SURF: SEMI-SUPERVISED REWARD LEARNING WITH DATA AUGMENTATION FOR FEEDBACK-EFFICIENT PREFERENCE-BASED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Preference-based reinforcement learning (RL) has shown potential for teaching agents to perform the target tasks without a costly, pre-defined reward function by learning the reward with a supervisor's preference between the two agent behaviors. However, preference-based learning often requires a large amount of human feedback, making it difficult to apply this approach to various applications. This data-efficiency problem, on the other hand, has been typically addressed by using unlabeled samples or data augmentation techniques in the context of supervised learning. Motivated by the recent success of these approaches, we present SURF, a semi-supervised reward learning framework that utilizes a large amount of unlabeled samples with data augmentation. In order to leverage unlabeled samples for reward learning, we infer pseudo-labels of the unlabeled samples based on the confidence of the preference predictor. To further improve the label-efficiency of reward learning, we introduce a new data augmentation that temporally crops consecutive subsequences from the original behaviors. Our experiments demonstrate that our approach significantly improves the feedback-efficiency of the state-of-the-art preference-based method on a variety of locomotion and robotic manipulation tasks.

1 INTRODUCTION

Reward function plays a crucial role in reinforcement learning (RL) to convey complex objectives to agents. For various applications, where we can design an informative reward function, RL with deep neural networks has been used to solve a variety of sequential decision-making problems, including board games (Silver et al., 2017; 2018), video games (Mnih et al., 2015; Berner et al., 2019; Vinyals et al., 2019), autonomous control (Schulman et al., 2015; Bellemare et al., 2020), and robotic manipulation (Kober & Peters, 2011; Kober et al., 2013; Kalashnikov et al., 2018; Andrychowicz et al., 2020). However, there are several issues in reward engineering. First, designing a suitable reward function requires more human effort as the tasks become more complex. For example, defining a reward function for book summarization (Wu et al., 2021) is non-trivial because it is hard to quantify the quality of summarization in a scale value. Also, it has been observed that RL agents could achieve high returns by discovering undesirable shortcuts if the hand-engineered reward function does fully specify the desired task (Amodei et al., 2016; Hadfield-Menell et al., 2017; Lee et al., 2021a). Furthermore, there are various domains, where a single ground-truth function does not exist, and thus personalization is required by modeling different reward functions based on the user's preference.

Preference-based RL (Akrour et al., 2011; Christiano et al., 2017; Ibarz et al., 2018; Lee et al., 2021a) provides an attractive alternative to avoid reward engineering. Instead of assuming a hand-engineered reward function, a (human) teacher provides preferences between the two agent behaviors, and an agent learns how to show the desired behavior by learning a reward function, which is consistent with the teacher's preferences. Recent progress of preference-based RL has shown that the teacher can guide the agent to perform novel behaviors (Christiano et al., 2017; Stiennon et al., 2020; Wu et al., 2021), and mitigate the effects of reward exploitation (Lee et al., 2021a). However, existing preference-based approaches often suffer from expensive labeling costs, and this makes it hard to apply preference-based RL to various applications.

Meanwhile, recent state-of-the-art system in computer vision, the label-efficiency problem has been successfully addressed through semi-supervised learning (SSL) approaches (Berthelot et al., 2019; 2020; Sohn et al., 2020; Chen et al., 2020b). By leveraging unlabeled dataset, SSL methods have improved the performance with low cost. Data augmentation also plays a significant role in improving the performance of supervised learning methods (Cubuk et al., 2018; 2019). By using multiple augmented views of the same data as input, the performance has been improved by learning augmentation-invariant representations.

Inspired by the impact of semi-supervised learning and data augmentation, we present SURF: a **S**emi-s**U**pervised **R**eward learning with data augmentation for **F**eedback-efficient preference-based RL. To be specific, SURF consists of the following key ingredients:

(a) Pseudo-labeling (Lee, 2013; Sohn et al., 2020): We leverage unlabeled data by utilizing the artificial labels generated by learned preference predictor, which makes the reward function produce a confident prediction (see Figure 1a). We remark that such a SSL approach is particularly attractive in our setup as an unlimited number of unlabeled data can be obtained with no additional cost, i.e., from past experiences stored in the buffer.

(b) Temporal cropping augmentation: We generate slightly shifted or resized behaviors, which are expected to have the same preferences from a teacher, and utilize them for reward learning (see Figure 1b). Our data augmentation technique enhances the feedback-efficiency by enforcing consistencies (Xie et al., 2019; Berthelot et al., 2020; Sohn et al., 2020) to the reward function.

We remark that SURF is not a naïve application of these two techniques, but a novel combination of semi-supervised learning and the proposed data augmentation, which has not been considered or evaluated in the context of the preference-based RL.

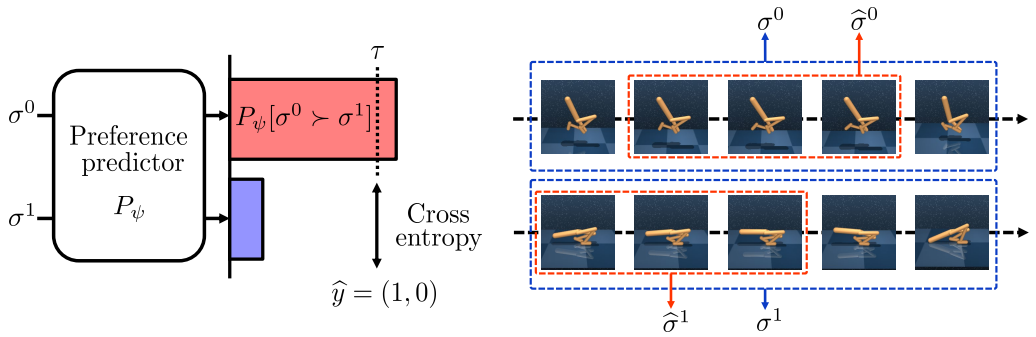
Our experiments demonstrate that SURF significantly improves the preference-based RL method (Lee et al., 2021a) on complex locomotion and robotic manipulation tasks from DeepMind Control Suite (Tassa et al., 2018; 2020) and Meta-world (Yu et al., 2020), in terms of feedback-efficiency. In particular, our framework could make RL agents achieve $\sim$100% of success rate on complex robotic manipulation task using only a few hundred preference queries, while its baseline method only achieves $\sim$50% of the success rate under the same condition (see Figure 3).

2 Related work

Preference-based RL. In the preference-based RL framework, a (human) supervisor provides preferences between the two agent behaviors and the agent uses this feedback to perform the task (Christiano et al., 2017; Ibarz et al., 2018; Leike et al., 2018; Stiennon et al., 2020; Wu et al., 2021; Lee et al., 2021a;b). Since this approach is only feasible if the feedback is practical for a human to provide, several strategies have been studied in the literature. Ibarz et al. (2018) initialized the agent's policy with imitation learning from the expert demonstrations, while Lee et al. (2021a) utilized unsupervised pre-training for policy initialization. Several sampling schemes (Sadigh et al., 2017; Biyik & Sadigh, 2018; Biyik et al., 2020) to select informative queries also have been adopted for improving the feedback-efficiency. Our approach differs in that we utilize unlabeled samples for reward learning, and also provide a novel data augmentation technique for the agent behaviors.

Data augmentation for RL. In the context of RL, data augmentation has been widely investigated for improving data-efficiency (Srinivas et al., 2020; Yarats et al., 2021), or RL generalization (Cobbe et al., 2019; Lee et al., 2019). For example, RAD (Laskin et al., 2020) demonstrated that data augmentation, such as random crop, can improve both data-efficiency and generalization of RL algorithms. While these methods are known to be beneficial to learn policy in the standard RL setup, they have not been tested for learning *rewards*. To the best of our knowledge, we present the first data augmentation method specially designed for learning reward function.

Semi-supervised learning. The goal of semi-supervised learning (SSL) is to leveraging unlabeled samples to improve a model's performance when the amount of labeled samples are limited. In an attempt to leverage the information in the unlabeled dataset, a number of techniques have been proposed, e.g., entropy minimization (Grandvalet & Bengio, 2004; Lee, 2013) and consistency regularization (Sajjadi et al., 2016; Miyato et al., 2018; Xie et al., 2019; Sohn et al., 2020). Recently, the combination of these two approaches have shown state-of-the-art performance in benchmarks, e.g., MixMatch (Berthelot et al., 2019), and ReMixMatch (Berthelot et al., 2020), when used with



(a) Pseudo-labeling (b) Temporal cropping

Figure 1: Overview of SURF. (a) We leverage unlabeled experiences by generating pseudo-labels $\widehat{y}$ from the preference predictor P_ψ in (1). To mitigate the negative effects from this semi-supervised learning, we only utilize pseudo-labels when the confidence of the predictor is higher than threshold τ. (b) Given two segments (σ^0, σ^1), we generate augmented segments $(\widehat{\sigma}^0, \widehat{\sigma}^1)$ by cropping the subsequence from each segment.

advanced data augmentation techniques (Zhang et al., 2018; Cubuk et al., 2019). Specifically, FixMatch (Sohn et al., 2020) revisits pseudo-labeling technique and demonstrates that joint usage of pseudo-labels and consistency regularization achieves remarkable performance due to its simplicity.

3 Preliminaries

Reinforcement learning (RL) is a framework where an agent interacts with an environment in discrete time (Sutton & Barto, 2018). At each timestep t, the agent receives a state $\mathbf{s}_t$ from the environment and chooses an action $\mathbf{a}_t$ based on its policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$. In conventional RL framework, the environment gives a reward $r(\mathbf{s}_t, \mathbf{a}_t)$ and the agent transitions to the next state $\mathbf{s}_{t+1}$. The return $\mathcal{R}_t = \sum_{k=0}^{\infty} \gamma^k r(\mathbf{s}_{t+k}, \mathbf{a}_{t+k})$ is defined as discounted cumulative sum of the reward with discount factor $\gamma \in [0, 1)$. The goal of the agent is to learn a policy that maximizes the expected return.

Preference-based reinforcement learning. In this paper, we consider a preference-based RL framework, which does not assume the existence of hand-engineered reward. Instead, a (human) teacher provides preferences between the agent's behaviors and the agent uses this feedback to perform the task (Christiano et al., 2017; Ibarz et al., 2018; Leike et al., 2018; Stiennon et al., 2020; Lee et al., 2021a;b; Wu et al., 2021) by learning a reward function, which is consistent with the observed preferences.

We formulate a reward learning problem as a supervised learning problem (Wilson et al., 2012; Christiano et al., 2017). Formally, a segment σ is a sequence of observations and actions $\{(\mathbf{s}_k, \mathbf{a}_k), ..., (\mathbf{s}_{k+H-1}, \mathbf{a}_{k+H-1})\}$. Given a pair of segments (σ^0, σ^1), a teacher gives a feedback indicating which segment is preferred, i.e., $y \in \{0, 1, 0.5\}$, where 1 indicates $\sigma^1 \succ \sigma^0$, 0 indicates $\sigma^0 \succ \sigma^1$, and 0.5 implies an equally preferable case. Each feedback is stored in a dataset $\mathcal{D}$ as a triple (σ^0, σ^1, y). Then, we model a preference predictor using the reward function $\widehat{r}_\psi$ following the Bradley-Terry model (Bradley & Terry, 1952):

$$P_\psi[\sigma^1 \succ \sigma^0] = \frac{\exp(\sum_t \widehat{r}_\psi(\mathbf{s}_t^1, \mathbf{a}_t^1))}{\sum_{i \in \{0,1\}} \exp(\sum_t \widehat{r}_\psi(\mathbf{s}_t^i, \mathbf{a}_t^i))}, \tag{1}$$

where $\sigma^i \succ \sigma^j$ denotes the event that segment i is preferable to segment j. The underlying assumption of this model is that the teacher's probability of preferring a segment depends exponentially on the accumulated sum of the reward over the segment. The reward model is trained through supervised learning with teacher's preferences. Specifically, given a dataset of preferences $\mathcal{D}$, the reward function is updated by minimizing the binary cross-entropy loss:

$$\mathcal{L}^{\texttt{CE}} = \mathop{\mathbb{E}}_{(\sigma^0,\sigma^1,y)\sim\mathcal{D}} \Big[\mathcal{L}^{\texttt{Reward}}\Big] = -\mathop{\mathbb{E}}_{(\sigma^0,\sigma^1,y)\sim\mathcal{D}} \Big[(1-y)\log P_\psi[\sigma^0 \succ \sigma^1] + y \log P_\psi[\sigma^1 \succ \sigma^0]\Big].$$

Algorithm 1 SURF

Require: Hyperparameters: unlabeled batch ratio μ, threshold parameter τ, and loss weight λ
Require: Set of collected labeled data $\mathcal{D}_l$, and unlabeled data $\mathcal{D}_u$

```
for each gradient step do
    Sample labeled batch {(σ_l^0, σ_l^1, y)^(i)}_{i=1}^B ~ D_l
    Sample unlabeled batch {(σ_u^0, σ_u^1)^(j)}_{j=1}^{μB} ~ D_u
    // DATA AUGMENTATION FOR LABELED DATA
    for i in 1 ... B do
        (σ̂_l^0, σ̂_l^1)^(i) ← TDA((σ_l^0, σ_l^1)^(i)) in Algorithm 2
    end for
    // PSEUDO-LABELING AND DATA AUGMENTATION FOR UNLABELED DATA
    for j in 1 ... μB do
        Predict pseudo-labels ŷ((σ_u^0, σ_u^1)^(j))
        (σ̂_u^0, σ̂_u^1)^(j) ← TDA((σ_u^0, σ_u^1)^(j)) in Algorithm 2
    end for
    Optimize L^SSL (3) with respect to ψ
end for
```

The reward function $\widehat{r}_\psi$ is usually optimized only using labels from real human, which are expensive to obtain in practice. Instead, we propose a simple yet effective method based on semi-supervised learning and data augmentation to improve the feedback-efficiency of preference-based learning.

4 SURF

In this section, we present SURF: a **S**emi-s**U**pervised **R**eward learning with data augmentation for **F**eedback-efficient preference-based RL, that can be used in conjunction with any existing preference-based RL methods. Our main idea is to leverage a large number of unlabeled samples collected from environments for reward learning, by inferring pseudo-labels. To further increase the effective number of training samples, we propose a new data augmentation that temporally crops the subsequence of the agent behaviors. The full procedure of our unified framework in Algorithm 1 (See Figure 1 for the overview of our method).

4.1 SEMI-SUPERVISED REWARD LEARNING

To improve the feedback efficiency, we propose a semi-supervised learning (SSL) method for leveraging unlabeled experiences in the buffer for reward learning. In addition to a *labeled dataset* $\mathcal{D}_l = \{(\sigma_l^0, \sigma_l^1, y)^{(i)}\}_{i=1}^{N_l}$, we utilize an *unlabeled dataset* $\mathcal{D}_u = \{(\sigma_u^0, \sigma_u^1)^{(i)}\}_{i=1}^{N_u}$ to optimize the reward model r_ψ.[1] Specifically, we generate the artificial labels $\widehat{y}$ by *pseudo-labeling* (Lee, 2013; Sohn et al., 2020) for the unlabeled dataset $\mathcal{D}_u$. We infer a preference $\widehat{y}$ for an unlabeled segment pair (σ_u^0, σ_u^1) as a class with higher probability as follows:

$$\widehat{y}(\sigma_u^0, \sigma_u^1) = \begin{cases} 0, & \text{if } P_\psi[\sigma_u^0 \succ \sigma_u^1] > 0.5 \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

By generating labels from the prediction model, we can obtain free supervision for optimizing our reward model.

However, pseudo-labels from low-confidence predictions can be inaccurate, and such noisy feedback can significantly degrade the peformance of preference-based learning (Lee et al., 2021b). To filter out inaccurate pseudo-labels, we only use unlabeled samples for training when the confidence of the predictor is higher than a pre-defined threshold (Rosenberg et al., 2005). Then the reward model r_ψ is optimized by minimizing the following objective:

$$\mathcal{L}^{\texttt{SSL}} = \mathop{\mathbb{E}}_{\substack{(\sigma_l^0, \sigma_l^1, y) \sim \mathcal{D}_l, \\ (\sigma_u^0, \sigma_u^1) \sim \mathcal{D}_u}} \Big[\mathcal{L}^{\texttt{Reward}}(\sigma_l^0, \sigma_l^1, y) + \lambda \cdot \mathcal{L}^{\texttt{Reward}}(\sigma_u^0, \sigma_u^1, \widehat{y}) \cdot \mathbb{1}(P_\psi[\sigma_u^{k^*} \succ \sigma_u^{1-k^*}] > \tau) \Big], \tag{3}$$

[1] The unlabeled dataset $\mathcal{D}_u$ is not constrained to a fixed size since one can collect those unlabeled samples flexibly by sampling arbitrary pairs of experiences from the buffer.

Algorithm 2 TDA: Temporal data augmentation for reward learning

Require: Minimum and maximum length $H_{\texttt{min}}$ and $H_{\texttt{max}}$, respectively, for cropping
Require: Pair of segments (σ^0, σ^1) with length H
1: $\sigma^0 = \{(\mathbf{s}_0^0, \mathbf{a}_0^0), ..., (\mathbf{s}_{H-1}^0, \mathbf{a}_{H-1}^0)\}$
2: $\sigma^1 = \{(\mathbf{s}_0^1, \mathbf{a}_0^1), ..., (\mathbf{s}_{H-1}^1, \mathbf{a}_{H-1}^1)\}$
3: Sample H' from a range of $[H_{\texttt{min}}, H_{\texttt{max}}]$
4: Sample k_0, k_1 from a range of $[0, H - H']$
5: // RANDOMLY CROP A SEQUENCE WITH LENGTH H'
6: $\widehat{\sigma}^0 \leftarrow \{(\mathbf{s}_{k_0}^0, \mathbf{a}_{k_0}^0), ..., (\mathbf{s}_{k_0+H'-1}^0, \mathbf{a}_{k_0+H'-1}^0)\}$
7: $\widehat{\sigma}^1 \leftarrow \{(\mathbf{s}_{k_1}^1, \mathbf{a}_{k_1}^1), ..., (\mathbf{s}_{k_1+H'-1}^1, \mathbf{a}_{k_1+H'-1}^1)\}$
8: Return $(\widehat{\sigma}^0, \widehat{\sigma}^1)$

where $k^* = \arg\max_{j \in \{0,1\}} \widehat{y}(j)$ is an index of the preferred segment from the pseudo-label, λ is a hyperparameter that balances the losses, and τ is a confidence threshold. Training with the pseudo-labels encourages the model to output more confident predictions on unlabeled samples. This can be seen as a form of entropy minimization (Grandvalet & Bengio, 2004), which is essential to the success of recent SSL methods (Berthelot et al., 2019; 2020). The entropy minimization can improve the reward learning by forcing the preference predictor to be low-entropy (i.e., high-confidence) on unlabeled samples. During training, we sample a larger minibatch of unlabeled samples than labeled ones by a factor of μ following (Sohn et al., 2020), since unlabeled samples with low confidence are dropped within minibatch.

4.2 TEMPORAL DATA AUGMENTATION FOR REWARD LEARNING

To further improve the feedback-efficiency in preference-based RL, we propose a new data augmentation technique specially designed for reward learning. Specifically, for a given two segments and preference (σ^0, σ^1, y), we generate augmented segments $(\widehat{\sigma}^0, \widehat{\sigma}^1, y)$ by cropping the subsequence from each segment (see Algorithm 2 for more details).[2] Then, we utilize augmented samples $(\widehat{\sigma}^0, \widehat{\sigma}^1)$ to optimize the cross-entropy loss in (3). The intuition behind the augmentation is that for a given pair of behavior clips, the human teacher may keep their relative preferences for slightly shifted or resized versions of them. In the context of SSL, data augmentation is also related to consistency regularization (Xie et al., 2019; Berthelot et al., 2020; Sohn et al., 2020) approaches that train the model to output similar predictions on augmented versions of the same sample. Namely, this *temporal cropping* method enables our framework can also enjoy the benefits of consistency regularization.

5 EXPERIMENTS

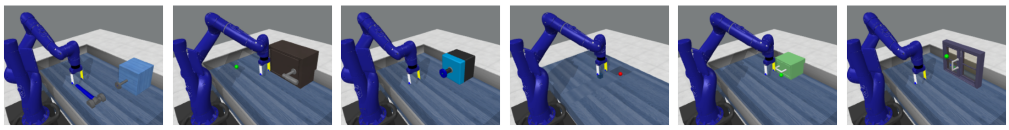
We design our experiments to investigate the following:

- How does SURF improve the existing preference-based RL method in terms of feedback efficiency?
- What is the contribution of each of the proposed components in SURF?
- How does the number of queries affect the performance of SURF?
- Is temporal cropping better than existing state-based data augmentation methods in terms of feedback efficiency?

5.1 SETUPS

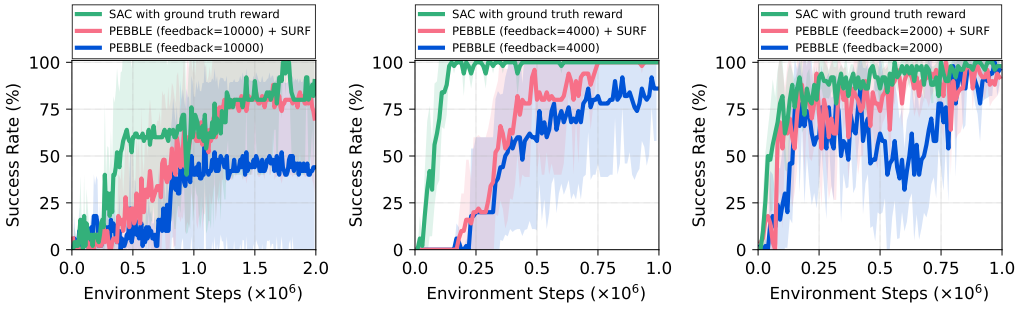
We evaluate SURF on several complex robotic manipulation and locomotion tasks from Meta-world (Yu et al., 2020) and DeepMind Control Suite (DMControl; Tassa et al. 2018; 2020), respectively. Similar to prior works (Christiano et al., 2017; Lee et al., 2021a;b), in order to systemically

[2]The length of the cropped segment is generated randomly across the batch but the same for segment pairs, because the preference predictor uses the accumulated sum of the reward over time.



(a) Hammer (b) Door Open (c) Button Press (d) Sweep Into (e) Drawer Open (f) Window Open

Figure 2: Rendered images of robotic manipulation tasks from Meta-world. Our goal is learning various locomotion and manipulation skills using preferences from a teacher.



(a) Hammer (b) Door Open (c) Button Press

(d) Sweep Into (e) Drawer Open (f) Window Open

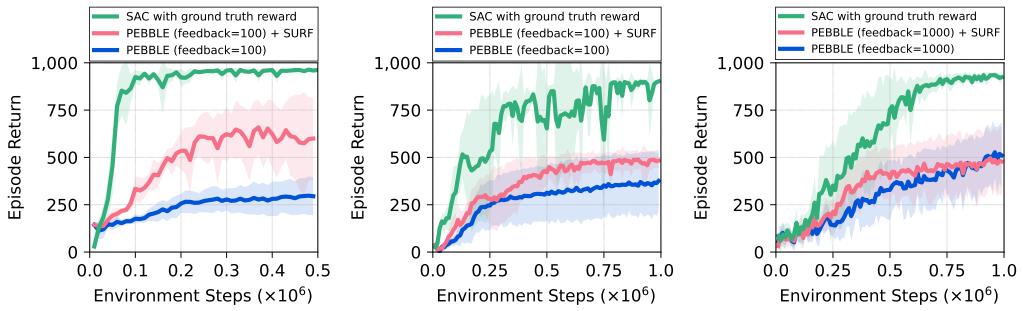
Figure 3: Learning curves on robotic manipulation tasks as measured on the success rate. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.

evaluate the performance, we consider a scripted teacher that provides preferences between two trajectory segments to the agent according to the underlying reward function.[3] Since preferences of the scripted teacher exactly reflects ground truth reward of the environment, one can evaluate the algorithms quantitatively by measuring the true return.

We remark that SURF can be combined with any preference-based RL algorithms by replacing the reward learning procedure of its backbone method. In our experiments, we choose state-of-the-art approach, PEBBLE (Lee et al., 2021a), as our backbone algorithm. Since PEBBLE utilizes SAC (Haarnoja et al., 2018) algorithm to learn the policy, we also compare to SAC using the ground truth reward directly, as an upper bound of PEBBLE and our method. We note that our goal is not to outperform SAC, but rather to perform closely using as few preference queries as possible.

Implementation details of SURF. For all experiments, we use the same hyperparameters used by the original SAC and PEBBLE algorithms, such as learning rate of neural networks and frequency of the feedback session. For query selection strategy, we use the disagreement-based sampling scheme,

[3]While utilizing preferences from the human teacher is ideal, this makes hard to evaluate algorithms quantitatively and quickly.



(a) Walker (b) Cheetah (c) Quadruped

Figure 4: Learning curves on locomotion tasks as measured on the ground truth reward. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.

which selects queries with high uncertainty, i.e., ensemble disagreement (see Appendix B for more details). At each feedback session, we sample unlabeled samples as 10 times of labeled ones by uniform sampling scheme, unless otherwise noted. Although we only use such amount of unlabeled samples for time-efficient training, we note that one can utilize much more unlabeled samples as needed. For the hyperparameters of SURF, we fix the loss weight $\lambda = 1$, and unlabeled batch ratio $\mu = 4$ for all experiments, and use threshold parameter $\tau = 0.999$ for Window Open, Sweep Into, Cheetah tasks, and $\tau = 0.99$ for the others. We provide more experimental details in Appendix B.

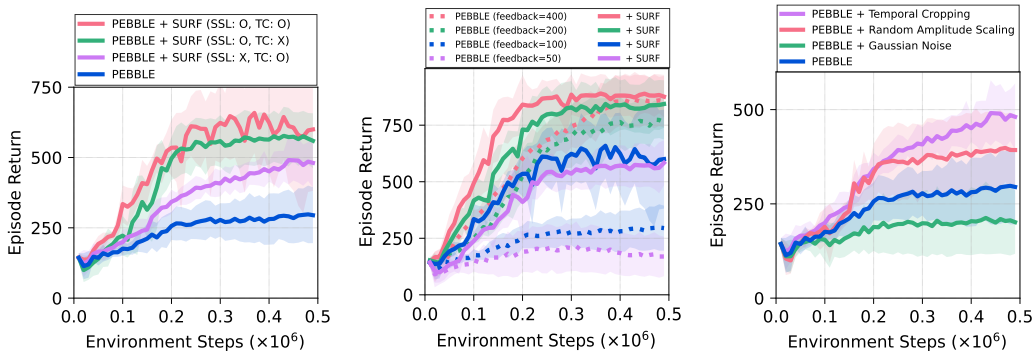
5.2 Benchmark tasks with scripted teachers

Meta-world experiments. Meta-world consists of 50 robotic manipulation tasks, which are designed for learning diverse manipulation skills. We consider six tasks from Meta-world, to investigate how SURF improves a preference-based learning method on a range of complex robotic manipulation tasks (see Figure 2). Figure 3 shows the learning curves of SAC, PEBBLE and SURF (which combined with PEBBLE) on the manipulation tasks. In each task, PEBBLE and SURF utilize the same number of preference queries for fair comparison. As shown in the figure, SURF significantly improves the performance of PEBBLE given the same number of feedback on all tasks we considered, and matches the performance of SAC using the ground truth reward on four tasks. For example, we find that when using 400 preference queries, SURF (red) reaches the same performance as SAC (green) while PEBBLE (blue) is far behind to SAC on Window Open task. We also observe that SURF achieves similar performance to PEBBLE with much less labels. For example, to achieve comparable performance to SAC on Window Open task, PEBBLE needs 2,500 queries (reported in (Lee et al., 2021a)), requiring about 6 times more queries than SURF. These results demonstrate that SURF significantly reduces the feedback requirement to solve the complex tasks.

DMControl experiments. For locomotion tasks, we choose three complex environments from DMControl: Walker-walk, Cheetah-run, and Quadruped-walk. Figure 4 shows the learning curves of the algorithms with same number of queries. We find that using a budget of 100 or 1,000 queries (which takes only few human minutes), SURF (red) could significantly improve the performance of PEBBLE (blue). These results again demonstrate that that SURF improves the feedback-efficiency of preference-based RL methods on a variety of complex tasks.

5.3 Ablation study

Component analysis. To evaluate the effect of each technique in SURF individually, we incrementally apply semi-supervised learning (SSL) and temporal cropping (TC) to our backbone algorithm, PEBBLE. Figure 5a shows the learning curves of SURF on Walker-walk task with 100 queries. We observe that leveraging unlabeled samples via pseudo-labeling (green) significantly improves PEBBLE, in terms of both sample-efficiency and asymptotic performance, while standard PEBBLE (blue) suffers from lack of supervision. In addition, both supervised (blue) and semi-supervised



(a) Contributions of each component (b) Query size (c) Effects of data augmentation

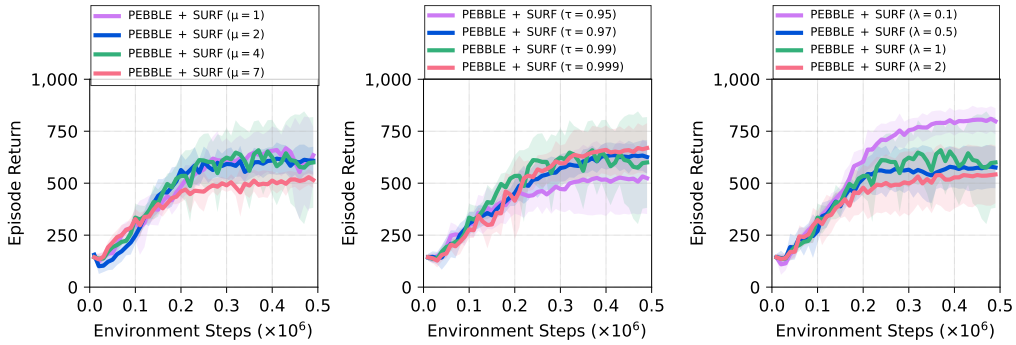
Figure 5: Ablation study on Walker-walk. (a) Contribution of each technique in SURF, i.e., semi-supervised learning (SSL) and temporal cropping (TC). (b) Effects of query size. (c) Comparison of augmentation methods. The results show the mean and standard deviation averaged over five runs.

(green) reward learning are further improved by additionally utilizing temporal cropping (purple and red, respectively). This implies that our augmentation method improves label-efficiency by generating diverse behaviors share the same labels. Also, the results show that the key components of SURF are both effective, and their combination is essential to our method's success.

Effects of query size. To investigate how the number of queries affects the performance of SURF, we evaluate the performance of SURF with a varying number of queries $N \in \{50, 100, 200, 400\}$. As shown in Figure 5b, SURF (solid lines) consistently improves the performance of PEBBLE (dotted lines) across a wide range of query sizes. The gain from SURF becomes even more significant in the extreme label-scarce scenarios, i.e., $N \in \{50, 100\}$.

Comparison to other augmentation for state-based inputs. To demonstrate that temporal cropping can enduce significant improvements for reward learning, we compare our method to other augmentation methods for state-based inputs. We consider random amplitude scaling (RAS) and adding Gaussian noise (GN) proposed in Laskin et al. (2020) as our baselines. RAS multiplies an uniform random variable z to the state, i.e., $\widehat{\mathbf{s}} = \mathbf{s} \cdot z$, where $z \sim \text{Unif}[\alpha, \beta]$, and GN adds a multivariate Gaussian random variable $\mathbf{z}$ to the state, i.e., $\widehat{\mathbf{s}} = \mathbf{s} + \mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(0, I)$. As proposed in Laskin et al. (2020), we apply these methods consistently along the time dimension, and choose the parameters for RAS as $\alpha = 0.8$, $\beta = 1.2$. Specifically, for a given segment $\sigma = (\{(\mathbf{s}_k, \mathbf{a}_k), ..., (\mathbf{s}_{k+H-1}, \mathbf{a}_{k+H-1})\})$, we obtain the augmented sample $\widehat{\sigma}$ by perturbing each state along the segment, i.e., $\widehat{\sigma} = (\{(\widehat{\mathbf{s}}_k, \mathbf{a}_k), ..., (\widehat{\mathbf{s}}_{k+H-1}, \mathbf{a}_{k+H-1})\})$. In Figure 5c, we plot the learning curves of PEBBLE with various data augmentations on Walker-walk task with 100 queries. We observe that RAS improves the performance of PEBBLE, but temporal cropping still outperforms these two methods. GN degrades the performance, possibly due to the noisy inputs. Since RAS is an orthogonal approach to augment state-based inputs, one can integrate them with our method to further improve the performance. This may be an interesting future direction for addressing feedback-efficiency in preference-based RL.

Effects of hyperparameters of SURF. We investigate how the hyperparameters of SURF affect the performance of preference-based RL. In Figure 6, we plot the learning curve of SURF with different set of hyperparameters: (a) unlabeled batch ratio $\mu \in \{1, 2, 4, 7\}$, (b) threshold parameter $\tau \in \{0.95, 0.97, 0.99, 0.999\}$, and (c) loss weight $\lambda \in \{0.1, 0.5, 1, 2\}$, respectively. First, we observe that SURF is quite robust on μ, but the performance slightly drops with a large batch size $\mu = 7$. We expect that this is because a large batch size makes the reward model overfit to unlabeled data. We also observe that SURF is also robust on the threshold τ, except for the smallest value of 0.95. Because there are only two classes in tasks, the optimal threshold could larger than the value typically used in previous SSL methods (Sohn et al., 2020), i.e., 0.95. In the case of the loss weight λ, tuning this parameter brings more improvements than other hyperparameters. Although



(a) Unlabeled batch ratio μ (b) Threshold parameter τ (c) Loss weight λ

Figure 6: Hyperparameter analysis on Walker-walk using 100 preference queries. The results show the mean and standard deviation averaged over five runs.

we use a simple choice, i.e., $\lambda = 1$, in our experiments, more tuning λ would further improve the performance of our method.

6 Discussion

In this work, we present SURF, a semi-supervised reward learning algorithm with data augmentation for preference-based RL. First, in order to utilize an unlimited number of unlabeled data, we utilize pseudo-labeling on confident samples. Also, to enforce consistencies to the reward function, we propose a new data augmentation method called temporal cropping augmentation. Our experiments demonstrate that SURF significantly improves feedback-efficiency of current state-of-the-art method on a variety of complex robotic manipulation and locomotion tasks. We believe that SURF can scale up deep RL to more diverse and challenging domains by making preference-based learning more tractable.

An interesting future direction is to extend state-based inputs to partially-observable or high-dimensional inputs, e.g., pixels. One can expect that representation learning based on unlabeled samples and data augmentation (Chen et al., 2020a; Grill et al., 2020) is crucial to handle such inputs. We think that our investigations on leveraging unlabeled samples and data augmentation would be useful in representation learning for preference-based RL.

Ethics statement. Preference-based RL can align RL agents with the teacher's preferences, which enables us to apply RL to diverse problems and obtain strong AI. However, there could be possible negative impacts if a malicious user corrupts the preferences to teach the agent harmful behaviors. Since we have proposed a method that makes preference-based RL algorithms more feedback-efficiently, our method may reduce the efforts for teaching not only the desirable behaviors, but also such bad behaviors. For this reason, in addition to developing algorithms for better performance and efficiency, it is also important to consider safe adaptation in the real world.

Reproducibility statement. We describe the implementation details of SURF in Appendix B, and also provide our source code in the supplementary material.

References

Riad Akrour, Marc Schoenauer, and Michele Sebag. Preference-based policy learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

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

Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019.

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020.

Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *Conference on Robot Learning*, 2018.

Erdem Biyik, Nicolas Huynh, Mykel J Kochenderfer, and Dorsa Sadigh. Active preference-based gaussian process regression for reward learning. In *Robotics: Science and Systems*, 2020.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.

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

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, 2019.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, 2004.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. Inverse reward design. In *Advances in Neural Information Processing Systems*, 2017.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In *Advances in Neural Information Processing Systems*, 2018.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Jens Kober and Jan Peters. Policy search for motor primitives in robotics. *Machine learning*, 84 (1-2):171–203, 2011.

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

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems*, 2020.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, 2013.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. *arXiv preprint arXiv:1910.05396*, 2019.

Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, 2021a.

Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-pref: Benchmarking preference-based reinforcement learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021b. URL https://openreview.net/forum?id=ps95-mkHF_.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. 2005.

Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2016.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

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

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

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, 2020.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*, 2020.

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

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm_control: Software and tasks for continuous control. *arXiv preprint arXiv:2006.12983*, 2020.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems*, 2012.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michael Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. A bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems*, 2012.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nissan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

A PEBBLE ALGORITHM

A state-of-the-art preference-based RL algorithm, PEBBLE (Lee et al., 2021a), consists of two main components: unsupervised pre-training and relabeling experiences. To collect diverse experience, PEBBLE pre-trains the policy by using intrinsic motivation (Oudeyer et al., 2007; Schmidhuber, 2010) in the beginning of training. Specifically, PEBBLE optimizes the policy to maximize the state entropy $\mathcal{H}(\mathbf{s}) = -\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\log p(\mathbf{s})\right]$ to efficiently explore the environment. Then PEBBLE learns the policy by using the state-of-the-art off-policy RL algorithm, SAC (Haarnoja et al., 2018). Since the learning process of off-policy algorithms with a non-stationary reward function can be unstable, PEBBLE stabilizes the learning process by relabeling all experiences in the buffer when the reward model is updated.

B EXPERIMENTAL DETAILS

Training details. We choose PEBBLE (Lee et al., 2021a) as a backbone algorithm of SURF, and use the hyperparameters in Table 1 for both PEBBLE and our method. For the reward model, we use a three-layer MLP with 256 hidden units and leaky ReLU activation. Following the implementation of PEBBLE (Lee et al., 2021a), we use an ensemble of three reward models and bound the output to $[-1, 1]$ using tanh function. Each model is trained by minimizing the cross-entropy loss using ADAM optimizer (Kingma & Ba, 2015) with the learning rate of 0.0003. For semi-supervised learning and data augmentation of SURF, we use hyperparameters in Table 2.

Sampling schemes. In preference-based RL methods, informative query sampling (Biyik & Sadigh, 2018; Biyik et al., 2020; Sadigh et al., 2017) has been adopted for improving the feedback-efficiency. For all experiments of PEBBLE and SURF, we use the disagreement-based sampling (Christiano et al., 2017) to choose queries for labeling: we first uniformly sample the initial batch of segments, and select $N_{\texttt{query}}$ pairs of segments[4] with high uncertainty based on the variance across ensemble of preference predictors $\{P_{\psi_i}[\sigma^1 \succ \sigma^0]\}_{i=1}^{N_{\texttt{en}}}$. Note that we use uniform sampling scheme for unlabeled samples, because the number of unlabeled samples are not limited. At each feedback session, we sample unlabeled samples as 10 times of labeled ones if the maximum budget of feedback is equal or larger than 1,000, and otherwise we sample unlabeled samples as 100 times of labeled ones.

Hyperparameter	Value	Hyperparameter	Value
Initial temperature	0.1	Hidden units per each layer	1024 (DMControl), 256 (Meta-world)
Length of segment	50	# of layers	2 (DMControl), 3 (Meta-world)
Learning rate	0.0003 (Meta-world) 0.0005 (Walker, Cheetah) 0.0001 (Quadruped)	Batch Size Optimizer	1024 (DMControl), 512 (Meta-world) Adam (Kingma & Ba, 2015)
Critic target update freq	2	Critic EMA τ	0.005
(β_1, β_2)	$(0.9, 0.999)$	Discount $\bar{\gamma}$	0.99
Frequency of feedback	5000 (Meta-world) 20000 (Walker, Cheetah) 30000 (Quadruped)	Maximum budget / # of queries per session	$1000/100, 100/10$ (DMControl) $10000/50, 4000/20$ (Meta-world) $2000/25, 400/10$ (Meta-world)
# of ensemble models $N_{\texttt{en}}$	3	# of pre-training steps	10000

Table 1: Hyperparameters of PEBBLE.

Hyperparameter	Value
Unlabeled batch ratio μ	4
Threshold τ	0.999 (Window Open, Sweep Into, Cheetah) 0.99 (others)
Loss weight λ	1
Min/Max length of cropped segment $[H_{\texttt{min}}, H_{\texttt{max}}]$	$[45, 55]$
Segment length before cropping	60

Table 2: Hyperparameters of SURF.

[4]We select 10% of the initial batch.

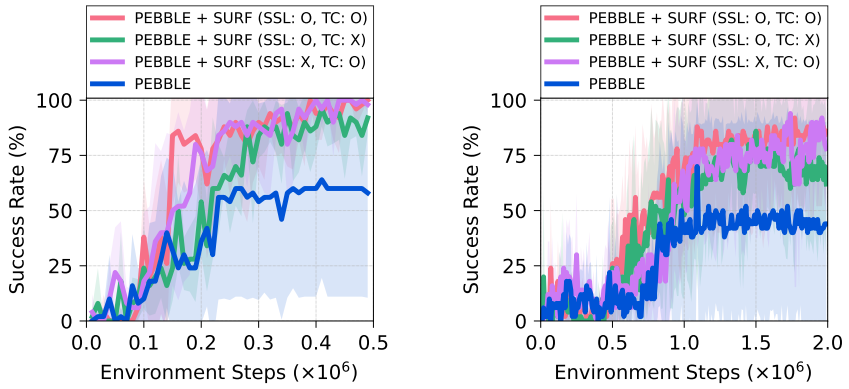
Implementation. We implement SURF using the publicly released implementation repository of the PEBBLE algorithm (https://github.com/pokaxpoka/B_Pref) with a full list of hyperparameters in Table 1. Note that DMControl environment depends on the MuJoCo simulator (Todorov et al., 2012), which is a commercial software. We follow the standard evaluation protocol for the locomotion tasks from DMControl. For robotic manipulation tasks from Meta-world, we measure the task success rate as defined by the authors. For each run of experiments, we utilize one Nvidia RTX 2080 Ti GPU and 4 CPU cores for training.

C ADDITIONAL EXPERIMENTAL RESULTS

Ablation study on Meta-world. We provide additional experimental results for component analysis on Meta-world (Yu et al., 2020). To evaluate the effect of each technique in SURF individually, we incrementally apply semi-supervised learning (SSL) and temporal cropping (TC) to our backbone algorithm, PEBBLE. Figure 7a and 7b show the learning curves of SURF on Window Open with 400 queries and Hammer with 10,000 queries, respectively. We observe that both semi-supervised learning (green) and data augmentation (purple) improve the baseline of PEBBLE (blue). Also, applying both of them further improves the performance (red). This shows that the key components of SURF are both effective.

Applying temporal cropping with other augmentations. In Section 5.3, we compared our method to random amplitude scaling (RAS) and adding Gaussian noise (GN) proposed in Laskin et al. (2020). To investigate if applying RAS or GN with the temporal cropping (TC) further improve the performance, we provide experimental results for the joint usage of the augmentations. In Figure 8a, we plot the learning curves of PEBBLE with various data augmentations on Walker-walk with 100 queries. We observe that the naive combination of two augmentations, i.e., RAS + TC and GN + TC, do not further improve the performance. Investigating how to combine several data augmentation methods for reward learning would be an interesting future direction.

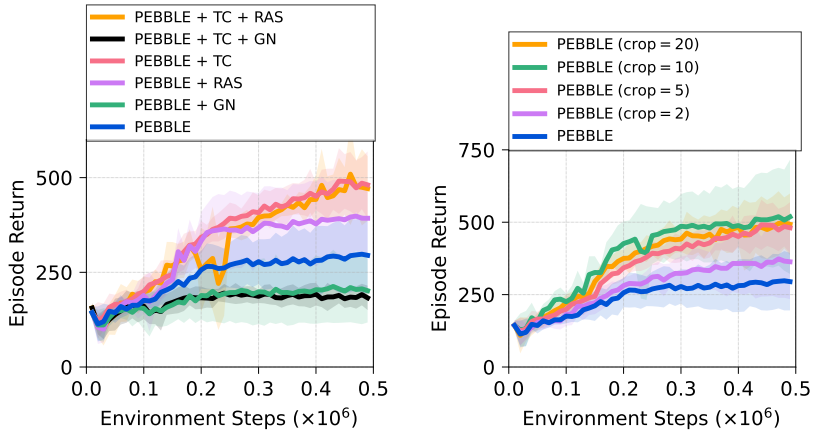
Effects of augmentation intensity. To investigate how does augmentation intensity affects the RL performance, we provide additional experimental results. In Walker-walk with 100 queries, we apply the temporal cropping to PEBBLE, with a varying cropping range, i.e., $(H_{\texttt{max}} - H_{\texttt{min}})/2$, from an array of $[2, 5, 10, 20]$. Figure 8b shows that temporal cropping consistently improves the performance of the PEBBLE. Although the performance with a large cropping range of 20 slightly underperforms the performance with 10, these results show that our augmentation method is quite robust to the choice of hyperparameters.



(a) Ablation study on Window Open

(b) Ablation study on Hammer

Figure 7: Contribution of each technique in SURF, i.e., semi-supervised learning (SSL) and temporal cropping (TC), in (a) Window Open, and (b) Hammer. The results show the mean and standard deviation averaged over five runs.



(a) Joint usage of the augmentations

(b) Effect of augmentation intensity

Figure 8: Analysis of the augmentation methods on Walker-walk task with 100 queries. (a) Comparison of the augmentation methods, i.e., applying random amplitude scaling (RAS) or Gaussian noise (GN) with temporal cropping (TC). (b) Comparison of the performance of PEBBLE with varying augmentation intensity, i.e., cropping range. The results show the mean and standard deviation averaged over five runs.