HINDSIGHT PRIORS FOR REWARD LEARNING FROM HUMAN PREFERENCES

Mudit Verma *
Arizona State University
Tempe, AZ, 85281
muditverma@asu.edu

Katherine Metcalf
Apple Inc.
Cupertino, CA, 95014
kmetcalf@apple.com

ABSTRACT

Preference based Reinforcement Learning (PbRL) removes the need to hand specify a reward function by learning a reward from preference feedback over policy behaviors. Current approaches to PbRL do not address the credit assignment problem inherent in determining which parts of a behavior most contributed to a preference, which result in data intensive approaches and subpar reward functions. We address such limitations by introducing a credit assignment strategy (Hindsight PRIOR) that uses a world model to approximate state importance within a trajectory and then guides rewards to be proportional to state importance through an auxiliary predicted return redistribution objective. Incorporating state importance into reward learning improves the speed of policy learning, overall policy performance, and reward recovery on both locomotion and manipulation tasks. For example, Hindsight PRIOR recovers on average significantly ($p < 0.05$) more reward on MetaWorld (20%) and DMC (15%). The performance gains and our ablations demonstrate the benefits even a simple credit assignment strategy can have on reward learning and that state importance in forward dynamics prediction is a strong proxy for a state's contribution to a preference decision.

1 INTRODUCTION

Preference-based reinforcement learning (PbRL) learns a policy from preference feedback removing the need to hand specify a reward function. Compared to other methods that avoid hand-specifying a reward function (e.g. imitation learning, advisable RL, and learning from demonstrations), PbRL does not require domain expertise nor the ability to generate examples of desired behavior. Additionally, PbRL can be deployed as human-in-the-loop allowing guidance to adapt on-the-fly to sub-optimal policies, and has shown to be highly effective for complex tasks where reward specification is not feasible (e.g. LLM alignment) Akrour et al. (2011); Ibarz et al. (2018); Lee et al. (2021a); Fernandes et al. (2023); Hejna III & Sadigh (2023); Lee et al. (2023); Korbak et al. (2023); Leike et al. (2018); Ziegler et al. (2019); Ouyang et al. (2022); Zhu et al. (2023). However, existing approaches to PbRL require large amounts of human feedback and are not guaranteed to learn well-aligned reward functions. A reward function is "well-aligned" when policy learned from it is optimal under the target reward function. We address the above limitations by incorporating knowledge about key states into the reward function objective.

Current approaches to learning a reward function from preference feedback do not impose a credit assignment strategy over how the reward function is learned. The reward function is learned such that preferred trajectories have a higher sum of rewards (returns) and consequentially are more likely to be preferred via a cross-entropy objective Christiano et al. (2017). Without imposing a credit assignment strategy to determine the impact of each state on the preference feedback, there are many possible reward functions that assign a higher return to the preferred trajectory. To select between possible reward functions large amounts of preference feedback are required. In the absence of enough preference labelled data, reward selection can become arbitrary, leading to misaligned reward functions. Therefore, we hypothesize that: (H1) guiding reward selection according to state importance will improve reward alignment and decrease the amount of preference feedback required

*Work during internship at Apple

to learn a well-aligned reward function and (H2) state importance can be approximated as the states that in hindsight are predictive of a behavior's trajectory.

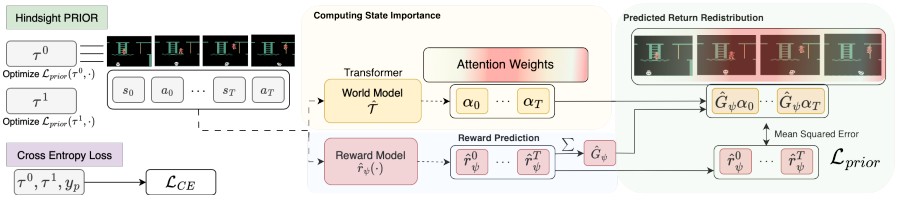

Figure 1: Hindsight PRIOR augments the existing PbRL cross-entropy loss by encouraging the magnitude of a reward to be proportional to the state's importance. Each reward update preference labelled trajectories are passed to a world model $\hat{\mathcal{T}}$ (yellow) and estimated reward $\hat{r}_\psi$ (red), which assign an importance score and a reward (respectively) to each state-action pair. The return $\hat{G}_\psi$ is then applied to the importance scores, which then serve as auxiliary targets for reward learning.

To this end, we introduce *PRIor On Reward* (PRIOR), a PbRL method that guides credit assignment according to estimated state importance. State importance is approximated with an attention-based world model. The reward objective is augmented with state importance as an inductive bias to disambiguate between the possible rewards that explain a preference decision. In contrast to previous work, our contribution mitigates the credit assignment problem, which decreases the amount of feedback needed while improving policy and reward quality. In particular, compared to baselines, Hindsight PRIOR achieves $\geq$ 80% success rate with as little as half the amount of feedback on MetaWorld and recovers on average significantly ($p < 0.05$) more reward on MetaWorld (20%) and DMC (15%). Additionally, Hindsight PRIOR is more robust in the presence of incorrect feedback.

2 Related Work

PbRL (Wirth et al., 2017), train RL agents with human preferences on tasks for which reward design is non-trivial and can introduce inexplicable and unwanted behavior (Vamplew et al., 2018; Krakovna et al., 2020). Christiano et al. (2017) extended PbRL to Deep RL and PEBBLE (Lee et al., 2021a) incorporated unsupervised pre-training, reward relabelling, and offline RL to reduce sample complexity. Subsequent works extended PEBBLE by incorporating pseudolabelling into the reward learning process (Park et al., 2022), guiding exploration with reward uncertainty (Liang et al., 2022), and monitoring Q-function performance on the preference feedback (Liu et al., 2022).

Kim et al. (2023) attempts to address the credit assignment problem by assuming that the preference feedback is based on a weighted sum of rewards and use a modified transformer architecture to assign rewards and weights to each state-action pair. However, introducing a transformer-based reward function increases reward complexity compared to earlier work and Hindsight PRIOR as well as tying the reward model to a specific architecture. While Hindsight PRIOR also uses a transformer architecture, it is independent of the reward architecture. Additionally, Kim et al. (2023) has not been extended to online RL.

Learning World Models: Reinforcement Learning, especially model-based RL, leverage learned world models for tasks such as planning (Allen & Koomen, 1983; Hafner et al., 2019b;a), data augmentation (Gu et al., 2016; Ball et al., 2021), uncertainty estimation (Feinberg et al., 2018; Kalweit & Boedecker, 2017), and exploration (Ladosz et al., 2022). In this work we learn a world model and use it to estimate the importance of state-action pairs. While Hindsight PRIOR can use any transformer-based world-model, we use the current state of the art in terms of sample complexity, Transformer-based World Models (TWM) (Robine et al., 2023). To our knowledge, existing work has not incorporated a world model in reward learning from preferences.

Feature Importance: Many methods exist to estimate the importance of different parts of an input to the model decision-making process. Some popular methods include gradient / saliency based approaches (Greydanus et al., 2018; Selvaraju et al., 2017; Simonyan et al., 2013; Weitkamp et al., 2019) and self-attention based methods Ras et al. (2022); Wiegreffe & Pinter (2019); Vashishth et al. (2019). Self-attention based methods have been used for video summarization and extraction of key

frames (Feng et al., 2020; Bilkhu et al., 2019; Apostolidis et al., 2021; Liu et al., 2019). Given our use of TWM,, we use a self-attention map based method.

Credit Assignment: Credit assignment challenges typically stem from sparse rewards and large state spaces and solutions aim to boost policy learning (Ke et al., 2018; Goyal et al., 2018; Ferret et al., 2020) Past works like Goyal et al. (2018) have learned a backward dynamics model to sample states that could have led to the current state and Ferret et al. (2020) equips a non-autoregressive sequence model to reconstruct a reward function and utilizes model attention for credit assignment. Return redistribution is another credit assignment solution that redistribute the ground-truth, non-stationary reward signal in order to denisfy the emitted reward signals (Ren et al., 2021; Arjona-Medina et al., 2019; Patil et al., 2020). This is in contrast to PbRL where predicted rewards are dense to begin with. We adapt the idea of return redistribution for PbRL by redistributing the predicted returns and is discussed in Section 4.3.

3 Preference-based Reinforcement Learning

To learn a policy with preference-based reinforcement learning (PbRL), the policy π_ϕ executes an action a_t at each time step t in environment $\mathcal{E}$ based on its observation o_t of the environment state s_t. For each action the environment $\mathcal{E}$ transitions to a new state s_{t+1} according to transition function $\mathcal{T}$ and emits a reward signal $\hat{r}_t = \hat{r}_\psi(s_t, a_t)$. The policy π_ϕ is trained to take actions that maximize the expected discounted return $\hat{G}_\psi = \sum_t \gamma \hat{r}_\psi(s_t, a_t)$. The reward $\hat{r}_\psi(\cdot)$ is trained to approximate the human's target reward function $\bar{r}_\psi(\cdot)$.

To learn $\hat{r}_\psi(\cdot)$ a dataset $\mathcal{D}$ of preference triplets (τ_0, τ_1, y_p) is collected from a teacher (human or synthetic) over the course of policy training. The preference label y_p indicates which, if any, of the two trajectory segments τ_0 or τ_1 with length l has a higher (discounted) return G_ψ under the target reward function $\bar{r}_\psi(\cdot)$. Following Park et al. (2022) and Lee et al. (2021a), feedback is solicited every K steps of policy training for the M *maximally informative* trajectories pairs (τ_0, τ_1) (e.g. pairs with the largest $\hat{r}_\psi(\cdot)$ uncertainty).

Given a preference dataset $\mathcal{D}$, $\hat{r}_\psi(\cdot)$ is learned such that preferred trajectories have higher predicted returns $\hat{G}_\psi$ than dispreferred trajectories. Using the Bradley-Terry model (Bradley & Terry, 1952), predicted trajectory returns are used to compute the probability that one trajectory is preferred over the other P_ψ:

$$P_\psi[\tau^0 \succ \tau^1] = \frac{\exp \sum_t \hat{r}_\psi(s_t^0, a_t^0)}{\sum_{i \in \{0,1\}} \exp \sum_t \hat{r}_\psi(s_t^i, a_t^i)}, \tag{1}$$

where τ_0 is preferred over τ_1. The probability estimate P_ψ is then used to compute and minimize the cross-entropy between the predicted and the true preference labels:

$$\mathcal{L}_{CE} = \underset{(\tau_0, \tau_1, y_p) \sim \mathcal{D}}{-\mathbb{E}} [y_p(0) \log P_\psi[\tau_0 \succ \tau_1] + y_p(1) \log P_\psi[\tau_1 \succ \tau_0]]. \tag{2}$$

The reward function $\hat{r}_\psi$ is learned over the course of policy π_ϕ training by iterating between updating π_ϕ according to the current estimate of $\bar{r}_\psi$ and updating $\hat{r}_\psi$ on $\mathcal{D}$, which is grown by M preference triplets sampled from π_ϕ's experience replay buffer $\mathcal{B}$ for each $\hat{r}_\psi$ update. To avoid training π_ϕ on a completely random $\hat{r}_\psi$ at the start of training, π_ϕ explores the environment to populate $\mathcal{D}$ with an initial set of trajectories following either a random policy or during an intrinsically motivated pre-training period Christiano et al. (2017); Lee et al. (2021a).

4 Hindsight PRIORs

PbRL relies on learning a high-quality reward function that generalizes and quickly adapts in a few-shot manner to unseen portions of the environment, and given its human in the loop nature, reducing the amount of preference feedback is vital. To learn the reward function $\hat{r}_\psi(s_t, a_t)$, trajectory-level feedback is provided and then is distributed to each of the trajectory's states-action pairs. Given two trajectories, a return per trajectory $(\hat{G}_\psi^0, \hat{G}_\psi^1)$, and a preference label, many reward functions assign a higher return for preferred trajectories, but do not align with the target reward function $\bar{r}_\psi(s_t, a_t)$ on unseen data. With a large enough dataset, a $\hat{r}_\psi(\cdot)$ that aligns with human preferences in all portions of the environment can be learned. However, given a set of reward functions $\hat{r}_\psi(\cdot)$,

each of which conforms to the preference dataset, a reward function will be arbitrarily selected in the absence of additional information or constraints. From insufficient preference feedback, the selected $\hat{r}_\psi(\cdot)$ is likely to represent a local minimum with respect to previously unseen trajectories, where the assigned returns $\hat{R}_\psi$ are correct, but the distribution of rewards within trajectories are incorrect. Incorrectly assigning rewards at the state-action level, or incorrectly solving the credit assignment problem, leads to reward functions that do not generalize outside of the preference dataset, resulting in suboptimal policies relative to the target reward function. Thus, we address the credit assignment problem and guide reward distribution within a trajectory through an auxiliary objective that provides a prior on state-action pair values computed after the trajectory has been observed (in hindsight).

The priors on state-action values are identified by answering the following question, "now that I have seen what happened, which state-action pairs best summarize what happened in the given trajectory?" We consider the states that summarize a trajectory to be those that are most predictive of future state-action pairs. The most predictive states are then used as a proxy for the most important states.The use of summarizing state-action pairs is motivated by previous work demonstrating that people have selective attention when evaluating a behavior – they attend only to the state-action pairs necessary to provide the evaluation (Desimone & Duncan, 1995; Bundesen, 1990; Ke et al., 2018). We therefore assign greater credit to those states that were likely to have been attended to and therefore influenced the preference feedback. As summarizing states are those that are predictive of future state-action pairs, we identify them using an attention-based forward dynamics model, where state-action pair importance is proportional to their weight in the attention layers. For example, in Figure 1 the important states (highlighted in red) identified from an action sequences in Montezuma's Review are those where the agent lines up to leap from the platform.

4.1 Approximating State Importance with Forward Dynamics

An attention-based forward dynamics model (Figure 1 yellow) is used to identify important (summarizing) states and address the PbRL credit assignment problem. The states that are key for a forward dynamics model to predict the future are assumed to be similar to those a human evaluator would use to predict future states, and thus summarize a trajectory. We use the attention layers in an attention-based forward dynamics model to approximate human attention and guide how feedback credit is distributed across a trajectory. In similar vein as Harutyunyan et al. (2019)'s State Conditioned Hindsight Credit Assignment, we consider the importance of a state in a trajectory given that a future state was reached.

World models have played a large role in model-based reinforcement learning. Given the power that recent work have shown them to convey in reinforcement learning (Manchin et al., 2019; Hafner et al., 2019a; Hu et al., 2019), we use world modelling techniques to learn an attention-based forward dynamics model. For a world model $\hat{\mathcal{T}}$ to identify important states and approximate human attention, it must have two characteristics. First, it must model environment dynamics and be able to predict the next future state $\hat{s}_T$ given a history of state-action pairs $\tau_{[1:T-1]}$: $\hat{\mathcal{T}}(\tau_{[1:T-1]}) = \hat{s}_T$. Second, it must expose a mechanism to compute state-action importance $\alpha_{[1:T-1]}$ vector over a given trajectory segment $\tau_{[1:T-1]}$ when performing the next-state prediction: $\hat{\mathcal{T}}(\tau_{[1:T-1]}, \hat{s}_T) = \alpha_{[1:T-1]}$. *Transformer based World Models* (TWM) Robine et al. (2023) meets both requirements in addition to being sample efficient (Robine et al., 2023; Micheli et al., 2023).

TWM is a Transformer XL based auto-regressive dynamics model $\hat{\mathcal{T}}$ that predicts the reward $\hat{r}_t$, discount factor $\hat{\gamma}_t$, and (latent) next state $\hat{z}_{t+1}$ given a history of state-action pairs ($\mathcal{W}(\tau_{[1:h]}) = s_h$). In the PbRL paradigm, predicting a transition's reward r_t is impractical as the reward function $\hat{r}_\psi$ is learned in conjunction with the world model. Therefore, we adapt TWM by removing the reward and discount heads, and use the observation and latent state models:

1. Observation Encoder and Decoder: $z_t \sim p_\mu(z_t|o_t)$; $\hat{o}_t \sim p_\mu(\hat{o}_t|z_t)$
2. Aggregation and Latent State Predictor: $h_t = f_\omega(z_{[1:t]}, a_{[1:t]})$; $\hat{z}_{t+1} \sim p_\omega(\hat{z}_{t+1}|h_t)$

Consequentially, the loss function for the dynamics model is updated as follows, where H is the cross entropy between the predicted and true latent next states:

$$\mathcal{L}_\omega^{\text{Dyn.}} = \mathbb{E}[\sum_{t=1}^{T} H(p_\mu(z_{t+1}|o_{t+1}), p_\omega(\hat{z}_{t+1})|h_t)]. \tag{3}$$

The *Latent State Predictor* is a transformer responsible for predicting the forward dynamics given the trajectory history, and is therefore responsible for approximating state-action importance. For a description of the latent state predictor and its architecture, specifically the parts that allow us to extract state importance, see Appendix C.2.

The world model is learned over the course of policy π_ϕ and reward $\hat{r}_\psi$ training. The observation encoder's and decoder's weights μ are trained during π_ϕ's exploration period to initially populate $\mathcal{D}$, and then frozen for the remainder of π_ϕ and $\hat{r}_\psi$ training. The weights of the dynamics model ω are trained during π_ϕ's exploration phase and then updated every j steps of policy training from the same replay buffer $\mathcal{B}$ the preference queries are sampled from. Using $\mathcal{B}$ removes the need to sample additional transitions or trajectories for the purpose of world model learning.

4.2 Computing the Hindsight PRIORs

The use of a transformer-based *Latent State Predictor* provides approximations of state importance in the form of attention weights (our second requirement in Section 4.1). When updating $\hat{r}_\psi$ the attention weights for each trajectory τ in the collected preference triplets $(\tau_0, \tau_1, y_p) \in \mathcal{D}$ are computed by passing τ to the Transformer XL model $\hat{\mathcal{T}}$ (Figure 1 yellow). The transformer uses a multi-headed, multi-layer attention mechanism, where H is the number of attention heads, L the number of layers, and $attn_t^l = (attn_{s_t}^l, attn_{a_t}^l) \in \mathcal{A}^{2T \times L}$ the attention weights of the l-th layer for state-action pair $(s_t, a_t) \in \tau_{1:T}$. The matrix $\mathcal{A}$ denotes the attention distribution in predicting the next state $\hat{z}_{T+1} = \hat{\mathcal{T}}(\tau)$ across all sequence timesteps and attention layers. The hindsight PRIOR (importance) α_t for a given state-action pair (s_t, a_t) is estimated as the mean across layers L at timestep t, $\alpha_t = {}^1/_L \sum_{l=1}^{L} attn_t^l$.

4.3 Reward Redistribution and Constructing the Hindsight PRIOR Loss

To guide reward function learning according to state-action pair importance, the attention maps $\mathcal{A}$ from $\hat{\mathcal{T}}$ are incorporated into the reward learning objective as redistribution guidance (Figure 1 orange). The attention map does not form a reward target, as state-action importance for predicting future states does not equate absolute value in the target reward function, therefore return redistribution (Arjona-Medina et al., 2019), a strategy typically used to address the challenge of delayed returns in reinforcement learning, is used to align reward assignment with state-action importance.

Return redistribution addresses the challenge of delayed returns by redistributing a trajectory segment's return among its constituent state-action pairs. The return redistribution use case in existing work (Arjona-Medina et al., 2019; Ren et al., 2021; Patil et al., 2020) relied on known and typically stationary, but sparse, rewards. In PbRL, while the learned reward function is dense, the feedback used to learn it occurs at the end of a trajectory and therefore is delayed and sparse. Therefore, to align rewards with estimated state importance, we introduce *predicted* return $\hat{G}_\psi$ redistribution to obtain state-action pair importance conditioned reward targets for a given trajectory τ, where $\hat{G}_\psi = \sum_t^T \hat{r}_\psi(\tau_t)$.

To obtain the reward targets for each trajectory τ in a preference triplet (τ_0, τ_1, y_p), the predicted return $\hat{G}_\psi$ is computed (Figure 1 red), the attention map $\mathcal{A}(\tau) \sim \hat{\mathcal{T}}(\tau)$ is extracted from the world model (Figure 1 yellow), and the mean attention value per state-action pair is taken over layers $\boldsymbol{\alpha} = \frac{1}{L} \sum_{l=1}^{L} (attn_{s_t}^l + attn_{a_t}^l)$. Reward value targets are then estimated by redistributing the predicted return $\hat{G}_\psi$ according to α to obtain $\mathbf{r}_{target} = \boldsymbol{\alpha} \odot \hat{G}_\psi$, where $\boldsymbol{\alpha}$ is a vector with length $|\tau|$ and $\hat{G}_\psi$ a scalar (Figure 1 orange). The state-action pair importance conditioned reward targets $\mathbf{r}_{target}$ are incorporated into reward learning via an auxiliary mean squared error loss between the predicted rewards $\hat{\mathbf{r}}_\psi = [\hat{r}_\psi(s_1, a_1), \hat{r}_\psi(s_2, a_2), ..., \hat{r}_\psi(s_T, a_T)]$ and $\mathbf{r}_{target}$:

$$\mathcal{L}_{prior} = MSE(\hat{\mathbf{r}}_\psi, \mathbf{r}_{target}). \tag{4}$$

The PbRL objective $\mathcal{L}_{CE}$ (Equation 2) is modified to be a linear combination of the proposed hindsight PRIOR loss $\mathcal{L}_{prior}$ to guide reward learning with both preference feedback and estimated

state-action importance:

$$\mathcal{L}_{pbrl}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\mathcal{L}_{CE}(\mathcal{D}_i) + \lambda * \mathcal{L}_{prior}(\mathcal{D}_i), \tag{5}$$

where λ is a constant to ensure $\mathcal{L}_{CE}$ and $\mathcal{L}_{prior}$ are on the same scale.

5 Empirical Evaluation

We evaluate the benefits of Hindsight PRIOR on the Deep Mind Control (DMC) Suite locomotion (Tunyasuvunakool et al., 2020) and MetaWorld control (Yu et al., 2020) tasks, compare against baselines (Lee et al., 2021a; Park et al., 2022; Liu et al., 2022; Liang et al., 2022), and ablate over Hindsight PRIOR's contributions. Following our baselines, tasks with hand-coded rewards are used to assess algorithm performance. The hand-coded rewards serve as the target reward functions (used by human in the loop) and are used to assign synthetic preference feedback (trajectories with the higher return are preferred). Therefore, PbRL policy performance is measure and compared according to how well and how quickly the target reward function is maximized. Additionally, a SAC (Haarnoja et al., 2018) policy is trained on the target reward function to provide a reasonable reference point for PbRL performance. Each PbRL method is compared to SAC using mean normalized return for DMC Lee et al. (2021b) and mean normalized success rate for MetaWorld. See Appendix F for the equations. For each comparison against baselines, mean (+standard deviation) policy learning curves and normalized returns are reported over 5 random seeds (see Appendix E). From the learning curves and normalized scores, feedback sample efficiency, environment interaction sample efficiency, and reward recovery are compared between Hindsight PRIOR and baselines.

While using synthetic feedback allows us to directly compare between the target $\bar{r}_\psi$ and learned $\hat{r}_\psi$ reward functions, humans do not always select the trajectory that maximizes the target reward function. Occasionally, humans will mislabel a trajectory pair and flip the preference ordering. Therefore, we evaluate Hindsight PRIOR and PEBBLE (the backbone algorithm for Hindsight PRIOR and the baselines) using a synthetic feedback labeller that provides incorrect feedback on a percentage (10%, 20%, 40%) of the preference triplets (mistake labeller from (Lee et al., 2021b)).

To better understand Hindsight PRIOR's performance gains over baselines (Section 5.1), we answer the following questions in Section 5.2:

- (Q1) Is it the use of a return redistribution strategy versus Hindsight PRIOR's specific strategy (guiding return redistribution according to state importance) that leads to the performance improvements?
- (Q2) Do the performance gains stem from incorporating environment dynamics?
- (Q3) What types of states does TWM identify as important?

and to verify that the incorporation of the world model does not negatively impact PbRL capabilities, we answer the following in Section 5.3:

- (Q4) Does Hindsight PRIOR's scale to longer trajectories in the preference triplets?
- (Q5) Does combining Hindsight PRIOR with a complementary baseline improve performance?
- (Q6) Does Hindsight PRIOR allow for the removal of preference feedback?

Hindsight PRIOR and all baselines extend PEBBLE as their underlying PbRL algorithm. The policy takes random actions for the first 1k steps of policy training and then trains with an intrinsically-motivated reward (as suggested by Lee et al. (2021a)) for 9k steps. The experimental set up and task configurations are selected following Park et al. (2022) which is the existing state of the art method. Algorithm-specific hyper-parameters match those used by the corresponding paper and hyper-parameters determining feedback schedules and amounts match those used in Park et al. (2022) (see Appendix E).

5.1 Comparing Against PbRL Baselines

Figure 2 and Table 1 compare the performance of Hindsight PRIOR to PEBBLE, SURF Park et al. (2022), RUNE Liang et al. (2022), and MRN Liu et al. (2022) with perfect feedback. The amount

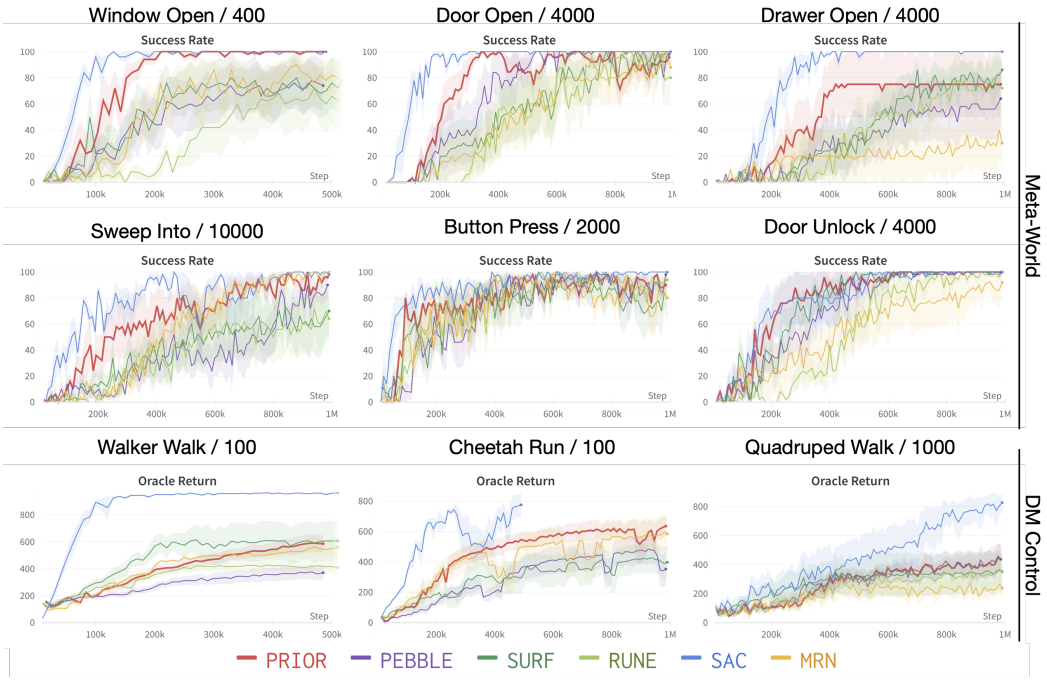

Figure 2: PbRL and SAC policy learning curves for six MetaWorld (top and middle rows) and three DMC (bottom row) tasks. Each experiment is specified as: task / feedback amount.

Table 1: Mean ($\pm$ variance) normalized success rates (MetaWorld) and normalized returns (DMC) across tasks.

Suite	PRIOR	PEBBLE	SURF	RUNE	MRN
MetaWorld	**0.66$\pm$ 0.002**	0.56 $\pm$ 0.007	0.56 $\pm$ 0.004	0.53 $\pm$ 0.009	0.55 $\pm$ 0.005
DMC	**0.59$\pm$ 0.003**	0.48 $\pm$ 0.018	0.55 $\pm$ 0.019	0.49 $\pm$ 0.007	0.53 $\pm$ 0.002

of feedback is held fixed across methods for a given task and is provided every 5k steps of policy training (X-axis), therefore learning curve performance in Figure 2 relative to the number of policy steps indicates both reward and policy sample complexity. For example, at policy step 30k for `walker-walk`, the preference dataset contains 10 preference triplets and 20 at 50k steps. Table 1 reports the mean normalized return and success rate for each algorithm across tasks and shows that Hindsight PRIOR has the best overall performance across tasks. A two-tailed paired t-test with dependent means was performed over the normalized returns and success rates to determine that Hindsight PRIOR's performance gains are statistically significant. (Appendix F for t and p-scores. Task specific normalized returns and success rates are reported in Appendix F).

For all tasks, Hindsight PRIOR matches or exceeds baseline performance, and for all except `quadruped-walk`, either converges to a higher performance point (e.g. 100% versus 80% success rate on `window-open`) or requires significantly less preference labels to achieve the same performance point (e.g. 100% success rate at $\sim$ 350k policy steps versus $\sim$ 550k for `door-open`). The results suggest that Hindsight PRIOR's credit assignment strategy improves PbRL beyond guiding exploration with reward uncertainty (Liang et al., 2022), increasing the amount of preference feedback through pseudo-labelling (Park et al., 2022), and incorporating information about policy performance in reward learning (Liu et al., 2022).

Figure 3 (returns left and success rates center) shows the performance differences for PEBBLE (Lee et al., 2021a) and Hindsight PRIOR on `window-open` across different amounts of preference feedback mistakes. The mistake amounts are percentages of the maximum feedback amount, specifically 0% (perfect labeller), 10%, 20%, and 40%. We compare against PEBBLE, because it has compara-

ble performance to the baselines (Figure 2 and Table 1) and is the underlying PbRL algorithm. For all mistake amount conditions, Hindsight PRIOR outperforms PEBBLE. Furthermore, Hindsight PRIOR trained on a dataset with 20% labelling errors beats the performance of PEBBLE with no labelling errors. The results suggest that the inclusion of a credit assignment strategy, specifically one guided by estimated state importance, makes reward and policy learning more robust to preference feedback labelling errors.

5.2 Understanding the Performance Gains

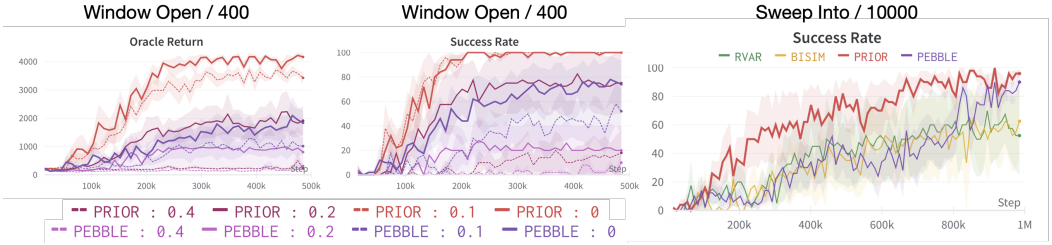

Figure 3: PbRL learning curves over different labelling mistake amounts (left & center : purple & pink for PEBBLE and red & magenta for PRIOR), and different methods for return distribution and dynamics-aware rewards (right).

In order to better understand sources of Hindsight PRIOR's performance gains we evaluate the importance of the state-importance guided return redistribution strategy by comparing against different redistribution strategies (Q1), assess the impact of Hindsight PRIOR making reward learning dynamics aware by replacing L_{prior} with an adapted bisimulation objective (Kemertas & Aumentado-Armstrong, 2021) (Q2), and qualitatively assess what the world model $\hat{\mathcal{T}}$ identifies as important states (Q3). The results show the benefits of the forward dynamics based state-importance redistribution strategy, demonstrate that Hindsight PRIOR's contributions extend beyond making reward learning dynamics aware, and that $\hat{\mathcal{T}}$'s attention weight identify reasonable state-action as important.

We compare against PEBBLE as it has comparable performance to the baselines (Figure 2 and Table 1) and is the underlying PbRL algorithm for all baselines.

Redistribution Strategy (Q1): Hindsight PRIOR's redistribution strategy is compared against an uninformed return redistribution strategy, using the mean attention weights $\boldsymbol{\alpha}$ serve as the reward targets (RVAR). The uniform strategy corresponds to assigning uniform importance to each state-action pair in a trajectory and each state-action pair is assumed to equally contribute to the preference feedback. The uniform strategy adapts Ren et al. (2021)(RRD) to obtain the reward target R_{target} by setting $\alpha_t = \frac{1}{T}$. Figure 3 (right - green) shows that while uniform predicted return redistribution is on par with PEBBLE (and in some cases better, see Appendix G.1), Hindsight PRIOR is superior in feedback and environment sample efficiency.

Given Hindsigh PRIOR's performance relative to a uniform redistribution strategy, we amplify Hindsight PRIOR's attention weights through a min-max normalization of the attention map followed by a softmax (NRP). Amplifying the attention map moves it further from the uniform redistribution strategy and potentially improves it. However, Hindsight PRIOR and NPR have comparable performance (Figure 6 in Appendix G.3) showing that explicitly discouraging a uniform redistribution strategy is not necessary.

Dynamics Aware Reward Learning (Q2): While Hindsight PRIOR does not directly use the forward dynamics of the world model $\hat{\mathcal{T}}$, knowledge of transition dynamics influence how the reward function is learned. Therefore, we assess the contribution of dynamics-aware reward learning in the absence of a return redistribution credit assignment strategy. To incorporate dynamics, a bisimulation-metric representation learning objective, which has been used as a data-efficient approach for policy learning, is incorporated into reward learning. See Appendix G.2 for details on incorporating the bisimulation auxiliary encoder loss Kemertas & Aumentado-Armstrong (2021) into Hindsight PRIOR.

The results show that making reward learning dynamics aware improves policy learning (Figure 3 (right-yellow)) compared to PEBBLE,but *not* compared to Hindsight PRIOR. Therefore, while incorporating of environment dynamics into reward learning explains part of Hindsight PRIOR's performance gains, it does not explain all of the performance gains highlighting the importance of Hindsight PRIOR's credit assignment strategy.

Examining Important States (Q3): Fig. 1 shows the attention over a trajectory snippet from Montezuma's Revenge (analysis in App. I). In our qualitative experiments with discrete domains of Atari Brockman et al. (2016) and control based domains of Metaworld Yu et al. (2020) we found a significant overlap between important states for future state prediction and underlying task.

5.3 Assessing Scalability and Compatibility

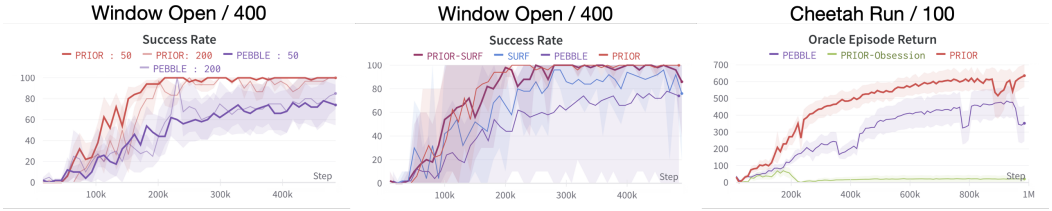

Figure 4: Learning curves evaluating different trajectory lengths (left), combining Hindsight PRIOR with SURF (center), and removing the influence of preference feedback (right).

Scalability (Q4): Since Hindsight PRIOR subroutines a forward dynamics model to obtain the attention map $\mathcal{A}$ we evaluate whether it can identify important states in longer trajectories that provide more context for human evaluators. Figure 4 (left) and Appendix H show that, following similar trends as PEBBLE, Hindsight PRIOR's performance is consistent given a 4x increase in trajectory length (50 versus 200 query length).

Combining with PEBBLE Extensions (Q5): We investigate the benefits of Hindsight PRIOR when used in parallel with another sample-efficient PbRL techniques, like SURF (Park et al., 2022). Figure 4 (center) shows combining Hindsight PRIOR with SURF (Park et al., 2022) improves policy performance relative to PEBBLE and SURF, but provides no real gain relative to Hindsight PRIOR alone.

Removing Preference Feedback (Q6): The results in Figure 4 (right) show the impact of making λ very large (green) in Equation 5 resulting in a reward function that is learned solely from $\mathcal{L}_{prior}$. The inability of Hindsight PRIOR to learn anything with a very large λ verifies that focusing the reward signal around important states is not sufficient for policy learning.

6 Conclusion

We have presented Hindsight PRIOR, a novel technique to guide credit-assignment during reward learning in PbRL that significantly improves both policy performance and learning speed by incorporating state importance into reward learning. We use the attention weights of a transformer-based world model to estimate state importance and guide predicted return redistribution to be proportional to state importance. The redistributed prediction rewards are then used as an auxiliary target during reward learning. We present results from extensive experiments on complex robot arm manipulation and locomotion tasks and compare against state of the art baselines to demonstrate the impact of Hindsight PRIOR and the importance of addressing the credit assignment problem in reward learning.

Limitations & Future Work: Hindsight PRIOR greatly improves PbRL and our qualitative assessment shows that the selected important states are reasonable. However, it relies on the assumption that states that are important to the world model are also important to an arbitrary human. Different humans might attribute importance to different states. Future work will investigate the alignment between the world model's important states and those people focus on when providing preference feedback as well investigation the personalization aspects of important state identification.

REFERENCES

Riad Akrour, Marc Schoenauer, and Michele Sebag. Preference-based policy learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 12–27. Springer, 2011.

James F Allen and Johannes A Koomen. Planning using a temporal world model. In *Proceedings of the Eighth international joint conference on Artificial intelligence-Volume 2*, pp. 741–747, 1983.

Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining global and local attention with positional encoding for video summarization. In *2021 IEEE international symposium on multimedia (ISM)*, pp. 226–234. IEEE, 2021.

Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Philip J Ball, Cong Lu, Jack Parker-Holder, and Stephen Roberts. Augmented world models facilitate zero-shot dynamics generalization from a single offline environment. In *International Conference on Machine Learning*, pp. 619–629. PMLR, 2021.

Manjot Bilkhu, Siyang Wang, and Tushar Dobhal. Attention is all you need for videos: Self-attention based video summarization using universal transformers. *arXiv preprint arXiv:1906.02792*, 2019.

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.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Claus Bundesen. A theory of visual attention. *Psychological review*, 97(4):523, 1990.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value expansion for efficient model-free reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, 2018.

Xuming Feng, Lei Wang, and Yaping Zhu. Video summarization with self-attention based encoder-decoder framework. In *2020 International Conference on Culture-oriented Science & Technology (ICCST)*, pp. 208–214. IEEE, 2020.

Patrick Fernandes, Aman Madaan, Emmy Liu, António Farinhas, Pedro Henrique Martins, Amanda Bertsch, José GC de Souza, Shuyan Zhou, Tongshuang Wu, Graham Neubig, et al. Bridging the gap: A survey on integrating (human) feedback for natural language generation. *arXiv preprint arXiv:2305.00955*, 2023.

Johan Ferret, Raphaël Marinier, Matthieu Geist, and Olivier Pietquin. Self-attentive credit assignment for transfer in reinforcement learning. 2020.

Anirudh Goyal, Philemon Brakel, William Fedus, Soumye Singhal, Timothy Lillicrap, Sergey Levine, Hugo Larochelle, and Yoshua Bengio. Recall traces: Backtracking models for efficient reinforcement learning. *arXiv preprint arXiv:1804.00379*, 2018.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International conference on machine learning*, pp. 1792–1801. PMLR, 2018.

Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In *International conference on machine learning*, pp. 2829–2838. PMLR, 2016.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32, 2019.

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

Hangkai Hu, Shiji Song, and Gao Huang. Self-attention-based temporary curiosity in reinforcement learning exploration. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(9): 5773–5784, 2019.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

Gabriel Kalweit and Joschka Boedecker. Uncertainty-driven imagination for continuous deep reinforcement learning. In *Conference on Robot Learning*, pp. 195–206. PMLR, 2017.

Nan Rosemary Ke, Anirudh Goyal ALIAS PARTH GOYAL, Olexa Bilaniuk, Jonathan Binas, Michael C Mozer, Chris Pal, and Yoshua Bengio. Sparse attentive backtracking: Temporal credit assignment through reminding. *Advances in neural information processing systems*, 31, 2018.

Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. *Advances in Neural Information Processing Systems*, 34:4764–4777, 2021.

Changyeon Kim, Jongjin Park, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Preference transformer: Modeling human preferences using transformers for rl. *arXiv preprint arXiv:2303.00957*, 2023.

D. Kingma and J. Ba. ADAM: A method for stochastic optimization. volume 3, 2015.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pp. 17506–17533. PMLR, 2023.

V Krakovna, J Uesato, V Mikulik, et al. Specification gaming: The flip side of ai ingenuity—deepmind, 2020.

Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021a.

Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*, 2021b.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

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.

Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. Reward uncertainty for exploration in preference-based reinforcement learning. *arXiv preprint arXiv:2205.12401*, 2022.

Runze Liu, Fengshuo Bai, Yali Du, and Yaodong Yang. Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:22270–22284, 2022.

Yen-Ting Liu, Yu-Jhe Li, Fu-En Yang, Shang-Fu Chen, and Yu-Chiang Frank Wang. Learning hierarchical self-attention for video summarization. In *2019 IEEE international conference on image processing (ICIP)*, pp. 3377–3381. IEEE, 2019.

Anthony Manchin, Ehsan Abbasnejad, and Anton Van Den Hengel. Reinforcement learning with attention that works: A self-supervised approach. In *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part V 26*, pp. 223–230. Springer, 2019.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=vhFu1Acb0xb`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. *arXiv preprint arXiv:2203.10050*, 2022.

Vihang P Patil, Markus Hofmarcher, Marius-Constantin Dinu, Matthias Dorfer, Patrick M Blies, Johannes Brandstetter, Jose A Arjona-Medina, and Sepp Hochreiter. Align-rudder: Learning from few demonstrations by reward redistribution. *arXiv preprint arXiv:2009.14108*, 2020.

Gabrielle Ras, Ning Xie, Marcel Van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research*, 73:329–396, 2022.

Zhizhou Ren, Ruihan Guo, Yuan Zhou, and Jian Peng. Learning long-term reward redistribution via randomized return decomposition. *arXiv preprint arXiv:2111.13485*, 2021.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*, 2023.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology*, 20(1):27–40, 2018.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. Attention interpretability across nlp tasks. *arXiv preprint arXiv:1909.11218*, 2019.

Laurens Weitkamp, Elise van der Pol, and Zeynep Akata. Visual rationalizations in deep reinforcement learning for atari games. In *Artificial Intelligence: 30th Benelux Conference, BNAIC 2018,'s-Hertogenbosch, The Netherlands, November 8–9, 2018, Revised Selected Papers 30*, pp. 151–165. Springer, 2019.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

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*, pp. 1094–1100. PMLR, 2020.

Banghua Zhu, Jiantao Jiao, and Michael I Jordan. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*, 2023.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

APPENDIX

A DOMAINS FOR EMPIRICAL EVALUATION

We consider six domains of Metaworld Yu et al. (2020) and three domains of DM Control Tunyasuvunakool et al. (2020) for our empirical evaluation.

For all our experiments we use the internal state representation as the agent state. This includes reward learning, policy learning and world model training. The state features are as described in Yu et al. (2020). Further we follow Park et al. (2022) to utilize the packaged task rewards in MetaWorld for our synthetic oracles.

B REWARD ARCHITECTURE FOR PRIOR

We follow Park et al. (2022) reward architecture for Metaworld and DMControl domains.

C WORLD MODEL LEARNING

We generally follow the training paradigm suggested in TWM Robine et al. (2023) to learn our forward dynamics model. Furter, we follow the same architecture with minor changes to work with continuous control domains : :

1. TWM is proposed for discrete actions image based atari Brockman et al. (2016) domains, therefore we modify the observation encode layer to take in a 1-D state vector (instead of 3 channel image vector). Additionally, the original TWM makes use of frame-stacking which we do not.
2. We change the state predictor to take continuous valued vector based action space (as in our case) instead of discrete 1-D actions (as in Atari).

Finally, we modify the world model sequence length and memory length to be same as the query length such that we can feed the whole trajectory as an input to the TWM model.

To obtain an attention map from TWM we auto-repressively feed in a trajectory, i.e. first we feed in the first state,action tuple as trajectory of size 1, then the next state, action tuple and so on. At the final step we take the attention vector from the attention layers in the architecture.

C.1 OBSERVATION MODEL

The observation model in our world model encodes the input state,action tuple. The observation model has two components, an encode and a decoder both of which are multi-layer perceptrons.

The encoder has two hidden layers of size 512 and an output layer of size (32,). The decoder layer has an input size of (32,) followed by two hidden layers of size 512 and the output shape is same as the number of state features. As recommended by Robine et al. (2023) we use SiLU activation in the MLP.

C.2 LATENT STATE PREDICTOR

We follow Robine et al. (2023) to contruct the world model architecture with minor changes required to adapt to continuous control action space and 1-D state vector. That is, vanilla TWM requires categorical action embeddings (because of the discrete action space) but we do not. Finally, TWM Robine et al. (2023) can operate in different modes with respect to the prediction heads from the latent state predictor such as, next state prediction, reward prediction, and discount prediction. For Hindsight PRIOR we only require the next-state prediction output head.

Similarly TWM supports different input modes i.e. it can take (state,action) tuple and (state, action, reward) tuple. We compute state conditioned hindsight priors and therefore only need the (state, action) as input. Robine et al. (2023) is inconclusive on the need for rewards in the inputs. Moreover,

since PRIOR is learning the reward model to begin with we only use (state, action) as input. Robine et al. (2023) uses a Transformer XL architecture which we borrow as is.

C.3 Training World Model

We share the replay buffer of the agent with the world model. Similar to PbRL paradigm, the world model first gets access to a bank of state transitions during PbRL's pre-training step. Once the pretraining step is complete the world model is trained on this data (to get the observation model). After this pretraining step only the dynamics model is trained on incoming data every jth step of PbRL loop. The world model is trained on its bank of transitions as suggested by Robine et al. (2023).

D Policy Evaluation Metrics

Algorithms are evaluated according to their learning curves over the course of policy and reward training along with their normalized returns for Deep Mind Control Suite and normalized success rates for MetaWorld Lee et al. (2021b). An algorithm's normalized return or success rate measures how well the policies trained jointly with a preference-learned reward function recovers the performance of a policy trained on the target reward function. For each episode of policy training, the PbRL or SAC policy's return or success rate is computed and then the PbRL policy is evaluated based on how well it is able to recover "optimal" performance approximated with SAC trained on the ground truth reward. The normalized returns are computed as:

$$\text{normalized returns} = \frac{1}{T}\sum_{t}\frac{r_\psi(s_t, \pi_\phi^{\hat{r}_\psi}(a_t))}{r_\psi(s_t, \pi_\phi^{r_\psi}(a_t))}, \tag{6}$$

where T is the number of policy training steps, $\bar{r}_\psi$ is the target reward function, $\pi_\phi^{\hat{r}_\psi}$ is the policy trained in conjunction the learned reward function, and $\pi_\phi^{\bar{r}_\psi}$ is the policy trained on the target reward function. The normalized success rates are computed as:

$$\text{normalized success rates} = \frac{1}{T}\sum_{t}\frac{\text{success}(\pi_\phi^{\hat{r}_\psi}(a_t))}{\text{success}(\pi_\phi^{r_\psi}(a_t))}, \tag{7}$$

where $\text{success}(\cdot)$ indicates whether action a_t resulted in the policy reaching the goal state.

E Hyper-parameters

A results in the paper are reported over five random seeds: $[12345, 23456, 34567, 45678, 56789]$.

E.1 Train Hyper-parameters

This section specifies the hyper-parameters (e.g. learning rate, batch size, etc) used for the experiments and results. The SAC, and PEBBLE experiments all match those used in Haarnoja et al. (2018) and Lee et al. (2021a) respectively. The SAC hyper-parameters are specified in Table 2, the PEBBLE hyper-parameters are given in Table 3, the hyper-parameters used to train on with Hindsight PRIOR are in Table 4, and finally the hyperparameters used for training our world model in Table 5. Table 5 only mentions the hyper-parameters that we change in the prescribed Robine et al. (2023) configuration.

Table 2: Training hyper-parameters for SAC (Haarnoja et al., 2018).

HYPER-PARAMETER	VALUE
Learning rate	1e-3 (cheetah), 5e-4 (walker), 1e-4 (quadruped), 3e-4 (Meta-World)
Batch size	512 (DMC), 1024 (Meta-World)
Total timesteps	500k, 1M (quadruped, sweep into)
Optimizer	Adam (Kingma & Ba, 2015)
Critic EMA τ	5e-3
Critic target update freq.	2
$(\mathcal{B}_1, \mathcal{B}_2)$	$(0.9, 0.999)$
Initial Temperature	0.1
Discount γ	0.99

Table 3: PEBBLE hyper-parameters (Lee et al., 2021a).

HYPER-PARAMETER	VALUE
Learning rate PEBBLE	3e-4 (Metaworld), 5e-4 (Walker, Cheetah), 1e-4 (Quadruped)
Optimizer	Adam (Kingma & Ba, 2015)
Segment length l	50
Feedback amount / number queries (M)	1000/100, 100/10 (DMC) 10000/50, 4000/20, 2000/25, 400/10 (Meta-World)
Steps between queries (K)	20000 (walker, cheetah), 30000 (quadruped), 5000 (Meta-World)

Table 4: Hindsight PRIOR hyper-parameters.

HYPER-PARAMETER	VALUE
λ	1000 (MetaWorld), 5 (DMC)
update frequency j	2000 steps

Table 5: World Model hyper-parameters in Hindsight PRIOR

HYPER-PARAMETER	VALUE
world model sequence length	50
world model memory length	50
world model batch size	100

F NORMALIZED RETURNS AND SUCCESS RATES BY TASK

The mean normalized scores for each algorithm on each task are given for MetaWorld in Table 6 and DMC in Table 7.

A two-tailed paired t-test with dependent means (significant at p ¡ .05) was performed over the normalized returns and success rates to determine that Hindsight PRIOR's performance gains are statistically significant over:

1. **MetaWorld**: PEBBLE ($t = -3.92$, $p = 0.006$), SURF ($t = -2.85$, $p = 0.025$), RUNE ($t = -5.39$, $p = 0.001$), and MRN ($t = -4.91$, $p = 0.002$)
2. **DMC**: PEBBLE ($t = -3.47$, $p = 0.00843$), SURF ($t = -2.52$, $p = .03541$), RUNE ($t = -7.745967$, $p = 0.00006$), and MRN ($t = -2.392232$, $p = 0.0437$)

Table 6: Normalized Success Rate for MetaWorld domains

Env/Algorithm	**PRIOR**	**PEBBLE**	**SURF**	**RUNE**	**MRN**
Window Open	**0.55**	0.40	0.45	0.35	0.43
Door Open	**0.65**	0.62	0.55	0.53	0.53
Drawer Open	**0.68**	0.59	0.59	0.59	0.53
Sweep Into	**0.69**	0.51	0.54	0.52	0.62
Button Press	**0.69**	0.65	0.65	0.67	0.65
Door Unlock	**0.68**	0.62	0.64	0.53	0.54

Table 7: Normalized Returns for DM Control domains

Env/Algorithm	**PRIOR**	**PEBBLE**	**SURF**	**RUNE**	**MRN**
Walker Walk	**0.60**	0.46	0.66	0.47	0.59
Cheetah Run	**0.51**	0.33	0.36	0.39	0.46
Quadruped Walk	**0.65**	0.66	0.64	0.60	0.54

G ADAPTED BASELINES FOR PBRL

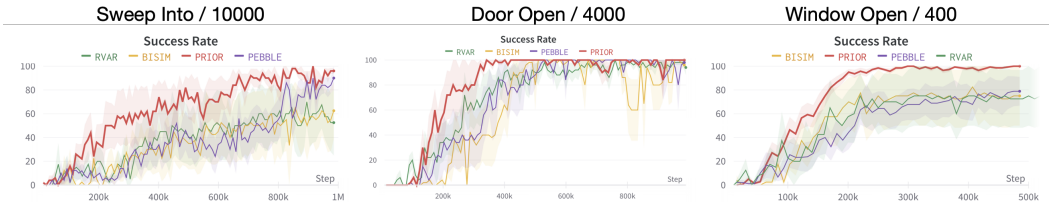

Figure 5: Learning curves of PRIOR, BISIM, RVAR and baseline PEBBLE

G.1 UNINFORMED RETURN REDISTRIBUTION (RVAR)

Inspired from Ren et al. (2021) we consider the Uninformed return redistribution baseline that essentially has reward targets as the mean reward in the trjaectory, i.e. $r_{target} = \frac{G}{|\tau|}$. This essentially reduces the variance of the trajectory rewards (hence the name RVAR). As discussed previously, RVAR is marginally better than PEBBLE and PRIOR is superior to such an uninformed redistribution technique. (See Fig. 5).

G.2 BISIMULATION METRIC : BISIM

To incorporate the bisimulation metrics into PbRL, we use the following bisimulation metric loss:

$$\mathcal{J}_{bisim}(\psi) = (||z_i - z_j||_1 - |r_i - r_j| - \gamma||z_i^{'(z_i,a_i)} - z_j^{'(z_j,a_j)})^2, \tag{8}$$

adapted from Equation 4 in Kemertas & Aumentado-Armstrong (2021). We use the reward model to provide the predicted rewards r_i, r_j required by bisim loss. Further we use the penultimate layer as the embedding layer to obtain z_i, z_j. Next, we add an additional head from the penultimate layer that predicts the next state embedding to obtain z_i', z_j' as next state predictors. Finally, we reuse the trajectory buffer (as used by Hindsight PRIOR) to obtain the states on which we compute the bisim-target distance and finally optimize the loss as above. We verify that the BISIM adapted baseline offers only marginal improvements over baseline PEBBLE as given in Fig. 5.

G.3 Normalized Redistribution PRIOR (NRP)

Readers may question whether the raw attention values extracted from the forward dynamics prediction task are the best choice or certain post processing may further improve PRIOR's performance. We construct a variant of PRIOR referred to as PRIOR-NRP where we perform a min-max normalization of the attention vector α. That is :

$$\hat{\alpha}_i = softmax(\frac{\alpha_i - \alpha_{min}}{\alpha_{max} - \alpha_{min}}) \tag{9}$$

The above equation amplifies the attention values thereby preventing the reward targets to become uniform. Fig. 6 shows that default PRIOR (without any postprocessing of attention) performs well and NRP like post-processing is not required.

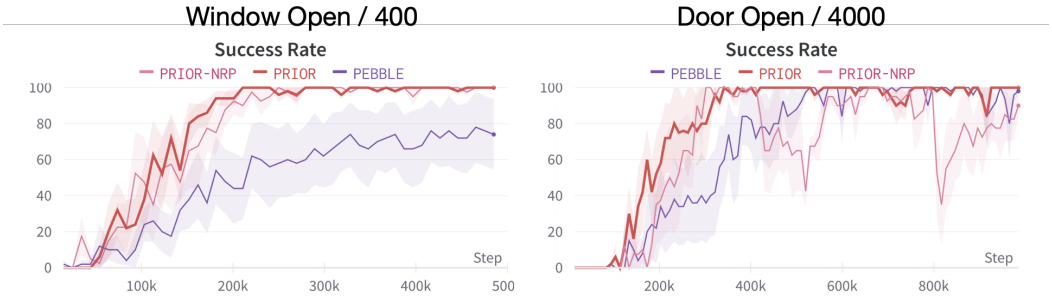

Figure 6: Learning curves of PRIOR, PRIOR-NRP and baseline PEBBLE

H Long Trajectories

Figure 7 shows the performance of PRIOR compared to PEBBLE when the trajectory length is 4x (200). While the Transformer XL architecture is already known for scaling to much longer trajectory lengths Robine et al. (2023) our experiments conclude that this is indeed the case for the challenging continuous control domains.

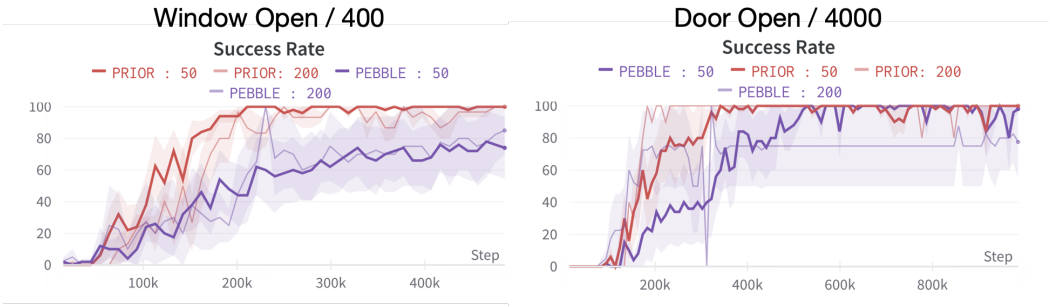

Figure 7: Learning curves of PRIOR and PEBBLE on query segment lengths of 50 and 200.

I QUALITATIVE STATE IMPORTANCE ASSESSMENT

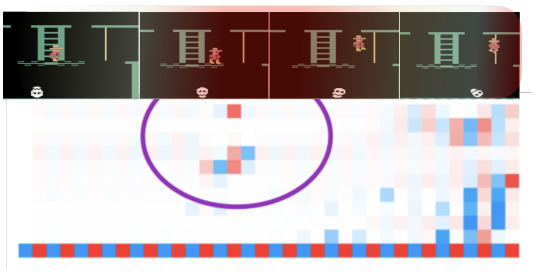

Figure 8: Attention analysis on Montezuma's Revenge. The plot shows a situation where the agent (red animated character) attempts to jump from the green platform towards the rope. Each row represents a layer in the Transformer Model and the columns represent the time step where left most column is time t-T and right most column is the present state of the agent. In hindsight, we compute this attention map to obtain the attention over the past states (given as blue cells) and attention over past actions (given as red cells). Note that the map begins with a blue cell (as $s_0, a_0, a_1, a_1 \cdots$). We highlight that the agent attends to past states which corresponds to point of launch which can be considered as an important summary state for the complete trajectory of jumping from the platform to the rope.

We qualitatively evaluate the states identified by the attention weights $\boldsymbol{\alpha}$ as important for a given trajectory. We evaluate whether the attention map $\mathcal{A}$ salience over a trajectory can capture critical events. We conduct experiments on Atari (Montezuma's Revenge) and Metaworld (Window-Open, Door Open) domains, and seek to answer whether the world model attends to states in the complete history (even for Markovian transitions) and whether that correlates with "critical" events (similar to how Kim et al. (2023) describes it). From figure 8 we can see that the forward model does attend to past states and actions. Moreover, upon closer inspection by executing known maneuvers, such as jumping across the platform to the rope in Montezuma's Revenge, we find that the agent is attending to past critical events like the point of launch. We do not expect attention over states in history to be aligned with human's reward function as human may have arbitrary preference unknown to the dynamics model. However, we do expect the attention to be aligned with "critical events" loosely defined as apriori states enough to summarize a trajectory. To further investigate this we force the reward model to predict rewards aligned with the PRIOR reward targets by very high value λ_{prior} in equation 5 and find that the reward model is unable to learn preference-relevant reward model at all. This shows that PRIOR reward targets are not task specific but contain salience information on states which can be considered "critical" in the sampled trajectory.