CUDA: Capturing Uncertainty and Diversity in Preference Feedback Augmentation

Sehyeok Kang [*1] **Jaewook Jeong** [*2] **Se-Young Yun** [1]

Abstract

Preference-based Reinforcement Learning (PbRL) effectively addresses reward design challenges in RL and facilitates human-AI alignment by enabling agents to learn human intentions. However, optimizing PbRL critically depends on abundant, diverse, and accurate human feedback, which is costly and time-consuming to acquire. While existing feedback augmentation methods aim to leverage sparse human preferences, they often neglect diversity, primarily generating feedback for trajectory pairs with extreme differences based on high confidence. This limitation restricts the diversity of augmented dataset, leading to an incomplete representation of human preferences. To overcome this, we introduce Capturing Uncertainty and Diversity in preference feedback Augmentation (CUDA), a novel approach that comprehensively considers both uncertainty and diversity. CUDA enhances augmentation by employing ensemble-based uncertainty estimation for filtering and extracting feedback from diverse clusters via bucket-based categorization. These two mechanisms enable CUDA to obtain diverse and accurate augmented feedback. We evaluate CUDA on MetaWorld and DMControl offline datasets, demonstrating significant performance improvements over various offline PbRL algorithms and existing augmentation methods across diverse scenarios.

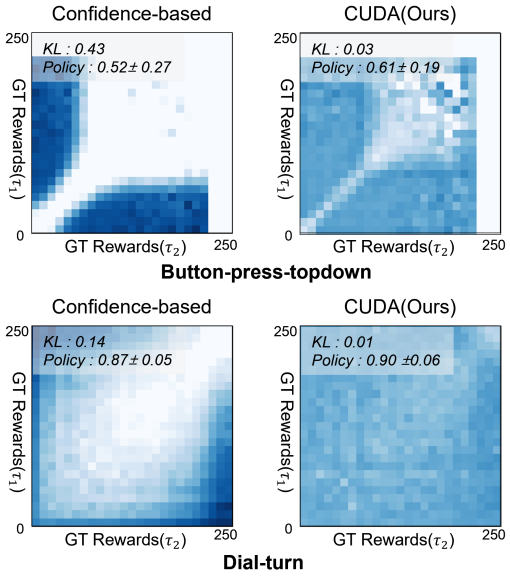


Figure 1. **Comparison of conventional augmentation methods (confidence-based) and CUDA.** The heatmap displays the distribution of ground truth rewards for the augmented pairs each model generates. The intensity of the heatmap indicates concentration. Confidence-based augmentation concentrates augmented feedback where two trajectories have large differences. In contrast, CUDA's augmented pairs show a dispersed distribution. *KL* denotes the KL divergence between estimated and ground truth rewards, which represents misalignment with the feedback; *Policy* indicates the success rate of the policy trained with augmented feedback. CUDA, notably, both aligns more closely with the ground truth reward and makes a more robust policy.

1. Introduction

Preference-based Reinforcement Learning (PbRL) offers a compelling solution to the challenges of reward design in RL and shines as a key human-AI alignment method, fundamentally enabling agents to learn and align their actions with human intentions. However, while ensuring PbRL performance requires a large quantity of diverse and accurate feedback for training, obtaining extensive human feedback faces considerable costs and time limitations. To address this, feedback augmentation methods (Park et al., 2022; Hwang et al., 2023; Choi et al., 2024) have emerged, allowing the system to leverage sparse human preferences.

Crucially, however, current augmentation strategies predominantly overlook the diversity aspect of the generated feedback, focusing almost exclusively on its accuracy. For

[*]Equal contribution [1]KAIST AI, Republic of Korea [2]Work done while an intern at KAIST AI. Correspondence to: Se-Young Yun <yunseyoung@kaist.ac.kr>.

ICML 2025 Workshop on Models of Human Feedback for AI Alignment, Vancouver, Canada.

instance, the strategies often opt to select only pairs with extremely confident estimated preference probabilities. This means augmented feedback primarily targets trajectory pairs already exhibiting pronounced differences. Such a selection criterion inherently restricts the diversity of the augmented dataset, creating a concentrated pool of feedback that fails to cover the full spectrum of human preferences and nuanced decision-making, and consequently, introduces bias into the augmented feedback. Figure 1 illustrates the point, with the left (confidence-based) showing the results of feedback augmentation using conventional strategies that filter using the difference between trajectories. In these scenarios, augmentation predominantly occurs for trajectory pairs where the reward difference is extreme, leading to a lack of diversity in the generated feedback.

To overcome these challenges, we introduce **C**apturing **U**ncertainty and **D**iversity in preference feedback **A**ugmentation (CUDA), a novel approach that comprehensively considers both uncertainty (confidence) and diversity in augmenting preference feedback. CUDA enhances feedback augmentation by utilizing unlabeled trajectories to ensure diversity through two key mechanisms.

- **Bucket-based sampling**: CUDA utilizes a strategy that distinguishes and places unlabeled trajectories into multiple buckets based on estimated rewards, and then samples across the buckets to ensure diverse pairs.
- **Uncertainty-based filtering**: Instead of relying on conventional confidence-based filters, CUDA applies ensemble-based uncertainty filtering by selecting only cases where confidence intervals do not overlap.

These two mechanisms enable CUDA to obtain diverse and accurate augmented feedback. The right side of Figure 1 illustrates the distribution of augmented feedback when using CUDA, showing a significantly more diverse range. Furthermore, it better aligns with the feedback and consequently contributes to generating a superior policy.

We evaluate the performance of CUDA on MetaWorld and DMControl offline datasets provided by Choi et al. (2024). Experimental results indicates that CUDA significantly outperforms a variety of offline PbRL algorithms. Furthermore, it demonstrates remarkable performance gains over augmentation-based methods in a wide range of scenarios.

2. Background

2.1. Offline Preference-Based Reinforcement Learning

Preference-Based Reinforcement Learning (PbRL) (Christiano et al., 2017) is a framework for training reinforcement learning agents without explicitly defining a reward function. The goal of offline PbRL is to optimize the policy function $\pi_\theta(a|s)$ using this pre-collected preference data, without any further interaction with the environment. Instead of directly designing a reward signal, a supervisor provides feedback in the form of preferences. Generally, a preference is composed of two trajectories (τ_1, τ_2) and a human feedback label (y) that encodes a comparison between them.

To model these preferences, the Bradley-Terry(Bradley & Terry, 1952) is commonly used in pairwise comparison problems. The probability that trajectory τ_1 is preferred over trajectory τ_2 is given by:

$$P[\tau_1 \succ \tau_2] = \frac{\varphi(f_\theta(\tau_1))}{\varphi(f_\theta(\tau_1)) + \varphi(f_\theta(\tau_2))} \tag{1}$$

Here, $f_\theta(\tau)$ represents the score assigned to trajectory τ by the model, which reflects its estimated quality or preference and φ is an activation function, commonly using either the exponential function or a linear function. The model parameters θ are learned by minimizing the cross-entropy loss between the predicted preference probabilities and the actual human feedback labels. The loss for a single pairwise comparison is defined as:

$$\mathcal{L} = -(y \log P[\tau_1 \succ \tau_2] + (1-y) \log P[\tau_2 \succ \tau_1]) \tag{2}$$

where $y = 1$ if τ_1 is preferred over τ_2, and $y = 0$ otherwise.

By optimizing this loss function over a set of labeled trajectory pairs, the model aligns its predictions with human preferences, effectively inferring a reward signal without the need for manual design.

There have been several approaches to optimizing policies in offline PbRL. Preference Transformer (PT) (Kim et al., 2023) models human preferences using a Transformer-based architecture and proposes a method to learn non-Markovian rewards as a weighted sum, extending existing Markovian reward assumptions. This approach effectively reflects temporal dependencies and enables the solution of complex control tasks. Offline Preference-based Reward Learning (OPRL) (Shin et al., 2023) approached offline PbRL by selecting queries with high Value of Information through active learning. Direct Preference-based Policy Optimization without reward modeling (DPPO) (An et al., 2023) learns policies directly from preference by utilizing a contrastive learning framework. Inverse Preference Learning (IPL) (Hejna & Sadigh, 2024) proposes an efficient algorithm that directly learns preference data using Q-functions without explicitly learning a reward function. This design allows for effective learning with a simpler structure and fewer parameters.

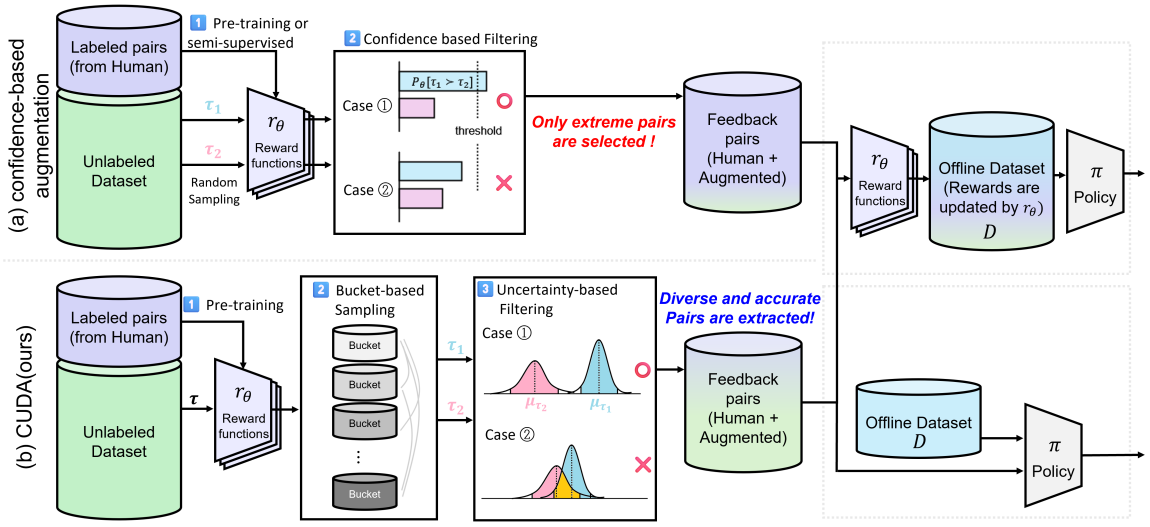


Figure 2. Comparison of CUDA and existing methods. (a) Conventional confidence-based feedback augmentation. (b) The CUDA framework. CUDA ensures diversity through bucket-based sampling, which groups estimated rewards and induces sampling from different buckets. Furthermore, uncertainty-based sampling enables CUDA to select pairs even without extreme differences, achieving diverse yet confident augmentation. CUDA's augmented feedback supports both reward-based and reward-free policy learning.

2.2. Feedback Augmentation in PbRL

Feedback augmentation is a key technique for improving the performance of PbRL by maximizing the information gained from limited human feedback.

SURF (Park et al., 2022) first introduces feedback augmentation to PbRL, particularly within an online learning framework. Its methodology involves augmenting preference pairs by filtering for those where the estimated reward based preference probability exceeds a specific threshold, subsequently leveraging these augmented feedback for semi-supervised learning. The reliance of SURF on only confidence-based filtering limits its augmented data to trajectory pairs exhibiting significant disparities. Our approach, however, can include pairs with even subtle differences between trajectories, leading to the creation of a richer and more diverse set of augmented feedback.

SeqRank (Hwang et al., 2023) utilizes sequential preference ranking to efficiently generate additional feedback through comparisons between defender and challenger. LiRE (Choi et al., 2024) proposed a method to improve the performance of the reward model without additional data collection by constructing a Ranked List of Trajectories (RLT) using second-order preferences. SeqRank and LiRE use trajectories for augmentation only if those trajectories have been included in a feedback ranking at least once. On the other hand, our approach can include unlabeled trajectories in the feedback generation process. This allows the model to infer preferences for previously unseen trajectories, which can greatly expand the scope of data augmentation. In addition, unlike existing methods, which require sequential and online feedback collection, our method can learn from arbitrary and simultaneous feedback without being bound by strict order.

3. Problem Definition

3.1. Offline PbRL

In PbRL, an agent learns a policy by utilizing human feedback rather than explicit reward functions. For the offline setting, we assume the following conditions. Given an offline dataset $\mathcal{D}$, it comprises a large quantity of unlabeled trajectories $\mathcal{D}_u$ and a limited number of trajectory pairs $\mathcal{F}_l$ labeled by human feedback. The goal is to learn an optimal policy $\pi_\theta(a|s)$ that maximizes the inferred reward signal derived from these preference labels.

3.2. Insufficient diversity in feedback augmentation

Optimizing the reward function with only a limited number of labeled pairs ($\mathcal{F}_l$) poses challenges. Consequently, some researchers augment preference feedback based on estimated rewards. They implement confidence-based approaches that calculate probabilities using estimated rewards from Eq. (1) and add a pair as augmented feedback if the value exceeds a certain threshold.

However, this method does not guarantee diverse augmented feedback. Typically, only pairs with extremely large trajectory differences exceed the threshold, meaning the augmented data does not include common cases, leading to

Table 1. Performance difference between confidence-based feedback augmentation and real feedback. **Confidence** refers to the case where we train the model using 10,000 augmented pairs from confidence-based augmentation. In contrast, **Oracle** represents obtaining 10,000 feedback samples using a scripted teacher model. Oracle and Confidence show a significant performance difference.

Method	button-press	box-close	sweep	Average
Confidence	0.55 ± 0.18	0.63 ± 0.27	0.76 ± 0.10	0.65
Oracle	0.73 ± 0.32	0.89 ± 0.13	0.86 ± 0.15	0.83
Difference	0.18	0.26	0.10	0.18

inherent bias in the augmented feedback. As a result, it shows a significant performance difference compared to using real feedback. Although confidence-based augmentation achieves high agreement with oracle labels, it leads to significantly lower policy performance compared to using 10,000 oracle feedbacks (Table 1).

Therefore, we require a novel feedback augmentation approach that resolves the diversity limitations and provides both diverse and confident feedback.

4. Method

4.1. Our Method: CUDA

We propose **CUDA**, a unified framework that combines reward model learning with feedback augmentation. While our overall goal is to capture both diversity and reliability in the augmented feedback, we achieve this through three main components:

(1) We train a bootstrapped ensemble of reward models on a limited set of labeled feedback to enable robust return prediction and uncertainty estimation.

(2) We partition the unlabeled trajectories into multiple buckets based on mean predictions, and sample feedback from inter-bucket pairs to ensure diversity.

(3) To maintain reliability, we filter out trajectory pairs with high predictive uncertainty.

This integrated pipeline allows us to selectively generate preference labels that are both diverse and reliable, enhancing reward model performance with minimal human effort.

4.2. Reward Models from Labeled Feedback

We use the Markovian Reward (MR) model as our reward model. During training, we apply an exponential transformation $\varphi(x) = e^x$, which is used to define the probability in the loss function.

We train an ensemble of N reward models with different random initializations on the labeled data. For each state-action pair (s_t, a_t) in a trajectory $\tau = \{(s_1, a_1), \ldots, (s_T, a_T)\}$, each model predicts a reward $r_n(s_t, a_t)$, and the ensemble mean is given by:

$$\bar{r}_t = \frac{1}{N} \sum_{n=1}^{N} r_n(s_t, a_t) \tag{3}$$

The predicted return for the trajectory is computed as:

$$\hat{R}(\tau) = \sum_{t=1}^{T} \bar{r}_t \tag{4}$$

To estimate uncertainty, we compute an uncertainty score $u(\tau)$ by summing the variance of the reward predictions at each timestep:

$$u(\tau) = \left(\sum_{t=1}^{T} \left(\frac{1}{N} \sum_{n=1}^{N} \left(r_n(s_t, a_t) - \bar{r}_t \right)^2 \right) \right)^{1/2} \tag{5}$$

This state-wise variance captures local prediction disagreement across the ensemble, and its sum provides a trajectory-level uncertainty estimate used for data filtering.

4.3. Bucket-based Sampling

To promote diversity in the augmented feedback, we cluster the set of unlabeled trajectories into k buckets using k-means++(Arthur & Vassilvitskii, 2006) clustering based on their predicted return $\hat{R}(\tau)$. Instead of fixing k in advance, we search for the optimal number of clusters within the range $[\frac{1}{2}k, k]$, and select the value of k that maximizes the silhouette score(Rousseeuw, 1987). The method ensures more natural groupings of trajectories with similar rewards.

To construct trajectory pairs, we consider all unordered bucket pairs (i, j) where $i < j$. For each such pair, we sample a number of feedback pairs proportional to the product of the sizes of the two buckets, i.e., $|\mathcal{B}_i| \cdot |\mathcal{B}_j|$, where $\mathcal{B}_i$ denotes the set of trajectories in bucket i. The total number of generated preference pairs is normalized so that the overall number of pairs sums to approximately n.

The strategy emphasizes high-volume bucket pairs while preserving return diversity, resulting in a balanced and representative feedback set for reward model training.

4.4. Uncertainty-based Filtering

To ensure the confindence of augmented feedback, we apply uncertainty-based filtering to the trajectory pairs selected from bucket-based sampling. For each trajectory τ, we use the uncertainty score $u(\tau)$ defined in Eq. (5).

When constructing a pair between two trajectories τ_i and τ_j from bucket $i < j$, we include the pair only if their predicted

Table 2. Average success rates on Metaworld.

	Algorithms	button-press-topdown	box-close	dial-turn	sweep	button-press-topdown-wall	sweep-into	drawer-open	lever-pull	avg
	Ground Truth*	0.69 ± 0.11	0.75 ± 0.02	0.51 ± 0.04	0.68 ± 0.06	0.30 ± 0.05	0.43 ± 0.05	0.20 ± 0.16	0.25 ± 0.19	0.48
Offline PbRL	MR-linear*	0.33 ± 0.23	0.44 ± 0.35	0.40 ± 0.14	0.95 ± 0.04	0.13 ± 0.09	0.25 ± 0.08	0.16 ± 0.16	0.51 ± 0.06	0.40
	MR*	0.03 ± 0.03	0.35 ± 0.37	0.30 ± 0.29	0.92 ± 0.11	0.05 ± 0.05	0.31 ± 0.11	0.15 ± 0.05	0.79 ± 0.24	0.33
	PT*	0.07 ± 0.11	0.02 ± 0.01	0.00 ± 0.00	0.15 ± 0.15	0.00 ± 0.00	0.16 ± 0.04	0.11 ± 0.04	0.07 ± 0.08	0.06
	OPRL†	0.12 ± 0.06	0.04 ± 0.03	0.54 ± 0.11	0.94 ± 0.06	0.00 ± 0.00	0.26 ± 0.08	0.94 ± 0.06	0.54 ± 0.12	0.42
	DPPO†	0.04 ± 0.04	0.10 ± 0.11	0.27 ± 0.22	0.10 ± 0.16	0.01 ± 0.01	0.23 ± 0.07	0.36 ± 0.11	0.10 ± 0.12	0.15
	IPL†	0.34 ± 0.14	0.06 ± 0.05	0.32 ± 0.12	0.27 ± 0.24	0.09 ± 0.09	0.32 ± 0.07	0.19 ± 0.13	0.31 ± 0.15	0.23
Augmentation	SURF (offline)*	0.37 ± 0.26	0.33 ± 0.36	0.38 ± 0.17	0.85 ± 0.28	0.13 ± 0.07	0.19 ± 0.07	0.09 ± 0.08	0.53 ± 0.15	0.36
	SeqRank*	0.34 ± 0.26	0.01 ± 0.02	0.26 ± 0.12	0.35 ± 0.33	0.02 ± 0.04	0.25 ± 0.08	0.27 ± 0.13	0.17 ± 0.25	0.20
	LiRE*	0.57 ± 0.21	0.79 ± 0.18	0.82 ± 0.16	0.68 ± 0.33	0.28 ± 0.07	0.33 ± 0.09	0.01 ± 0.06	0.82 ± 0.28	0.54
	CUDA (Ours)	0.59 ± 0.27	0.64 ± 0.23	0.90 ± 0.06	0.88 ± 0.07	0.29 ± 0.11	0.36 ± 0.09	0.04 ± 0.06	0.84 ± 0.05	0.56

$*$ shows our implementation results, and † presents results from Choi et al. (2024)

Table 3. Performance on DMControl

Algorithm	hopper-hop	walker-walk	humanoid-walk	avg
Ground Truth†	157.95 ± 9.64	839.6 ± 36.57	250.9 ± 11.62	416.15
MR-linear*	127.69 ± 26.22	635.02 ± 94.35	89.92 ± 12.78	284.21
SURF*	127.60 ± 29.00	696.56 ± 78.61	100.95 ± 15.21	308.37
SeqRank†	80.84 ± 27.67	698.81 ± 91.71	80.68 ± 14.67	286.77
LiRE†	99.14 ± 12.28	822.27 ± 50.83	104.08 ± 17.45	341.83
CUDA (Ours)	132.50 ± 19.68	784.64 ± 34.21	148.06 ± 35.94	355.07

$*$: our implementation; †: results from Choi et al. (2024)

returns $\hat{R}(\tau_i)$ and $\hat{R}(\tau_j)$ are separated by a sufficient margin relative to their uncertainties. Specifically, the pair is chosen if it satisfies the following condition:

$$\hat{R}(\tau_i) + z \cdot u(\tau_i) < \hat{R}(\tau_j) - z \cdot u(\tau_j) \tag{6}$$

,where z is a confidence parameter controlling the tolerance for uncertainty. This criterion ensures that the relative preference between τ_i and τ_j can be inferred with high confidence, effectively filtering out low-confidence comparisons.

We combine the original labeled feedback with the augmented preference pairs generated from unlabeled trajectories to construct the final training set of preference feedback.

4.5. Reward Learning and Policy Optimizing

We use the augmented feedbacks to train a reward model. The model follows the Markovian Reward (MR) architecture and applies a final activation layer of $\tanh(x) + 1$ to ensure the output is strictly positive.

To compute preference probabilities during training, we apply a linear function, $\varphi(x) = x$, which is applied to the cumulative rewards of each trajectory. This formulation particularly excels at capturing second-order preference that can emerge from augmented feedback (Choi et al., 2024).

Once trained, the reward model is used to generate pseudo-rewards for the unlabeled offline dataset. These pseudo-rewards are then used to train a policy using Implicit Q-Learning (IQL) (Kostrikov et al., 2021) , allowing the agent to optimize behavior aligned with the learned preferences.

5. Experimental Results

5.1. Settings

5.1.1. Offline Dataset

We evaluate CUDA on two widely used benchmark datasets in offline reinforcement learning: Meta-World (Yu et al., 2020) and DeepMind Control Suite (DMControl) (Tassa et al., 2018). These environments provide diverse tasks for evaluating both locomotion and robotic manipulation. We use the dataset collected by (Choi et al., 2024) from Meta-World and DMControl. For our experiments, we utilize feedback generated by a scripted teacher based on this dataset. The scripted teacher, a widely used method in PbRL (Choi et al., 2024; Hwang et al., 2023; Kim et al., 2023), replaces human feedback with rewards from the environment to pose preferences. These environmental rewards are solely used for preference calculation and not for model training.

5.1.2. Baselines

To evaluate the effectiveness of our proposed method, we compare it against multiple baselines, including both offline PbRL methods and feedback augmentation-based PbRL methods. We implement a Markovian Reward (MR) baseline, where the reward model is trained under the assumption that the reward depends only on the current state-action

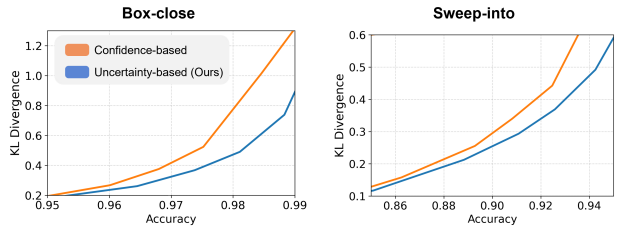


Figure 3. KL divergence of selected feedback distributions compared to random sampling, measured at matched accuracy levels by adjusting the threshold (confidence-based) or z-value (uncertainty-based). Lower KL indicates greater diversity.

Table 4. KL divergence under different strategies. **CS** = Confidence-based Sampling; **UF** = Uncertainty-based Filtering; **BS** = Bucket-based Sampling.

Method	button-press -topdown	sweep-into	lever-pull	box-close	avg
CS	0.4291	0.2686	0.3122	0.3166	0.3316
BS+CS	**0.0302**	**0.0115**	**0.0466**	**0.0126**	**0.0252**
UF	0.0617	0.0261	0.0486	0.0287	0.0413
UF+BS	**0.0305**	**0.0109**	**0.0477**	**0.0124**	**0.0254**

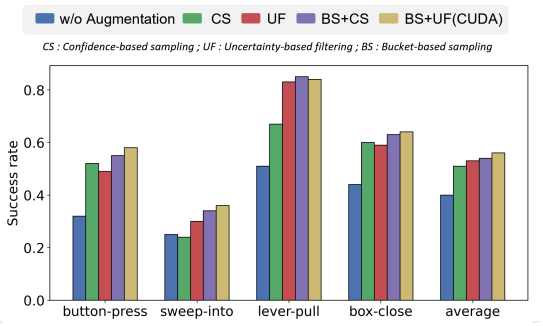


Figure 4. Performance comparison with and without application of each strategy. *CS* denotes the conventional augmentation method.

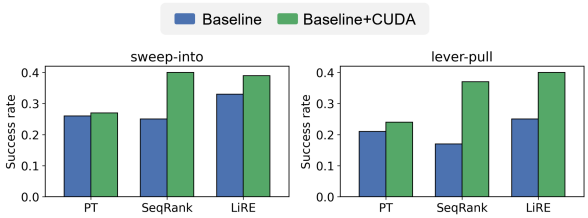


Figure 5. Experimental results confirming CUDA's compatibility.

pair (s, a), following the standard Markov decision process (MDP) formulation. We also evaluate an MR-linear baseline that incorporates a linear activation function ($\varphi(x) = x$). Moreover, we compare CUDA against Preference Transformer (PT) (Kim et al., 2023), OPRL (Shin et al., 2023), DPPO (An et al., 2023), and IPL (Hejna & Sadigh, 2024) to evaluate its performance relative to other PbRL methods. Furthermore, we compare CUDA with feedback augmentation methods such as SURF (Park et al., 2022), SeqRank (Hwang et al., 2023), and LiRE (Choi et al., 2024). We re-implement the online version of SURF for offline use.

5.1.3. Implementation Details

Both CUDA and all baselines utilize a 500 feedback pair dataset per scenario for model training. Also, all reward models—including those used in baselines and CUDA—are trained using the MR-linear architecture. For policy learning, we employ Implicit Q-Learning (IQL) (Kostrikov et al., 2021) as the policy optimization algorithm. Each experiment is run with ten different random seeds, and the final results are reported as the mean and standard deviation across these runs. Further details on hyperparameters settings are provided in the Appendix A.2.

5.2. Main Results

To evaluate the performance of CUDA, we conduct a comparative study against existing offline PbRL approaches and PbRL methods that utilize feedback augmentation. Table 2 presents the average success rates of various methods across different tasks in the MetaWorld environment. The results indicate significant performance variations among the methods, highlighting the effectiveness of different algorithms. CUDA demonstrates superior performance across multiple tasks, achieving the highest success rates in five tasks. Additionally, the average value for CUDA across all scenarios surpasses other baselines. Notably, CUDA outperforms offline PbRL methods. Moreover, it also demonstrates superiority when compared to augmentation methodologies.

Additionally, our method demonstrates better performance on the DMControl environment, as shown in Table 3, which presents the average rewards of different methods on DMControl. CUDA consistently shows improved performance against baselines. These results collectively prove that CUDA is capable of generating better augmented feedback in diverse scenarios and across different environments.

5.3. The efficacy of bucket-based sampling and uncertainty-based filtering

Uncertainty-based filtering Figure 3 compares confidence-based and uncertainty-based filtering by plotting KL divergence (relative to random sampling) against accuracy (agreement with ground-truth preference labels). Across all tasks, uncertainty-based filtering consistently achieves lower KL divergence than confidence-based sampling at the same accuracy level. This indicates that uncertainty-based methods can yield more diverse augmented feedback while maintaining comparable label correctness.

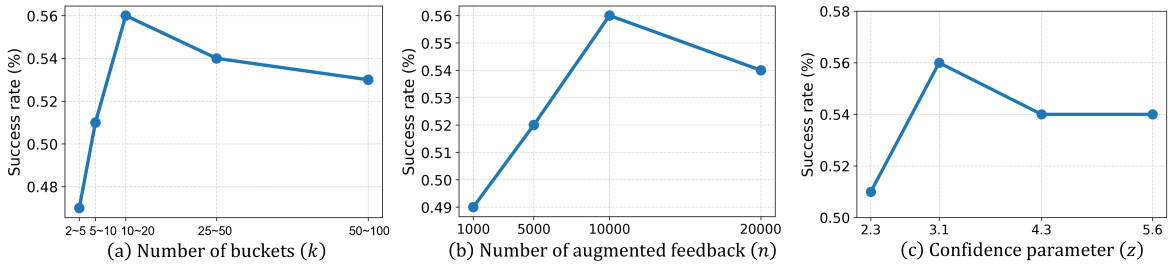


Figure 6. Ablation study findings. The success rate in the graph represents the average value across all tasks in Metaworld. (a) Success rate according to changes in the number of buckets (k). (b) Success rate based on the number of augmented feedback (n). (c) Success rate variation with respects to confidence parameter z

Table 5. Experimental results using human feedback

Scenario	MR-linear	LiRE	CUDA(Ours)
`Button-press-topdown`	0.06 ± 0.13	0.12 ± 0.07	0.36 ± 0.16

Bucket-based sampling Table 4 shows the effect of applying bucket-based sampling. Bucket-based sampling consistently reduces KL divergence across environments, whether used with confidence-based or uncertainty-based filtering. This suggests that bucket-based sampling enhances feedback diversity by preventing bias toward extreme pairs. The full results across all 8 environments are available in Table 7.

Overall Policy Performance Comparison Figure 4 summarizes the overall policy performance under different combinations of strategies. **CUDA** (BS + UF) achieves the highest performance across environments, while using only confidence-based sampling (CS) results in the worst performance among the augmented variants.

5.4. Comparability

To evaluate CUDA's compatibility by assessing whether the augmented feedback it generates can enhance the performance of various existing offline PbRL methods, we apply CUDA to PT, SeqRank and LiRE. Figure 5 presents the results of the application. Compared to the baselines, we observe a general increase in performance (indicated by green) when we include augmented pairs generated by CUDA in the training. Even for models like SeqRank and LiRE that already incorporate augmented data, CUDA's augmented data significantly contributes to improving policy performance. Therefore, CUDA is compatible with both preference-based RL methods and augmentation-based approaches.

5.5. Alignment with human feedback

To confirm CUDA's alignment with human preferences, we perform experiments using human feedback. Table 5 shows the performance in the `button-press-topdown` task in the Metaworld. We use 200 human preference feedback samples provided by (Choi et al., 2024). The results demonstrate CUDA achieves significant performance improvements compared to both MR-linear and LiRE.

5.6. Ablation Studies

Number of buckets(k): CUDA leverages bucket-based sampling to achieve feedback diversity. Figure 6 (a) depicts policy performance changes relative to the number of buckets. We determine the optimal bucket count using the silhouette score, selecting a value within minimum and maximum bounds. Experiments show the best performance when k ranges from 10 to 20.

Number of augmented feedback(n): Figure 6 (b) shows the change in policy performance based on the number of augmented feedback. Performance generally improves with more feedback, reaching its best at 10,000, after which it declines. We attribute the performance drop at 20,000 to a decrease in the accuracy of the augmented feedback, highlighting the importance of augmenting an appropriate number of samples and the accuracy.

Confidence parameter(z): We use z as the confidence boundary for uncertainty-based filtering in Equation (6). Since confidence changes with z, Figure 6 (c) compares performance across various z values to assess its impact. The results show that z yields the best performance at 3.1.

6. Conclusion and Limitations

In this work, we propose Capturing Uncertainty and Diversity in Preference Feedback Augmentation (CUDA), a method that addresses the challenge of limited human feedback in PbRL. CUDA introduces a novel approach that comprehensively integrates both uncertainty (confidence) and diversity into the augmentation process. It achieves this through two key mechanisms: utilizing ensemble-based uncertainty estimation for robust filtering, and promoting a broader range of augmented preferences by extracting feedback from diverse, bucket-categorized trajectory clus-

ters. These innovations allow CUDA to generate highly diverse and accurate augmented feedback. Our extensive evaluations on MetaWorld and DMControl offline datasets clearly demonstrate CUDA's superior performance, outperforming both conventional offline PbRL algorithms and existing augmentation-based methods across a wide array of scenarios. This work highlights the critical importance of diversity alongside accuracy in feedback augmentation for advancing PbRL capabilities.

Since CUDA augments feedback based on a model trained with original feedback, it becomes crucial to perform effective initial sampling and obtain accurate true feedback. Inadequate feedback hinders its ability to perform well. Furthermore, maintaining the preference relationships between feedback pairs is essential for effective augmentation. For example, cyclic feedback has limitations in generating buckets, thus impairing performance. Therefore, as our future work, we are considering methods to obtain high-quality feedback during the initial feedback acquisition process, and methods to robustly train the sore function even when feedback with cycles is given.

Acknowledgements

This work was supported by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD) (UD230017TD).

References

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K.-M., and Song, H. O. Direct preference-based policy optimization without reward modeling. *Advances in Neural Information Processing Systems*, 36:70247–70266, 2023.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

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

Choi, H., Jung, S., Ahn, H., and Moon, T. Listwise reward estimation for offline preference-based reinforcement learning. *arXiv preprint arXiv:2408.04190*, 2024.

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

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Hejna, J. and Sadigh, D. Inverse preference learning: Preference-based rl without a reward function. *Advances in Neural Information Processing Systems*, 36, 2024.

Hwang, M., Lee, G., Kee, H., Kim, C. W., Lee, K., and Oh, S. Sequential preference ranking for efficient reinforcement learning from human feedback. *Advances in Neural Information Processing Systems*, 36:49088–49099, 2023.

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

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Lee, K., Smith, L., and Abbeel, P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021.

Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., and Lee, K. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. *arXiv preprint arXiv:2203.10050*, 2022.

Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Shin, D., Dragan, A. D., and Brown, D. S. Benchmarks and algorithms for offline preference-based reward learning. *arXiv preprint arXiv:2301.01392*, 2023.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

A. Experimental Details

A.1. Dataset

In this study, we utilized the dataset originally collected in the LiRE(Choi et al., 2024). The dataset was generated by collecting replay buffers during the training of online reinforcement learning agents. Specifically, the online Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018), implemented in PEBBLE(Lee et al., 2021), using ground-truth rewards.

Replay buffers are collected by evaluating the online policy's success rate every 50,000 steps. We use data gathered until the policy's success rate approaches 50%, same as the one used in the source paper. This approach ensures that the performance is compared fairly under the same conditions as described in the source paper.

A.2. Hyperparameters and Setup

Table 6. Hyperparameters of the reward model and the baselines.

Hyperparameter	**Value**
Reward Model	
Optimizer	Adam
Learning rate	1e-3
Batch size	32
Hidden layer dim	256
Hidden layers	2
Activation function	ReLU
Final activation	Tanh
Epochs	200
# of ensembles (labeled / augmented)	7 / 3
Ensemble aggregation Method	Average
IQL (Kostrikov et al., 2021)	
Optimizer	Adam
Critic, Actor, Value hidden dim	256
Critic, Actor, Value hidden layers	2
Critic, Actor, Value activation function	ReLU
learning rate	3e-4
Batch size	256
Discount factor γ	0.99
Soft Update Rate τ	0.05
Temperature β	3.0
Expectile	0.7

Setup We use a single NVIDIA RTX A5000 GPU and 96 CPU cores (Intel Xeon Gold 5220R @ 2.20GHz) in our experiments. The system runs on Ubuntu 22.04 with Linux kernel 6.5.0 and CUDA 11.8.

A.3. CUDA Algorithm Pseudocode

Algorithm 1 CUDA Algorithm

Require: Unlabeled dataset $\mathcal{D}_u$, Labeled feedback $\mathcal{F}_\ell$, Number of buckets K
Ensure: Feedback set to train policy $\mathcal{F}_{\text{refined}}$

1. Train an ensemble of reward models $\mathcal{R}_\theta = \{r_1, \ldots, r_N\}$ on labeled feedback $\mathcal{F}_\ell$

2. For each trajectory $\tau = \{(s_t, a_t)\}_{t=1}^{T}$ in $\mathcal{D}_u$, compute the ensemble-averaged reward at each timestep:

$$\bar{r}_t = \frac{1}{N} \sum_{n=1}^{N} r_n(s_t, a_t)$$

and the predicted return:

$$\hat{R}(\tau) = \sum_{t=1}^{T} \bar{r}_t$$

3. For each trajectory τ, compute an uncertainty score:

$$u(\tau) = \left(\sum_{t=1}^{T} \left(\frac{1}{N} \sum_{n=1}^{N} (r_n(s_t, a_t) - \bar{r}_t)^2 \right) \right)^{1/2}$$

4. Cluster the trajectories in $\mathcal{D}_u$ into K buckets $\{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$ using k-means++ on the predicted return $\hat{R}(\tau)$. The optimal number of clusters k^* is selected as:

$$k^* = \arg \max_{k \in [\frac{1}{2}K, K]} \texttt{silhouette\_score}\left(k, \mathcal{D}_u; \mathcal{R}_\theta\right)$$

5. Assign sampling weights $w_{ij} \propto |\mathcal{B}_i| \cdot |\mathcal{B}_j|$ for all unordered bucket pairs (i, j) with $i < j$, and normalize such that:

$$\sum_{i<j} w_{ij} \approx n$$

6. For each bucket pair (i, j) with $i < j$, sample trajectory pairs $(\tau_i, \tau_j) \in \mathcal{B}_i \times \mathcal{B}_j$ until w_{ij} valid feedback pairs are collected, subject to the confidence-separated preference condition:

$$\hat{R}(\tau_i) + z \cdot u(\tau_i) < \hat{R}(\tau_j) - z \cdot u(\tau_j)$$

Let $\mathcal{F}_{\text{aug}}$ be the set of all such trajectory pairs.

7. Combine the original labeled feedback with the augmented set:

$$\mathcal{F}_{\text{refined}} = \mathcal{F}_\ell \cup \mathcal{F}_{\text{aug}}$$

B. Bucket-based sampling and uncertainty-based filtering

B.1. KL Divergence of Augmented Feedback from Random Sampling

Table 7. KL divergence of reward models under different feedback selection strategies across Metaworld tasks. **CS** = Confidence-based Sampling, **UF** = Uncertainty-based Filtering, **BS+UF (CUDA)** = Bucket-based Sampling combined with Uncertainty Filtering (our method).

Method	button-press -topdown	box-close	dial-turn	sweep	button-press -topdown-wall	sweep-into	drawer-open	lever-pull	avg
CS	0.4291	0.3166	0.1411	0.2902	0.3836	0.2686	0.2566	0.3122	**0.2997**
UF	0.0617	0.0287	0.0360	0.0354	0.0545	0.0261	0.0403	0.0486	**0.0414**
BS+CS	0.0302	0.0126	0.0141	0.0216	0.0273	0.0115	0.0232	0.0466	**0.0234**
BS+UF (CUDA)	0.0305	0.0124	0.0138	0.0248	0.0257	0.0109	0.0227	0.0477	**0.0236**

B.2. Agreement Rate of Augmented Preference Pairs with True Rewards

Table 8. Pairwise preference prediction accuracy of reward models under different feedback selection strategies across Metaworld tasks. **CS** = Confidence-based Sampling, **UF** = Uncertainty-based Filtering, **BS+UF (CUDA)** = Bucket-based Sampling combined with Uncertainty Filtering (our method).

Method	button-press -topdown	box-close	dial-turn	sweep	button-press -topdown-wall	sweep-into	drawer-open	lever-pull	avg
CS	0.9990	0.9853	0.9621	0.9905	0.9967	0.9400	0.9951	0.9951	**0.9830**
UF	0.9809	0.9119	0.9000	0.9290	0.9742	0.8025	0.9564	0.9543	**0.9262**
BS+CS	0.9706	0.8883	0.8576	0.9205	0.9638	0.7510	0.9457	0.9520	**0.9062**
BS+UF (CUDA)	0.9696	0.8910	0.8575	0.9207	0.9627	0.7525	0.9453	0.9522	**0.9064**

B.3. True Reward Heatmap of Augmented Preference Pairs

Figure 7 shows the true reward heatmaps of the selected augmented preference pairs for each environment under CS and CUDA sampling strategies. The heatmaps visualize the relative sampling density of each trajectory pair compared to random sampling, with the color intensity indicating values roughly between 0.6 and 1.4.

B.4. True Reward Distribution per Bucket

Figure 8 shows a detailed visualization of the bucket structure in the *button-press-topdown* environment.

Subfigure (a) presents the pairwise preference prediction accuracy between buckets, with values closer to 1 indicating strong ordering agreement. Subfigures (b) and (c) show the distributions of true and predicted rewards in each bucket, respectively, highlighting the effectiveness of k-means-based bucketing in separating trajectories with different reward profiles.

B.5. Performance without k-Means

Table 9. Performance comparison with and without k-Means bucketing across environments. Results are reported as mean ± standard deviation. The better performing condition per row is highlighted.

Environment	**Without k-Means**	**With k-Means**
button-press-topdown	0.6108 ± 0.1916	0.5885 ± 0.4432
box-close	0.6220 ± 0.2349	0.6396 ± 0.2297
dial-turn	0.8984 ± 0.0596	0.9028 ± 0.0616
sweep	0.8544 ± 0.0914	0.8856 ± 0.0744
button-press-topdown-wall	0.2544 ± 0.1406	0.2956 ± 0.1164
sweep-into	0.3488 ± 0.1018	0.3608 ± 0.0985
drawer-open	0.0536 ± 0.0584	0.0408 ± 0.0631
lever-pull	0.8608 ± 0.0731	0.8420 ± 0.0581
Average	0.5629 ± 0.3231	0.5635 ± 0.3193

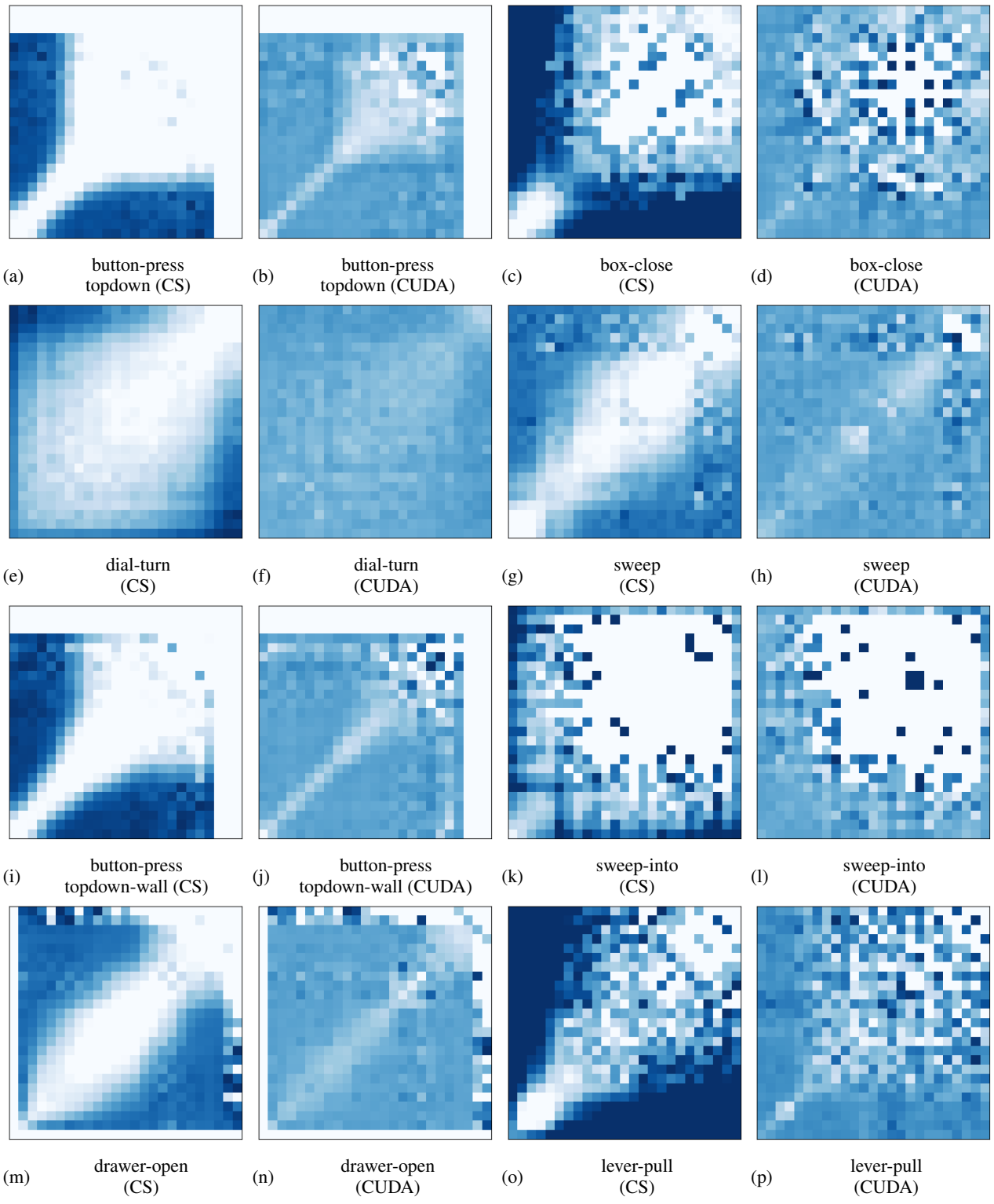
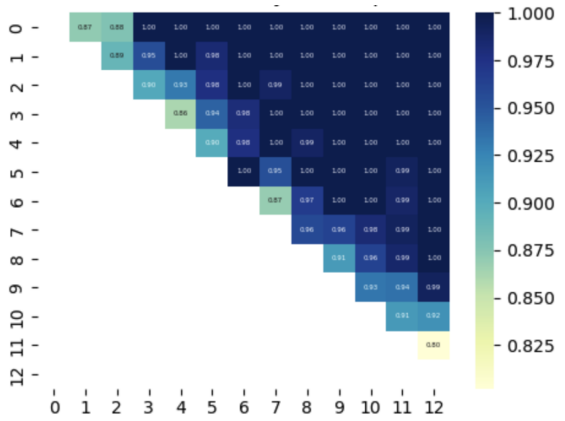
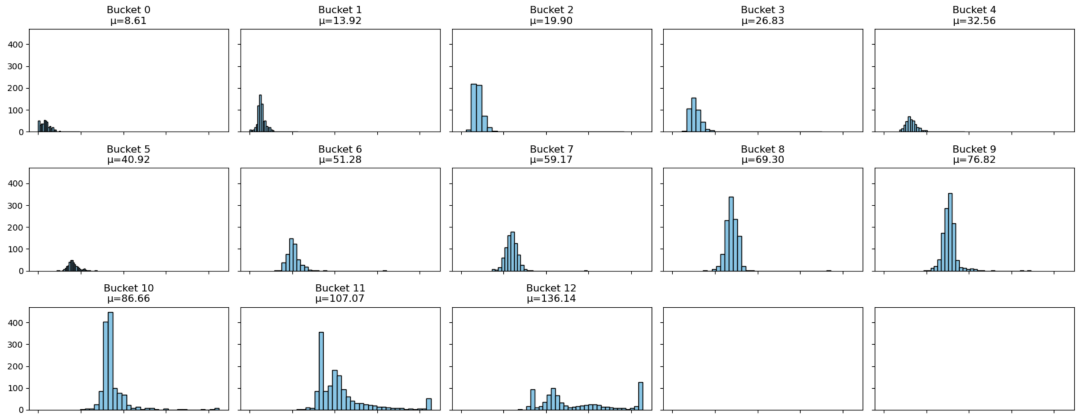


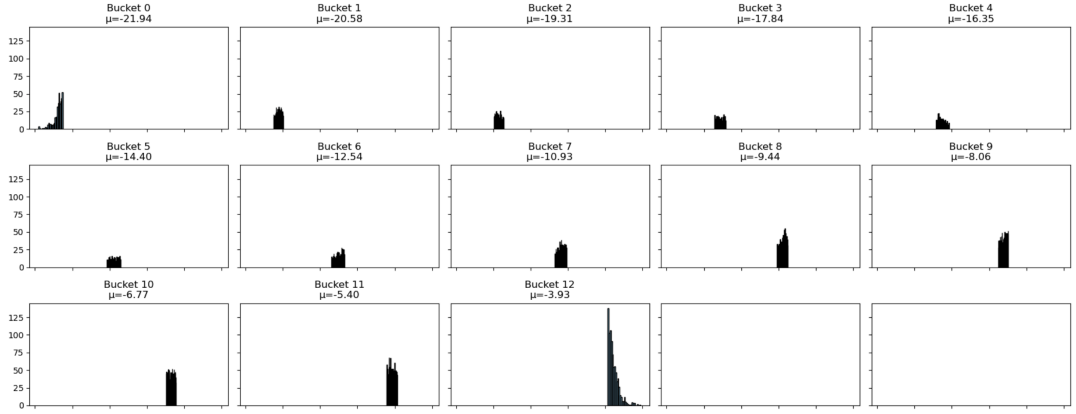
Figure 7. Heatmaps of true rewards for augmented preference pairs.



(a) Inter-bucket preference accuracy matrix



(b) True reward distribution per bucket



(c) Predicted reward distribution per bucket

Figure 8. Analysis of the k-means bucket structure in the *button-press-topdown* environment. Each trajectory is assigned to a bucket based on predicted return. (a) shows the pairwise preference prediction accuracy across buckets, while (b) and (c) visualize the true and predicted reward distributions within each bucket.