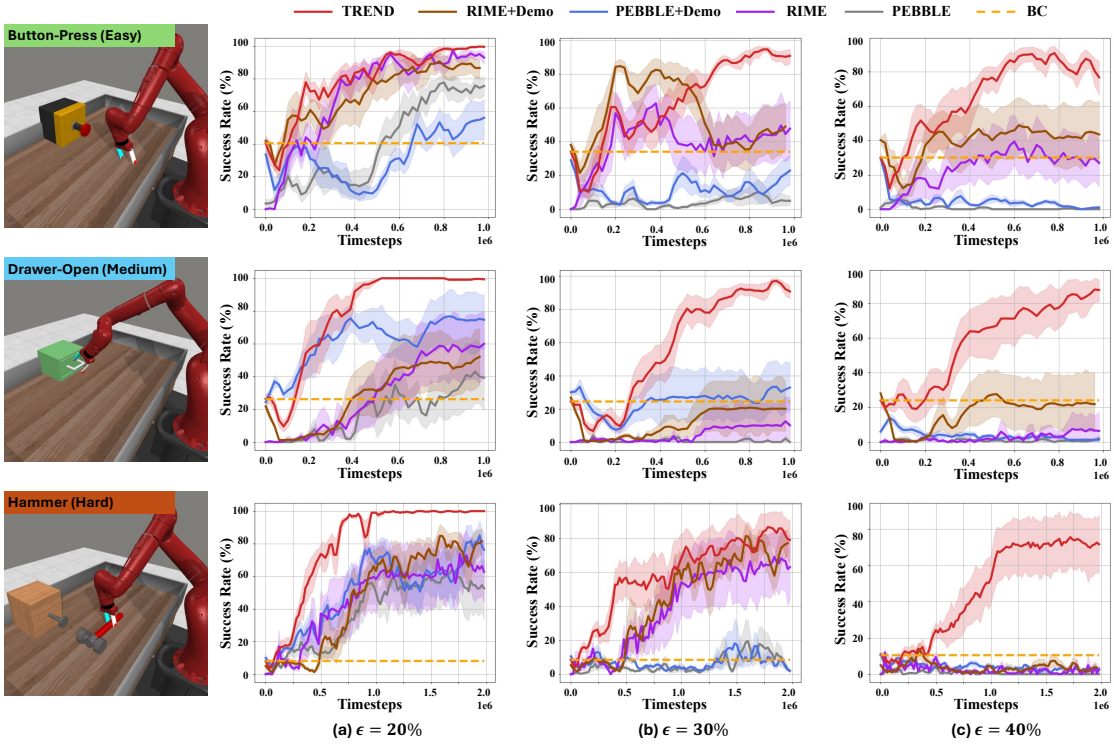


Fig. 2: **Learning curves for robot manipulation tasks on Meta-world.** Each row represents results for a specific task and each column corresponds to a different error rate ϵ. Results are averaged over five seeds. Shaded Areas represent standard deviation across seeds.

can lead to sub-optimal performance. To further mitigate the impact of noise and to provide a clean supervisory signal, we incorporate expert demonstrations into the pertaining and online training process. Expert trajectories offer high-quality information, which is particularly valuable when learning under high-noise conditions. These demonstrations serve to guide the agent toward better behavior by ensuring that a portion of the training data is noise-free.

Our method builds upon the widely used PbRL algorithm PEBBLE [1]. In the early stages of training, querying the teacher for preference labels often yields uninformative results, as the agent's policy tends to be random and the states uninteresting. To mitigate this issue, PEBBLE [1] introduced unsupervised exploration through pretraining the agent with an intrinsic reward based on state entropy [50], a strategy that has been widely adopted in subsequent research [2, 10].

To encourage more diversified exploration, we replace the unsupervised exploration phase of the PEBBLE algorithm with policy pretraining on demonstration data using behavior cloning (BC). Even though we are only using a few demonstrations, BC pretraining is significantly more effective than PEBBLE's intrinsic reward-based pretraining [1, 50]. BC establishes a stronger prior for the policy, enhances exploration, and improves the sampling of informative preference pairs. This is especially beneficial to hard tasks where extropy-based exploration struggles to find useful signal.

Next, after the policy initialization phase by BC, during the online training phase, for each batch of training data, we sample a proportion of $\alpha\%$ from the expert demonstration dataset $\mathcal{D}_{\text{expert}}$, where $\alpha\%$ controls the proportion of expert demonstrations. For the remaining data, we sample from the non-expert RL replay buffer $\mathcal{D}_{\text{replay}}$. Specifically, for data from the expert demonstrations, we update the policy network π_θ using the BC loss as

$$\mathcal{L}_{\pi_\theta}^{\text{BC}} = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}_{\text{expert}}}\left[\|\pi_\theta(s_t) - a_t\|^2\right]. \tag{6}$$

For non-expert data from the replay buffer, we apply the standard Soft Actor-Critic (SAC) [51] loss to update the policy network π_θ

$$\mathcal{L}_{\pi_\theta}^{\text{SAC}} = \mathbb{E}_{s_t\sim\mathcal{D}_{\text{replay}},a_t\sim\pi_\theta(\cdot|s_t)}\left[\alpha \log \pi_\theta(a_t \mid s_t) - Q_\phi(s_t,a_t)\right],$$

where Q_ϕ is the critic network. We then update the policy network π_θ using a weighted average of the two losses as

$$\mathcal{L}_{\pi_\theta}^{\text{TREND}} = \mathcal{L}_{\pi_\theta}^{\text{SAC}} + \lambda_{\text{BC}}\mathcal{L}_{\pi_\theta}^{\text{BC}}. \tag{7}$$

This way, the BC loss serves as a strong regularization term during online training, helping the policy network retain its BC initialization and avoid over-reliance on noisy preference data, which is especially beneficial in high-noise conditions.

V. Experiments

A. Experimental Setup

Tasks. We evaluate TREND on three robotic manipulation tasks from Meta-world [11]: Button-Press, Drawer-Open, and Hammer. These tasks are visually rich, with diverse textures and shading, requiring precise, fine-grained control to successfully manipulate objects. The tasks are shown in the first column of Figure 2.

To analyze our method's effectiveness, we consider two types of noisy annotators: one that generates preferences

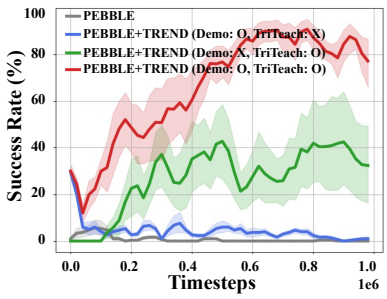
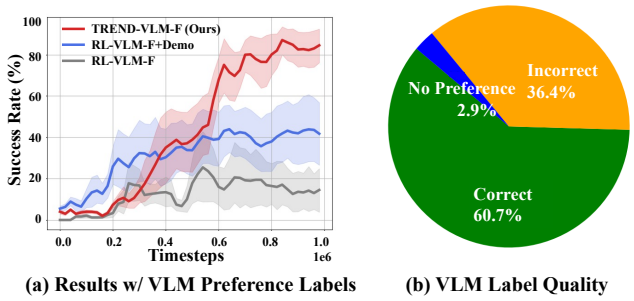


Fig. 3: **Contribution of each component on Button-Press ($\epsilon = 40\%$).** Our tri-teaching boosts performance significantly, and adding a single expert demonstration further enhances success, emphasizing the need for both components.



(a) Results w/ VLM Preference Labels (b) VLM Label Quality

Fig. 4: **Results on Drawer-Open using VLM (Gemini-1.5-flash) to generate preference feedback.** Our TREND-VLM-F achives the best result (left) under the high noise rate of VLM labels (right).

using a scripted policy and another that utilizes a VLM-based preference generator.

Scripted Noisy Preference Annotator. We generate synthetic preference feedback with an oracle reward following the technique used in previous works [1, 10]. Specifically, we assume a scripted teacher that determines the preference between two trajectory segments based on the sum of the ground-truth reward for each segment. To introduce noise, we flip each preference label with a probability of $\epsilon = 20\%, 30\%, 40\%$ as done in RIME [10]. This allows us to easily control the level of noise in preference labels.

VLM Preference Annotator. We also evaluate TREND using preference labels generated by a VLM. This approach follows the work done in RL-VLM-F [8]. For all VLM experiments, we use Gemini-1.5-Flash [52]. We prompt the VLM using rendered images of the two trajectory segments and the task description and obtain the generated preference label following [8]. If the VLM predicts that the provided segments are too similar, it will return a "no preference" answer and we skip this pair for reward training [8].

Implementation Details. We use PEBBLE [1] as our backbone algorithm. For the tri-teaching part of our method, we set $\gamma = 0.4$ (in Eq. 4). TREND uses 1 expert demonstration for Button Press, 2 for Drawer Open, and 3 for Hammer, with the BC loss coefficient $\lambda = 4.0$ (in Eq. 7). The proportion of sampling expert demos $\alpha\%$ is linearly decayed from 50% to 25% within the first 50k timesteps. We use the same hyperparameters for all experiments unless otherwise stated.

All other hyperparameters are consistent with PEBBLE. Both the baselines and our method use an ensemble of three reward models. The segment size H (i.e., length of σ) is 50. Preference queries are selected using disagreement sampling, following [10]. Disagreement sampling chooses pairs with high uncertainty based on the variance across an ensemble of predictors. All baseline methods are implemented using publicly available repositories. The initial state is randomly initialized for each episode, ensuring diverse configurations. We run each experiment with five random seeds and report the average performance with standard deviations.

B. Main Results

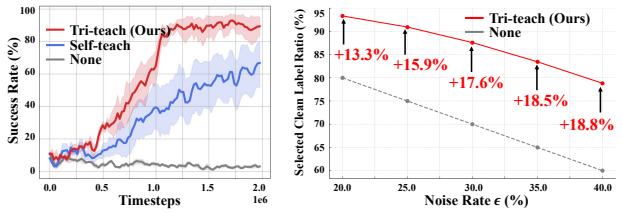
Figure 2 presents the learning curves of TREND alongside baseline methods on Meta-world, evaluated across three tasks: Button-Press (easy), Drawer-Open (moderate), and Hammer (hard). We compare our method against PEBBLE [1], which lacks denoising techniques, and RIME [10], the current state-of-the-art (SOTA) method for denoising in preference-based reinforcement learning (PbRL). For a fair comparison, we incorporated expert demonstrations into both PEBBLE and RIME in the same way and the same amount as ours, referring to these as PEBBLE+demo and RIME+demo. We also include a behavior cloning (BC) baseline learning purely from the demonstrations without reinforcement learning.

As shown in Figure 2, our approach consistently outperforms all baselines across all noise levels, from 20% to 40%. Notably, we achieve over 80% success at 40% noise with just 1 demonstration for Button-Press, 2 for Drawer-Open, and 3 for Hammer. This success is driven by our tri-teaching strategy for label selection and the integration of expert demonstrations. On average the baselines either perform at half of our success rate or fail completely. TREND is also superior to PEBBLE+demo and RIME+demo, showing the benefits of TREND come from both the expert demonstrations and the cross-model selection, not the expert demonstrations alone. These results underscore TREND's robustness to noisy preferences and its effectiveness in high-noise environments.

C. Ablation Study

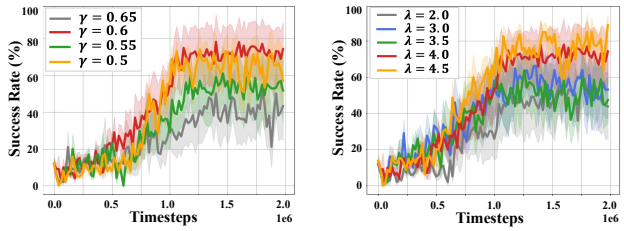
Effects of individual components. We study the effects of individual components as shown in Figure 3. When 40% of the labels are noisy, PEBBLE (grey) completely fails, as it lacks any denoising mechanism. However, when our tri-teaching strategy is applied, the success rate jumps to approximately 40% (green), highlighting its effectiveness in selecting clean labels from noisy data. Adding just one expert demonstration on top of the tri-teaching (orange) boosts the success rate to around 90%, demonstrating the critical role of even minimal clean supervision. Notably, adding an expert demo to PEBBLE alone (blue) does help avoid total failure, but is insufficient to achieve strong performance. These results underscore the necessity of both robust label selection through tri-teaching and additional clean supervision from expert demonstrations to effectively deal with high noise levels.

Performance with VLM Preference Annotator. Figure 4



(a) Comparison of Denoising Strategies (b) Analysis of Selected Clean Label Ratio

Fig. 5: **(Left):** Comparison between our method (w. Tri-teach) and a baseline denoising strategy (w. Self-teach) on Hammer ($\epsilon = 40\%$). **(Right):** Comparison of clean label ratio under different noise levels between our method (w. Tri-teach) and the baseline (w/o. Tri-teach) on Hammer.



(a) Selection rate γ **(b) BC loss weight λ**

Fig. 6: **Hyperparameter analysis on Hammer** (ϵ=40%).

compares our method with RL-VLM-F [8] on the Drawer-Open task, where a VLM is used for preference labeling instead of a scripted annotator. We observe that even with an advanced proprietary VLM such as Gemini-1.5-Flash, the generated preference labels still have a noise rate as high as 36.4%. Consequently, as shown in the figure, RL-VLM-F (grey) achieves only a ∼20% success rate due to the high noise level, although adding one expert demonstration improves its success rate to about 40% (blue).[1] In contrast, our method, TREND-VLM-F, achieves the best performance, reaching a ∼90% average success rate. This demonstrates TREND's effectiveness in handling high noise VLM-generated preference labels.

We also tested on Hammer and Button Press, but the limitations of the current VLM preference annotator, Gemini-1.5-Flash, led to high noise rates of up to 50%, causing both the baseline and our method to struggle. These results underscore the challenges that current VLMs face, such as difficulties in adapting to the visual appearance of simulations, understanding visual and temporal content, and selecting optimal viewpoints. While future advances in VLM development may help mitigate these issues, our efforts for denoising preference labels address a complementary aspect of improving VLM preference annotation quality.

Is cyclic preference data selection and training really necessary? In Figure 5 (left), we compare TREND with a baseline denoising strategy on the Hammer task with $\epsilon = 40\%$. Instead of cyclically selecting data with a small loss and passing it to another model for training, the baseline performs self-teaching, where each model uses its own small-loss samples for training. As shown in the figure, our method achieves the highest success rate, while the ablated self-teach approach performs worse than ours but still better than PEBBLE, which does not have any denoising mechanism. This demonstrates that selecting samples based on small loss can reduce noise to some extent, and that cyclic cross-model selection can further help denoising the preference labels.

Analysis of label quality with Tri-teach. In Figure 5 (right), we further analyze the clean preference label ratio achieved by our tri-teaching denoising method. Here, clean preference label ratio refers to the proportion of training data that has correct preference labels. On the Hammer task, using a scripted preference annotator, we documented the clean label ratio under varying noise levels (20%, 25%, 30%, 35%, and 40%) and compared it to the baseline PEBBLE without any denoising mechanisms. As shown in the figure, with our tri-teach mechanism, even at a noise ratio as high as 40%, the clean label ratio reaches ∼80%. This demonstrates that our method significantly enhances preference label quality, particularly in high noise conditions.

Impact of hyperparameters. We analyze how the hyperparameters of TREND influence the performance of PbRL under a high noise rate of 40%. In Figure 6, we present the learning curves for TREND with varying hyperparameters: (a) selection rate $\gamma \in \{50\%, 55\%, 60\%, 65\%\}$ and (b) BC loss weight $\lambda \in \{2.0, 3.0, 3.5, 4.0, 4.5\}$. We observe that TREND is generally robust to changes in γ, but performance can degrade when γ is too high or too low. This is likely due to selecting more noisy labels or insufficient data. Ideally, selecting a γ that aligns with the estimated clean label rate can optimize performance. Regarding the BC loss weight λ, TREND shows robustness across different values, with potential performance boosts when appropriately tuned.

VI. Conclusion

In this work, we study learning from noisy preference labels in PbRL formulation. We propose TREND, which leverages a tri-teaching strategy and incorporates few-shot expert demonstrations. Through cyclic peer training among three reward networks, TREND effectively filters out incorrect preference labels, improving label quality even under high noise levels. The integration of a few expert demonstrations further enhances its performance, providing a strong supervisory signal and regularization. Our experimental results on robotic manipulation tasks on Meta-world demonstrate that TREND consistently outperforms baseline methods, achieving high success rates even when the noise in the preference labels reaches up to 40%, highlighting the robustness of our methods under noisy preference labels.

Acknowledgement We acknowledge the constructive suggestions from reviewers and Eric Zhu. This work was partially supported by NSF CAREER Award (#2238769) to AS. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF or the U.S. Government.

[1]Note that the results reported in the original RL-VLM-F paper were obtained using a deterministic environment based on its released codebase, whereas we follow standard practice by randomizing the initial robot arm position and goal location.

REFERENCES

[1] K. Lee, L. Smith, and P. Abbeel, "Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. 1, 2, 3, 4, 5

[2] J. Park, Y. Seo, J. Shin, H. Lee, P. Abbeel, and K. Lee, "Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2022. 2, 3, 4

[3] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," in *Robotics: Science and Systems*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:12226563 2

[4] M. Levy, N. Saini, and A. Shrivastava, "Wayex: Waypoint exploration using a single demonstration," in *International Conference on Robotics and Automaction(ICRA)*, 2024.

[5] P. M. Pilarski, M. R. Dawson, T. Degris, F. Fahimi, J. P. Carey, and R. S. Sutton, "Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning," in *2011 IEEE international conference on rehabilitation robotics*. IEEE, 2011, pp. 1–7. 2

[6] A. Wilson, A. Fern, and P. Tadepalli, "A bayesian approach for policy learning from trajectory preference queries," *Advances in neural information processing systems*, vol. 25, 2012.

[7] C. Wirth, J. Fürnkranz, and G. Neumann, "Model-free preference-based reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016. 1, 2

[8] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Biyik, D. Held, and Z. Erickson, "Rl-vlm-f: Reinforcement learning from vision language foundation model feedback," *arXiv preprint arXiv:2402.03681*, 2024. 1, 2, 5, 6

[9] K. Lee, L. Smith, A. Dragan, and P. Abbeel, "B-pref: Benchmarking preference-based reinforcement learning," in *Proceedings of the Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021. 1, 2

[10] J. Cheng, G. Xiong, X. Dai, Q. Miao, Y. Lv, and F.-Y. Wang, "Rime: Robust preference-based reinforcement learning with noisy preferences," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. 1, 2, 3, 4, 5

[11] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019. [Online]. Available: https://arxiv.org/abs/1910.10897 1, 4

[12] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017. 2

[13] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, "Reward learning from human preferences and demonstrations in atari," *Advances in neural information processing systems*, vol. 31, 2018. 2

[14] E. Biyik and D. Sadigh, "Batch active preference-based learning of reward functions," in *Conference on robot learning*. PMLR, 2018, pp. 519–528. 2

[15] E. Bıyık, N. Huynh, M. J. Kochenderfer, and D. Sadigh, "Active preference-based gaussian process regression for reward learning," *arXiv preprint arXiv:2005.02575*, 2020. 2

[16] X. Liang, K. Shu, K. Lee, and P. Abbeel, "Reward uncertainty for exploration in preference-based reinforcement learning," 2022. [Online]. Available: https://arxiv.org/abs/2205.12401 2

[17] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 11, pp. 8135–8153, 2022. 2

[18] W. Wang, F. Feng, X. He, L. Nie, and T.-S. Chua, "Denoising implicit feedback for recommendation," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 373–381. 2

[19] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances in neural information processing systems*, vol. 31, 2018. 2, 3

[20] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in *International conference on learning representations*, 2022. 2

[21] M. Lukasik, S. Bhojanapalli, A. Menon, and S. Kumar, "Does label smoothing mitigate label noise?" in *International Conference on Machine Learning*. PMLR, 2020, pp. 6448–6458. 2

[22] W. Xue, B. An, S. Yan, and Z. Xu, "Reinforcement learning from diverse human preferences," *arXiv preprint arXiv:2301.11774*, 2023. 2

[23] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang, "Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies," *arXiv preprint arXiv:2412.10345*, 2024. 2

[24] S. Huang, D.-A. Huang, Z. Yu, S. Lan, S. Radhakrishnan, J. M. Alvarez, A. Shrivastava, and A. Anandkumar, "What is point supervision worth in video instance segmentation?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 2671–2681.

[25] S. Huang, S. Suri, K. Gupta, S. S. Rambhatla, S.-n. Lim, and A. Shrivastava, "Uvis: Unsupervised video instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 2682–2692.

[26] K. Choi, C. Cundy, S. Srivastava, and S. Ermon, "Lmpriors: Pre-trained language models as task-specific priors," *arXiv preprint arXiv:2210.12530*, 2022.

[27] Y. Du, O. Watkins, Z. Wang, C. Colas, T. Darrell, P. Abbeel, A. Gupta, and J. Andreas, "Guiding pretraining in reinforcement learning with large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 8657–8677.

[28] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, and D. Lindner, "Vision-language models are zero-shot reward models for reinforcement learning," *arXiv preprint arXiv:2310.12921*, 2023.

[29] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma, "Structured attentions for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1291–1300.

[30] S. Huang, L. Yang, B. He, S. Zhang, X. He, and A. Shrivastava, "Learning semantic correspondence with sparse annotations," in *European Conference on Computer Vision*. Springer, 2022, pp. 267–284.

[31] S. Huang, Q. Wang, and X. He, "Confidence-aware adversarial learning for self-supervised semantic matching," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2020, pp. 91–103.

[32] M. Klissarov, P. D'Oro, S. Sodhani, R. Raileanu, P.-L. Bacon, P. Vincent, A. Zhang, and M. Henaff, "Motif: Intrinsic motivation from artificial intelligence feedback," 2023. [Online]. Available: https://arxiv.org/abs/2310.00166

[33] B. He, X. Yang, H. Wang, Z. Wu, H. Chen, S. Huang, Y. Ren, S.-N. Lim, and A. Shrivastava, "Towards scalable neural representation for diverse videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6132–6142.

[34] S. Huang, Q. Wang, S. Zhang, S. Yan, and X. He, "Dynamic

context correspondence network for semantic alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2010–2019.

[35] S. Huang, M. Levy, Z. Jiang, A. Anandkumar, Y. Zhu, L. Fan, D.-A. Huang, and A. Shrivastava, "Ardup: Active region video diffusion for universal policies," *arXiv preprint arXiv:2406.13301*, 2024.

[36] M. Levy, S. Haldar, L. Pinto, and A. Shirivastava, "P3-po: Prescriptive point priors for visuo-spatial generalization of robot policies," in *International Conference on Robotics and Automaction(ICRA)*, 2024.

[37] X. Wu, T. Guan, D. Li, S. Huang, X. Liu, X. Wang, R. Xian, A. Shrivastava, F. Huang, J. L. Boyd-Graber, *et al.*, "Autohallusion: Automatic generation of hallucination benchmarks for vision-language models," *arXiv preprint arXiv:2406.10900*, 2024.

[38] Y. Wei, Y. Sun, R. Zheng, S. Vemprala, R. Bonatti, S. Chen, R. Madaan, Z. Ba, A. Kapoor, and S. Ma, "Is imitation all you need? generalized decision-making with dual-phase training," *arXiv preprint arXiv:2307.07909*, 2023.

[39] R. Zheng, C.-A. Cheng, H. D. III, F. Huang, and A. Kolobov, "PRISE: LLM-style sequence compression for learning temporal action abstractions in control," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=p225Od0aYt

[40] R. Zheng, Y. Liang, X. Wang, S. Ma, H. Daumé III, H. Xu, J. Langford, P. Palanisamy, K. S. Basu, and F. Huang, "Premier-taco is a few-shot policy learner: Pretraining multitask representation via temporal action-driven contrastive loss," in *Forty-first International Conference on Machine Learning*, 2024.

[41] R. Zheng, X. Wang, Y. Sun, S. Ma, J. Zhao, H. Xu, H. D. III, and F. Huang, "TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=ezCsMOy1w9 2

[42] T. Xie, S. Zhao, C. H. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu, "Text2reward: Reward shaping with language models for reinforcement learning," 2024. [Online]. Available: https://arxiv.org/abs/2309.11489 2

[43] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," *arXiv preprint arXiv:2303.00001*, 2023. 2

[44] R. S. Sutton, "Reinforcement learning: An introduction," *A Bradford Book*, 2018. 2

[45] R. Liu, F. Bai, Y. Du, and Y. Yang, "Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 270–22 284, 2022. 3

[46] C. Kim, J. Park, J. Shin, H. Lee, P. Abbeel, and K. Lee, "Preference transformer: Modeling human preferences using transformers for rl," *arXiv preprint arXiv:2303.00957*, 2023. 3

[47] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952. 3

[48] G. Chao, K. Zhang, X. Wang, and D. Chu, "Three-teaching: A three-way decision framework to handle noisy labels," *Applied Soft Computing*, vol. 154, p. 111400, 2024. 3

[49] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017. 3

[50] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk, "Nearest neighbor estimates of entropy," *American journal of mathematical and management sciences*, vol. 23, no. 3-4, pp. 301–321, 2003. 4

[51] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1861–1870. 4

[52] G. Team, "Gemini: A family of highly capable multimodal models," 2024. [Online]. Available: https://arxiv.org/abs/2312.11805 5