Latent Embedding Adaptation for Human Preference Alignment in Diffusion Planners

Wen Zheng Terence Ng[1,2], Jianda Chen[1], Yuan Xu[1], Tianwei Zhang[1]
[1]Nanyang Technological University, [2]Continental Automotive Singapore
{ngwe0099, jianda001, xu.yuan, tianwei.zhang}@ntu.edu.sg

***Abstract*— This work addresses the challenge of personalizing trajectories generated in automated decision-making systems by introducing a resource-efficient approach that enables rapid adaptation to individual users' preferences. Our method leverages a pretrained conditional diffusion model with Preference Latent Embeddings (PLE), trained on a large, reward-free offline dataset. The PLE serves as a compact representation for capturing specific user preferences. By adapting the pretrained model using our proposed preference inversion method, which directly optimizes the learnable PLE, we achieve superior alignment with human preferences compared to existing solutions like Reinforcement Learning from Human Feedback (RLHF) and Low-Rank Adaptation (LoRA). To better reflect practical applications, we create a benchmark experiment using real human preferences on diverse, high-reward trajectories.**

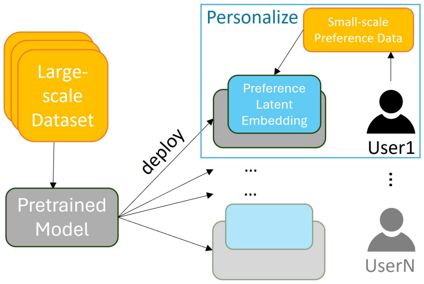


Fig. 1: **Overview of personalizing decision-making models.** We leverage large-scale offline data for pretraining, followed by rapid and efficient personalization using small-scale preference data.

I. Introduction

In today's increasingly automated world, personalization is crucial for decision-making systems to effectively cater to individual's needs, preferences, and circumstances. Tailoring experiences enhances the system effectiveness and user satisfaction across diverse applications, such as customizing self-driving vehicles [1], [2], transforming robotic assistants into adaptive companions [3]–[7], and optimizing prosthetics for wearers' unique requirements [8]–[10]. However, accurately capturing and aligning the abstract and dynamic human preferences with automated systems remains a complex challenge [11]–[13].

This work tackles this personalization challenge in trajectories generated by automated decision-making systems, aiming to create adaptable and reusable models that cater to individual user's needs [14]–[17]. While large-scale pretrained models offer broad capabilities [18]–[21], they lack the individual customization, and training personalized models for every user is infeasible. In contrast, it is more promising to first pretrain a model on large-scale offline data, and then align it with human preferences using smaller, user-specific preference datasets, as shown in Figure 1. The adoption of pretrained models from offline data avoids costly or risky direct interaction, enabling broader applications in challenging environments [22]–[24]. The process for adapting human preferences must be computationally efficient, enabling updates for many users and deployment on edge devices, through minimal data requirements. Therefore, we adopt this pretrain-align framework to achieve personalized decision making.

However, there are still a couple of difficulties to realize this system. (1) For pretraining the decision-making models, some approaches [25], [26] perform the training without rewards, but require the online interaction with the environment, which is not applicable in our offline setting. Other approaches with offline reinforcement learning (RL) [27]–[30] relies on rewards, which are often unavailable or difficult to quantify for human preferences. It is important but challenging to address both requirements simultaneously. (2) For adapting models to human preferences, RLHF [12] has emerged as a key technique for integrating human preferences into decision-making systems [31]–[34]. It works by first learning reward models to capture individual preferences and then refining policies based on those learned reward models. *Direct policy optimization* (DPO) offers an alternative approach by directly aligning policies with human preferences, bypassing the need for a separate reward model [35]. However, both RLHF and DPO face computational challenges due to the large number of parameters involved during alignment, making them less resource-efficient. Furthermore, both methods require careful tuning to prevent the adapted model from deviating too far from the base model.

We propose a pretrain-align framework to enable efficient and rapid personalized decision-making. Our solution is built atop of diffusion-based planners [36], which leverage the expressive power of diffusion models [37] to learn flexible and tractable models for trajectory generation. We introduce *preference latent embeddings* (PLE), low-dimensional vectors that effectively encode human preferences, for rapidly adapting pretrained models to individual user preferences.

Our method involves three stages: (1) Pretraining a diffusion model without reward supervision using a large state-action-only sequence dataset, (2) adapting the model to specific preferences using a small set of human labels, and (3) generating trajectories aligned with the learned preferences. Results demonstrate that our method adapts more accurately to human preferences with less data, both in offline datasets and in our custom dataset with real human labels. Our key contributions can be summarized as follows:

- Introducing a reward-free pretraining approach that jointly learns meaningful representations for PLE.
- Proposing an adaptation method for rapid preference alignment through preference inversion.
- Creating a benchmark experiment using real human preferences on diverse, high-reward trajectories.
- Conducting detailed evaluations and ablation studies using both our human-annotated dataset and an existing dataset.

II. Background and Related Work

Diffusion Probabilistic Models (DPMs). These powerful and versatile generative models offer a high degree of flexibility and tractability in modeling complex data distributions [37], [38]. The core principle behind DPMs is to learn to reverse a diffusion process by progressively denoising data points that have been transformed into random noise through a forward Markov chain. Given a data point sampled from a real data distribution, $\mathbf{x}_0 \sim q(\mathbf{x})$, this chain is defined by $q(x_k|x_{k-1}) = \mathcal{N}(x_k|\sqrt{\alpha_k}x_{k-1}, (1-\alpha_k)I)$, where $\mathcal{N}(\mu, \Sigma)$ represents a Gaussian distribution with mean μ and covariance Σ, and α_k determines the noise schedule with discrete noise time-step k. From these noise-augmented data points, a variational reverse Markov chain, parameterized by $p_\theta(x_{k-1}|x_k) = \mathcal{N}(x_{k-1}|\mu_\theta(x_k,k), \Sigma_\theta(x_k,k))$, is used to reconstruct the original data point x_0. [37] introduced an optimized surrogate loss function to simply this process:

$$\mathscr{L}(\theta) = \mathbb{E}_{k\sim[1,K],x_0\sim q,\varepsilon\sim\mathcal{N}(0,I)}\left[\|\varepsilon - \varepsilon_\theta(x_k,k)\|^2\right], \quad (1)$$

where ε is the sampled noise and ε_θ is the noise predicting model. By optimizing this loss, new data points can be generated through a sampling process via the forward Markov chain with $\mu_\theta(x_k,k) = \frac{1}{\sqrt{\alpha_k}}\left(x_k - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}k}}\varepsilon_\theta(x_k,k)\right)$.

DPMs can be extended to conditional generation using the *classifier-free* guidance [39], which enables the conditional model, $p_\theta(x_{k-1}|x_k,c)$, to generate samples conditioned on a context input c. During sampling, the predicted noise is adapted to a weighted combination of conditional and non-conditional sampling: $\hat{\varepsilon}_\theta(x_k,c,k) = (1+v)\varepsilon_\theta(x_k,c,k) - v\varepsilon_\theta(x_k,\varnothing,k)$, where $\varnothing$ is the null context and v controls the balance between sample quality and diversity. Alternatively, in *classifier-guided* diffusion [40], a separate classifier $h_\phi(c|\mathbf{x}_k,k)$ is trained, and the gradient $\nabla_\mathbf{x}\log h_\phi(c|\mathbf{x}_k)$ is used for the classifier-guided sampling as follows: $\bar{\varepsilon}_\theta(\mathbf{x}_k,k) = \varepsilon_\theta(x_k,k) - \sqrt{1-\bar{\alpha}_k}\, v\nabla_{\mathbf{x}_k}\log h_\phi(c|\mathbf{x}_k)$.

Diffusion Planning. Offline RL is a setting where an agent aims to learns an optimal policy from a fixed, previously collected dataset without further interaction with the environment [27], [41]. This problem can be framed as a sequence modeling task [42], [43]. Recently, diffusion-based planners [36], [44] have utilized DPMs to generate trajectories which can address the challenges of Offline RL, as discussed in [27]. One typical example of diffusion planning is *Diffuser* [36], which utilizes expressive DPMs to model trajectories in the following form:

$$\tau = \begin{bmatrix} s_0 & s_1 & \ldots & s_H \\ a_0 & a_1 & \ldots & a_H \end{bmatrix}, \quad (2)$$

where H is the planning horizon. The model is optimized based on Equation 1, with $\varepsilon_\theta(\tau_k,k)$ being modeled by U-Nets [45], chosen for their non-autoregressive, temporally local, and equivariant characteristics. A separate model $\mathscr{J}_\phi$ is trained to predict the cumulative rewards, and the gradients of $\mathscr{J}_\phi$ are used to guide the trajectory following the classifier-guided sampling procedure. Another example is *Decision Diffuser* [44], which adopts a classifier-free approach [39] and utilizes reward information as context to generate high-return trajectories. In this work, We adopt the classifier-free approach for its flexibility in incorporating contextual information, a key requirement of our method.

Inversion for Image Manipulation. In the domain of *generative adversarial networks* (GANs) [46], manipulating images often involves finding the corresponding latent representation of a given image, a process known as *inversion* [47], [48]. This can be achieved through optimization-based techniques [49]–[51], which directly optimize a latent vector to recreate the target image when passed through the GAN, or through the use of encoders [52]–[54]. Similarly, in the domain of DPMs, inversion enables image manipulations such as cross-image interpolations and semantic editing in *DALL-E 2* [55]. Lastly, textual inversion [56] represents visual concepts as novel tokens in a frozen text-to-image model, enabling personalized embeddings.

Preference Learning. It has proven to be effective to leverage relative human judgments through pairwise preference labels for optimizing human preferences without direct access to the reward function. This approach shows significant success in various natural language processing tasks, such as translation [57], summarization [58], [59], story-telling [59], and instruction-following [60], [61]. It typically learns a reward function using a preference model like the Bradley-Terry model [62], and subsequently trains the model using RL algorithms [63], [64] to maximize the learned reward. *Direct policy optimization* (DPO) has been proposed as an alternative to directly align the policy with human preferences and learn from collected data without a separate reward model [35]. DPO variants [65]–[68] have shown great alignment with human preferences that matches or surpasses reward-based methods. In the domain of RL, learning policies from preferences has been studied, as designing a suitable reward function can be challenging. Various approaches have been proposed [12], [25], [31]–[34] that learn a reward function from trajectory segment pairs.

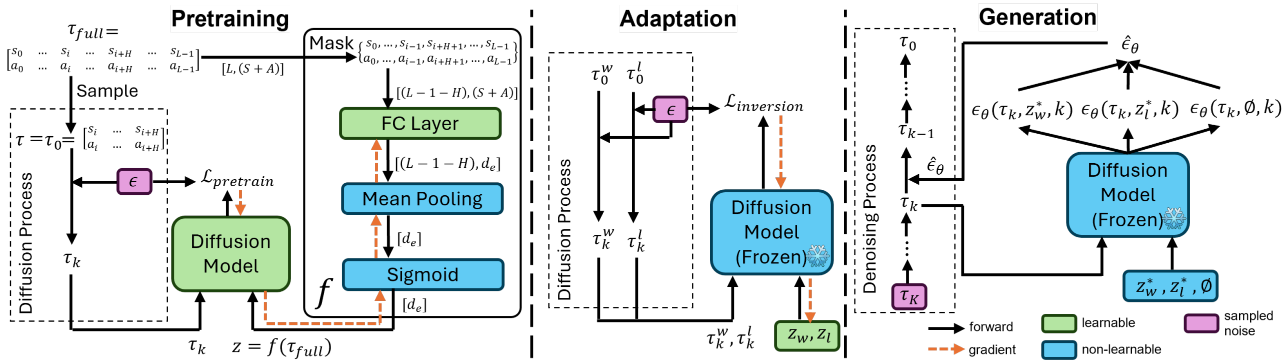


Fig. 2: **Overview of the proposed method.** (Left) Pretraining: A placeholder for preference latent embedding (PLE), z, is co-trained with the diffusion model, without reward supervision. (Middle) Adaptation: With diffusion model weights frozen, PLEs are aligned to user labelled query pairs via preference inversion. (Right) Generation: Conditional sampling with learned PLEs generate trajectories that match the users' preference.

III. Methodology

To enable rapid adaptation of our pretrained model to individual user preferences, we introduce the concept of ***preference latent embeddings*** (PLE), denoted by z. PLEs are low-dimensional vectors that encode human preferences efficiently. Our method comprises three distinct stages, as illustrated in Figure 2: **pretraining, adaptation, and generation**. During **pretraining**, we train the diffusion model without reward supervision, initializing a placeholder for the PLE to establish a general-purpose generative model. In **adaptation**, a small set of human preference labels fine-tunes the model, identifying the PLE that aligns with user preferences in a low-dimensional representation. Finally, in **generation**, the learned PLE guides trajectory generation to match encoded human preferences. We now detail each stage.

A. Pretraining with Masked Trajectories

The pretraining stage of our method addresses two concurrent goals: training a general-purpose generative model that comprehensively understands the task domain without reward supervision, and initializing a meaningful representation for the PLE placeholder. To achieve the first goal, we employ the decision diffuser [44], which excels in training on offline trajectory datasets without explicit reward requirements. Its ability to incorporate additional context is crucial for our second goal.

For the second goal, we posit that each preference correlates with a set of similar trajectories. Consequently, we aim to map similar trajectories to similar embeddings to give structure for the PLE placeholder. We propose a learnable mapping function: $f:\mathbb{R}^{L\times(S+A)}\rightarrow\mathbb{R}^{d_e}, z:=f(\tau_{\text{full}})$, where τ_{full} represents the full trajectory from which a sub-trajectory τ is sampled. L denotes the full trajectory length, S and A are the dimensions of the state and action spaces respectively, and d_e is the PLE dimension, a tunable hyperparameter. The mapping f consists of a sequence of operations as visualized in Figure 2 (left). Initially, we apply a fixed mask to τ_{full} to prevent information leakage about τ. This is followed by a transformation of state-action features into the latent space using a learnable feed-forward layer. We then apply mean pooling to mitigate time leakage and accommodate variable horizon lengths. Normalization is achieved via sigmoid activation, a choice motivated by the adaptation process described in the subsequent section. Finally, we compose these differentiable components to obtain the PLE placeholder, z. The resulting z is then fed into the decision diffuser as context, training end-to-end to achieve both goals simultaneously with the following objective:

$$\mathscr{L}_{\text{pretrain}}(\theta)=\mathbb{E}_{k\sim[1,K],x_0\sim q,\varepsilon\sim\mathcal{N}(0,I)}\left[\left\|\varepsilon-\varepsilon_\theta\left(\tau_k,f(\tau_{\text{full}}),k\right)\right\|^2\right].$$

The proposed objective serves two crucial purposes in our approach: constructing a representation for the PLE placeholder that groups similar trajectories, and pretraining the model to generate trajectories adhering to the offline dataset's distribution. However, the model remains unaligned with specific user preferences at this stage.

B. Adaptation via Preference Inversion

During adaptation, our goal is to quickly identify the PLE that aligns with the user's preferences. To achieve this, we utilize a small set of human preference labels to partially fine-tune the pretrained model, rather than performing full fine-tuning. This approach is made possible by the placeholder PLE trained during the pretraining step. We begin the adaptation with a randomly initialized, learnable PLE, z. The sigmoid activation applied during the pretraining stage enables us to choose a prior bounded between 0 and 1 for this initialization. To align z, we freeze all weights of the diffuser model and backpropagate the loss gradients towards z. We refer to this PLE alignment process as ***preference inversion***, drawing an analogy to the inversion methods for image manipulation described in Section II.

To design a loss function that leverages pairwise preference labels, we sub-categorize the PLE into two types: winner PLE z_w, and loser PLE z_l. This enables us to optimize and obtain learned preferences z_w^* and z_l^* based on the reconstruction loss of the respective winner and loser trajectories, x^w and x^l as follows:

$$\mathscr{L}_{\text{inversion}}(z^w, z^l) = \mathbb{E}_{k\sim[1,K],x_0\sim q,\varepsilon\sim\mathcal{N}(0,I)} \Big[\|\varepsilon - \varepsilon_\theta(\tau_k^w, z_w, k)\|^2 + \left\|\varepsilon - \varepsilon_\theta\left(\tau_k^l, z_l, k\right)\right\|^2 \Big].$$

where θ is fixed. A key advantage of having a frozen pretrained model is that it does not require the additional constraints to remain aligned to the base model, as is the case with RLHF or DPO. During training, it is possible to simultaneously optimize z_w and z_l within a single batch, since their gradients do not interfere with each other.

C. Generating Preferred Trajectories

To sample a trajectory aligned with human preferences, we utilize a linear combination of the winner and loser PLEs, similar to the approach used in [39] to predict noise:

$$\hat{\varepsilon}_\theta\left(x_t, z_w^*, z_l^*, k\right) = (1+v)\dot{\varepsilon}_\theta\left(x_t, z_w^*, z_l^*, k\right) - v\varepsilon_\theta\left(x_t, \varnothing, k\right),$$
$$\text{where } \dot{\varepsilon}_\theta\left(x_t, z_w^*, z_l^*, k\right) = (1+u)\varepsilon_\theta\left(x_t, z_w^*, k\right) - u\varepsilon_\theta\left(x_t, z_l^*, k\right).$$

Here, v and u are hyper-parameters. v controls the strength of the guidance, while u controls the influence of the loser information. To gain an intuition, we rewrite $\dot{\varepsilon}_\theta\left(x_t, z_w^*, z_l^*, k\right) = \varepsilon_\theta\left(x_t, z_w^*, k\right) + u(\varepsilon_\theta\left(x_t, z_w^*, k\right) - \varepsilon_\theta\left(x_t, z_l^*, k\right))$, which shows that we are pushing the score estimations away from the loser, originating at the winner. This allows us to efficiently leverage the pretrained model while quickly adapting to individual user preferences using a small amount of preference data. By learning only the low-dimensional PLE while keeping our pretrained model fixed, we reduce computational cost and enhance the stability of the adaptation process compared to fine-tuning the entire model. The overall proposed method is illustrated in the Figure 2.

IV. Experiments

We comprehensively evaluate the effectiveness of our method in integrating user preferences, examining pretraining performance, the impact of preference queries and z_l^*, and adaptation stability. Additionally, we assess its ability to capture human preferences from diverse, high-reward queries using a custom dataset. For a strong benchmark, we compare against diverse baselines.

- **Diffuser:** A pretrained diffusion-based planner [69] representing the distribution of training dataata but not adapted to user labels.
- **Guided Sampling:** Following RLHF, we train a reward model using the Bradley-Terry model [62], but employ classifier-guidance sampling [40].
- **Finetuning (Full):** This baseline directly fine-tunes the pretrained Diffuser model using DPO [35] with $\beta = 5000$.
- **Finetuning (LoRA):** This baseline provides a more direct comparison with our proposed method, where partial finetuning is performed. To achieve this, we utilize LoRA [70] with a rank of $r = 8$.
- **Preference Transformers:** Transformer-based architecture for modeling preferences over trajectories [34].
- **Preference Inversion (Proposed):** Our method partially fine-tunes the pretrained model to retrieve the user preference context. The pretraining stage utilizes a diffuser conditioned on masked trajectories.

All hyperparameters related to the diffusion model follow those in [69].

Experimental Setup. To examine the effectiveness of various personalization methods in automated decision-making systems, we tested our approach on a preference learning benchmark [34] that utilizes challenging control tasks from the d4rl dataset [41] in an offline setting. Specifically, we used the Hopper, HalfCheetah, and Walker2D tasks from the d4rl dataset, focusing on the medium-expert and medium-replay settings. For the preference labels, we follow [34], where query pairs (pairs of trajectory segments) are randomly sampled from the D4RL offline dataset. Within each pair, the trajectory segment with the higher return is designated as the winner, with the other segment consequently labeled as the loser. To ensure proper convergence of our large dataset, all baselines underwent 1 million updates for pretraining. During the alignment stage, we fixed $N_{\text{adapt}} = 5000$ updates for our main experiments and also conducted ablation studies on N_{adapt}. Pretraining utilizes the full dataset for its respective tasks. During evaluation, we sample each method for 100 episodes across 5 different random seeds within their designated environments.

A. Latent Space Analysis

To understand the impact of our proposed mapping in the pretraining for the PLE placeholder, we conducted a visualization of the latent space representation. Using their respective pretrained models, we sampled 1000 random trajectories from each dataset and obtained the PLE, z. These embeddings were then projected onto a two-dimensional space using t-SNE (with perplexity set to 30) as seen in Figure 3, with color intensity representing the normalized score of the corresponding masked trajectory. The normalized score is defined as normalized score $= 100 \times \frac{\text{score}-\text{random score}}{\text{expert score}-\text{random score}}$ as in [41]. Examining the latent space of the medium-expert dataset, we observed distinct clusters representing low, medium, and high returns, respectively. These clusters were well separated and mostly linearly separable. In contrast, the medium-replay dataset, which consists of the entire replay buffer with a continual range of returns, exhibited a different pattern. The latent space reflected this continuous return distribution, showing not distinct clusters but rather a smooth transition of the latent embeddings based on their return values.

Overall, our proposed pretraining enables the PLE placeholder to be structured in an organized and meaningful manner, demonstrating that our proposed mapping, f, is able to organize similar trajectories close together. This organized latent space could accelerate the preference inversion process by first navigating the loss landscape to a local region of similar trajectories, and then refining the search for a more precise alignment with the true reward (human preference).

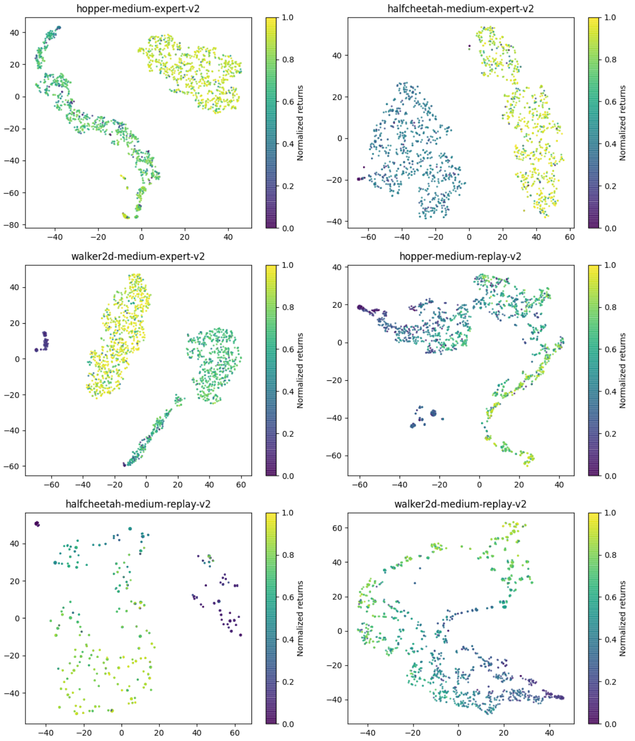


Fig. 3: **Latent space analysis**: We visualize t-SNE plots of PLEs post-pretraining, where each point represents a trajectory, and color intensity reflects its normalized score. The smooth gradient in return distribution indicates that our pretraining effectively structures the PLE space.

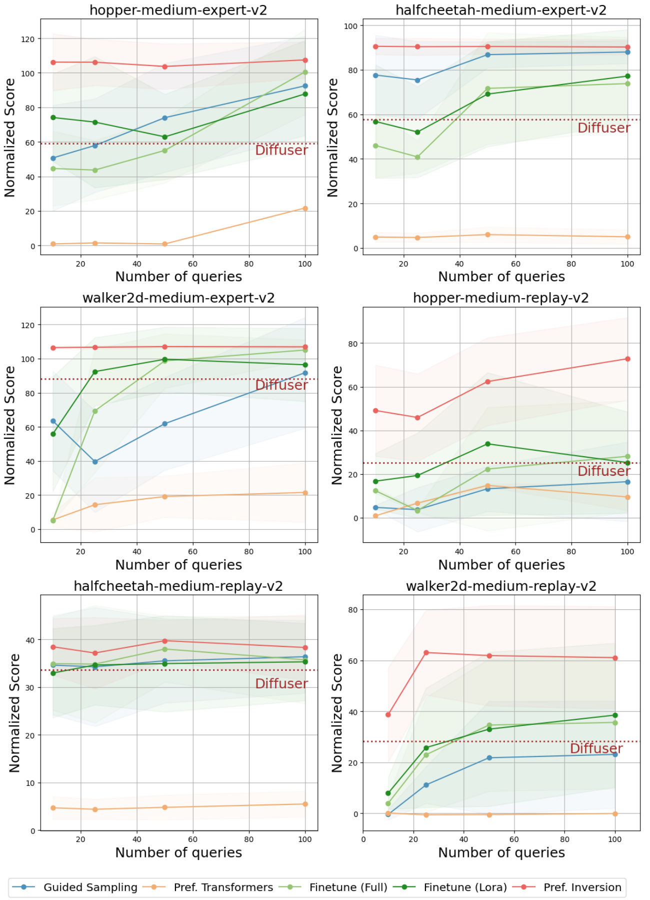


Fig. 4: **Main results evaluated over different numbers of queries across six control tasks report the normalized score.**

B. Main Results

We compare our method against the baselines and assess the impact of the number of query pairs, N_{query}. We evaluate each model with N_{query} values of 10, 25, 50, and 100, analyzing their ability to align with user preferences using a small set of human-annotated queries and a limited number of updates ($N_{\text{adapt}} = 5000$). For the main experiment, we set $d_e = 16$ and $u = 0.02$.

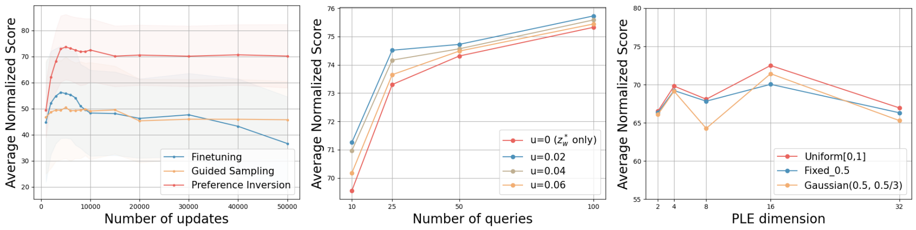
Our experimental results (Figure 4) show that our method consistently outperforms baselines, with a growing advantage as queries decrease. This is particularly evident in hopper-medium-replay and walker2d-medium-replay tasks. No baseline consistently outperforms others: LoRA fine-tuning closely matches full fine-tuning, with slight gains at lower queries (N=10, 25) and mixed results for N=50, 100, while Preference Transformers perform worst, likely due to training from scratch without pretrained models. Baselines (except Preference Transformers) surpass the pretrained Diffuser when $N_{\text{query}} > 50$ but often fail with fewer queries, leading to negative adaptation. In contrast, our method consistently outperforms Diffuser, maintaining high performance even at $N_{\text{query}} = 10$. Performance correlates positively with N_{query}, with 50 queries generally sufficient across tasks, except for hopper-medium-expert, which requires 100. These results validate our resource-efficient approach, reducing labeled data needs with minimal updates.

C. Ablation Studies

We perform a series of ablation experiments to gain deeper insights into the relative importance of different design choices and determine the sensitivity of our approach to variations in model components and hyperparameters.

Number of adaptation steps. Figure 5a shows that all methods peak around $N_{\text{adapt}} = 5000$. Preference inversion and guided sampling remain stable after minor drops at 15000 and 20000 updates, respectively. However, the performance of finetuning consistently declines after peaking, with rapid deterioration after $N_{\text{adapt}} = 30000$, suggesting excessive deviation from the base model. The stability of our method is advantageous for practical applications, where the optimal stopping point is often unknown in advance.

Loser PLE. Figure 5b shows that incorporating the loser PLE, z_l^* with $u > 0$, consistently improves the sampling performance compared to using only the winner PLE with $u = 0$. The improvements peak at $u = 0.02$, resulting in $1 \sim 3\%$ gains across various N_{query} values, and gradually



(a) Adaptation stability across N_{adapt}.

(b) The impact of utilizing loser PLE, z_l^* for sampling

(c) Choice of different priors for initialization, choice of PLE dimension

Fig. 5: **A series of ablation experiments.** The average normalized score is reported across all tasks and N_{query}, except for the loser PLE analysis, where averaging is performed across tasks only.

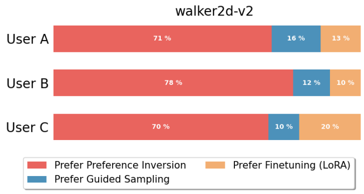
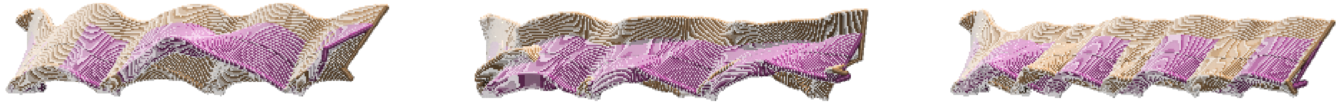


Fig. 6: **Survey Results on Real Human Preference.** Users select their preferred trajectories from samples generated by top 2 baselines and our proposed method.



(a) User A: ‘Back leg swing up high and body learning forward’

(b) User B: ‘Body upright, front knee bent, rear leg 45degrees’

(c) User C: ‘Gentle hopping with moderately fast strides’

Fig. 7: **Trajectories generated using our proposed method, an aligned model conditioned on user’s respective PLE.** The generated samples closely match each user’s description of their preference.

diminish as u decreases further. Utilizing z_l^* provides a small boost when N_{query} is high, but notably enhances the sampling when N_{query} is low.

Choice of prior and PLE dimension. Given that our PLE z is constrained to the range [0, 1], we test three different priors for initialising z within this interval: Uniform$[0,1]$, Gaussian$(0.5, 0.5/3)$, and Fixed_0.5. Figure 5c demonstrates that all three settings perform comparably well, with the uniform prior slightly outperforming the others. Similarly, varying the PLE dimension d_e across 2, 4, 8, 16, and 32 consistently yields good results, demonstrating relative insensitivity to this hyper-parameter, with a slight advantage observed at $d_e = 16$.

D. Real Human Preference on Quality Diversity Dataset

Previous experiments, following the design of [34], focus on recovering a hidden task reward. While useful for validating human preferences, it **prioritizes high-reward over diversity** and may not fully capture practical scenarios. In contrast, practical decision-making often involves selection from a **diverse set of high-reward** trajectories. To better reflect this, we design a new experiment based on Quality Diversity (QD) [71]–[73], which in policy learning refers to an algorithm’s ability to discover diverse, high-performing policies with distinct behaviors. To implement this, we train a set of QD policies based on [73], generating a diverse dataset of 750 high-reward Walker2D episodes for model pretraining without preference alignment. Additionally, we use QD policies to create query pairs and gather preference labels from three users. They are instructed to maintain a consistent selection strategy and provide written descriptions of their decision criteria. We then adapt the pretrained policy to each user’s preference labels, consisting of 100 query pairs each.

For evaluation, we generate 100 trajectories per baseline and ask users to choose the closest match to their preference criteria. The survey results, shown in Figure 6, indicate that our proposed method receives the vast majority of votes, demonstrating its effectiveness in capturing human preferences. Figure 7 qualitatively displays the sampled trajectories from the aligned model, generated using preference inversion, which closely match the users’ descriptions. This real human preference dataset provides a good initial indication of our method’s practical real-world applicability. To establish more robust findings, we plan to survey additional users in our future work.

V. Conclusion

This work presents a novel approach that enables a policy to quickly adapt to a small human preference dataset. It consists of pretraining followed by adaptation on latent embeddings via preference inversion for rapid alignment. Evaluation results demonstrate that our method adapts more accurately to human preferences with minimal preference labels, outperforming baselines in both offline datasets and our custom dataset with real human labels. This promising method shows potential for further applications across diverse settings.

Acknowledgment

This study is supported under RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

[1] I. Bae, J. Moon, J. Jhung, H. Suk, T. Kim, H. Park, J. Cha, J. Kim, D. Kim, and S. Kim, "Self-driving like a human driver instead of a robocar: Personalized comfortable driving experience for autonomous vehicles," *arXiv preprint arXiv:2001.03908*, 2020.

[2] X. He and C. Lv, "Toward personalized decision making for autonomous vehicles: a constrained multi-objective reinforcement learning technique," *Transportation research part C: emerging technologies*, vol. 156, p. 104352, 2023.

[3] E. OhnBar, K. Kitani, and C. Asakawa, "Personalized dynamics models for adaptive assistive navigation systems," in *Conference on Robot Learning*. PMLR, 2018, pp. 16–39.

[4] Y. Gao, W. Barendregt, M. Obaid, and G. Castellano, "When robot personalisation does not help: Insights from a robot-supported learning study," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2018, pp. 705–712.

[5] C. Moro, G. Nejat, and A. Mihailidis, "Learning and personalizing socially assistive robot behaviors to aid with activities of daily living," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 7, no. 2, pp. 1–25, 2018.

[6] Y. Wen, J. Si, A. Brandt, X. Gao, and H. H. Huang, "Online reinforcement learning control for the personalization of a robotic knee prosthesis," *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2346–2356, 2019.

[7] B. Woodworth, F. Ferrari, T. E. Zosa, and L. D. Riek, "Preference learning in assistive robotics: Observational repeated inverse reinforcement learning," in *Machine learning for healthcare conference*. PMLR, 2018, pp. 420–439.

[8] X. Tu, M. Li, M. Liu, J. Si, and H. H. Huang, "A data-driven reinforcement learning solution framework for optimal and adaptive personalization of a hip exoskeleton," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 610–10 616.

[9] Q. Zhang, V. Nalam, X. Tu, M. Li, J. Si, M. D. Lewek, and H. H. Huang, "Imposing healthy hip motion pattern and range by exoskeleton control for individualized assistance," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 126–11 133, 2022.

[10] V. Nalam, X. Tu, M. Li, J. Si, and H. H. Huang, "Admittance control based human-in-the-loop optimization for hip exoskeleton reduces human exertion during walking," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6743–6749.

[11] C. Wirth, R. Akrour, G. Neumann, J. Fürnkranz, *et al.*, "A survey of preference-based reinforcement learning methods," *Journal of Machine Learning Research*, vol. 18, no. 136, pp. 1–46, 2017.

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

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

[14] A. Haydari and Y. Yılmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 11–32, 2020.

[15] C. Yu, J. Liu, S. Nemati, and G. Yin, "Reinforcement learning in healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–36, 2021.

[16] Z. Zhang, D. Zhang, and R. C. Qiu, "Deep reinforcement learning for power system applications: An overview," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 1, pp. 213–225, 2019.

[17] R. Nian, J. Liu, and B. Huang, "A review on reinforcement learning: Introduction and applications in industrial process control," *Computers & Chemical Engineering*, vol. 139, p. 106886, 2020.

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[20] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, "A generalist agent," *Transactions on Machine Learning Research*, 2022.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[22] M. Zare, P. Mohsenzadeh Kebria, and A. Khosravi, "Leveraging optimal transport for enhanced offline reinforcement learning in surgical robotic environments," *Abbas, Leveraging Optimal Transport for Enhanced Offline Reinforcement Learning in Surgical Robotic Environments*.

[23] K. Fan, Z. Chen, G. Ferrigno, and E. De Momi, "Learn from safe experience: Safe reinforcement learning for task automation of surgical robot," *IEEE Transactions on Artificial Intelligence*, 2024.

[24] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *IEEE international conference on robotics and automation*, 2018.

[25] K. Lee, L. M. Smith, and P. Abbeel, "Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6152–6163.

[26] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," *arXiv preprint arXiv:1802.06070*, 2018.

[27] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[28] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.

[29] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=68n2s9ZJWF8

[30] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 20 132–20 145, 2021.

[31] W. B. Knox, S. Hatgis-Kessell, S. Booth, S. Niekum, P. Stone, and A. Allievi, "Models of human preference for learning reward functions," *arXiv preprint arXiv:2206.02231*, 2022.

[32] D. J. Hejna III and D. Sadigh, "Few-shot preference learning for human-in-the-loop rl," in *Conference on Robot Learning*. PMLR, 2023, pp. 2014–2025.

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

[34] C. Kim, J. Park, J. Shin, H. Lee, P. Abbeel, and K. Lee, "Preference transformer: Modeling human preferences using transformers for RL," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Peot1SFDX0

[35] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *arXiv preprint arXiv:2305.18290*, 2023.

[36] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[37] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[38] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[39] J. Ho, "Classifier-free diffusion guidance," *ArXiv*, vol. abs/2207.12598, 2022.

[40] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[41] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.

[42] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.

[43] M. Janner, Q. Li, and S. Levine, “Offline reinforcement learning as one big sequence modeling problem,” *Advances in neural information processing systems*, vol. 34, pp. 1273–1286, 2021.

[44] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, “Is conditional generative modeling all you need for decision making?” in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=sP1fo2K9DFG

[45] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[46] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[47] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, “Gan inversion: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 3121–3138, 2022.

[48] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*. Springer, 2016, pp. 597–613.

[49] R. Abdal, Y. Qin, and P. Wonka, “Image2stylegan: How to embed images into the stylegan latent space?” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4432–4441.

[50] ——, “Image2stylegan++: How to edit the embedded images?” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8296–8305.

[51] J. Gu, Y. Shen, and B. Zhou, “Image processing using multi-code gan prior,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3012–3021.

[52] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, “Encoding in style: a stylegan encoder for image-to-image translation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2287–2296.

[53] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, “In-domain gan inversion for real image editing,” in *European conference on computer vision*. Springer, 2020, pp. 592–608.

[54] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, “Designing an encoder for stylegan image manipulation,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[55] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[56] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, “An image is worth one word: Personalizing text-to-image generation using textual inversion,” *arXiv preprint arXiv:2208.01618*, 2022.

[57] J. Kreutzer, J. Uyheng, and S. Riezler, “Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning,” *arXiv preprint arXiv:1805.10627*, 2018.

[58] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, “Learning to summarize with human feedback,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 3008–3021, 2020.

[59] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, “Fine-tuning language models from human preferences,” *arXiv preprint arXiv:1909.08593*, 2019.

[60] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, “Training language models to follow instructions with human feedback,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[61] R. Ramamurthy, P. Ammanabrolu, K. Brantley, J. Hessel, R. Sifa, C. Bauckhage, H. Hajishirzi, and Y. Choi, “Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization,” *arXiv preprint arXiv:2210.01241*, 2022.

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

[63] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, pp. 229–256, 1992.

[64] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[65] M. G. Azar, M. Rowland, B. Piot, D. Guo, D. Calandriello, M. Valko, and R. Munos, “A general theoretical paradigm to understand learning from human preferences,” *arXiv preprint arXiv:2310.12036*, 2023.

[66] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela, “Kto: Model alignment as prospect theoretic optimization,” *arXiv preprint arXiv:2402.01306*, 2024.

[67] H. Xu, A. Sharaf, Y. Chen, W. Tan, L. Shen, B. Van Durme, K. Murray, and Y. J. Kim, “Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation,” *arXiv preprint arXiv:2401.08417*, 2024.

[68] Y. Zhao, R. Joshi, T. Liu, M. Khalman, M. Saleh, and P. J. Liu, “Slic-hf: Sequence likelihood calibration with human feedback,” *arXiv preprint arXiv:2305.10425*, 2023.

[69] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior synthesis,” in *International Conference on Machine Learning*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248965046

[70] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.

[71] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality diversity: A new frontier for evolutionary computation,” *Frontiers in Robotics and AI*, vol. 3, p. 202845, 2016.

[72] E. Conti, V. Madhavan, F. Petroski Such, J. Lehman, K. Stanley, and J. Clune, “Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents,” *Advances in neural information processing systems*, vol. 31, 2018.

[73] S. Wu, J. Yao, H. Fu, Y. Tian, C. Qian, Y. Yang, Q. Fu, and Y. Wei, “Quality-similar diversity via population based reinforcement learning,” in *The Eleventh International Conference on Learning Representations*, 2022.