POIL:Preference Optimization for Imitation Learning

Ren jyun Huang jyun.cs11@nycu.edu.tw
Kuan yen Liu albert321liu.mg11@nycu.edu.tw
Chang Chih Meng mcc.cs11@nycu.edu.tw
I Chen Wu icwu@cs.nycu.edu.tw
National Yang Ming Chiao Tung University

Editors: Hung-yi Lee and Tongliang Liu

Abstract

Imitation learning (IL) enables agents to learn policies by mimicking expert demonstrations. While online IL methods require interaction with the environment, which is costly, risky, or impractical, offline IL allows agents to learn solely from expert datasets without any interaction with the environment. In this paper, we propose Preference Optimization for Imitation Learning (POIL), a novel approach inspired by preference optimization techniques in large language model alignment. POIL eliminates the need for adversarial training and reference models by directly comparing the agent's actions to expert actions using a preference-based loss function. We evaluate POIL on MuJoCo control tasks and Adroit manipulation tasks. Our experiments show that POIL consistently delivers superior or competitive performance against state-of-the-art methods in the past, including Behavioral Cloning (BC), IQ-Learn, MCNN, and O-DICE, especially in data-scarce scenarios, such as using single trajectory. These results demonstrate that POIL enhances data efficiency and stability in offline imitation learning, making it a promising solution for applications where environment interaction is infeasible and expert data is limited, even in high-dimensional and complex control tasks.

Keywords: Offline Imitation Learning; Preference Optimization

1. Introduction

Reinforcement learning (RL) (Sutton, 2018) has achieved remarkable success across various domains, including video games (Mnih et al., 2015; Schrittwieser et al., 2020), robotics (Kober et al., 2013), and even nuclear fusion control (Degrave et al., 2022). However, defining suitable reward functions remains a significant challenge (Eschmann, 2021), especially in tasks where desired behaviors are abstract or hard to specify, such as control problems (Kiumarsi et al., 2017). Poorly designed reward functions can lead to unintended or unsafe behaviors (Amodei et al., 2016), and deep RL algorithms are often sensitive to reward sparsity (Ladosz et al., 2022), complicating the development of effective reward signals.

Imitation learning (IL) (Zare et al., 2024) offers an alternative by learning policies directly from expert demonstrations without requiring explicit reward functions. In *online IL*, the agent interacts with the environment to learn the expert's behavior. Prominent methods like generative adversarial imitation learning (GAIL) (Ho and Ermon, 2016), adversarial inverse reinforcement learning (AIRL) (Fu et al., 2017), and discriminator actor-critic (DAC) (Kostrikov et al., 2018) employ a generator (the policy) and a discriminator (distinguishing between expert and agent behaviors) in an adversarial setup to encourage the agent to mimic the expert closely (Garg et al., 2021). Despite their effectiveness, these methods face practical challenges: the adversarial optimization process can be unstable and difficult to train, leading to biased, high-variance gradient estimators and convergence issues (Garg et al., 2021). Moreover, the need for environment interaction makes them impractical in real-world scenarios where such interaction is costly, risky, or infeasible (Lyu, 2024; Prudencio et al., 2023).

To address these limitations, *offline IL* methods have been developed to learn from pre-collected expert demonstrations without environment interaction. Behavior cloning (BC) (Pomerleau, 1991) is a straightforward approach that directly replicates expert actions through supervised learning. However, BC suffers from compounding errors due to distribution shifts and often requires large amounts of expert data. Recent advances aim to mitigate these issues by correcting for distribution discrepancies. The DICE family of algorithms, including ValueDICE (Kostrikov et al., 2019), DemoDICE (Kim et al., 2022), and O-DICE (Mao et al., 2024), improve upon BC by addressing distribution shift.

In this paper, we propose a novel offline imitation learning method called *preference optimization for imitation learning* (POIL), inspired by recent advances in preference-based alignment techniques in large language models (LLMs) (Wang et al., 2024), and clearly different from previous offline IL methods. An overview of the POIL process is illustrated in Figure 1, described in more detail in Subsection 3.3. Specifically, we draw inspiration from direct preference optimization (DPO) (Rafailov et al., 2024), contrastive preference optimization (CPO) (Xu et al., 2024), and self-play fine-tuning (SPIN) (Chen et al., 2024). These methods have been successful in aligning LLMs with human preferences but have specific requirements that limit their direct application to offline IL, namely, DPO and CPO require preference datasets and, in the case of DPO, a reference model. SPIN leverages an expert dataset but still relies on a reference model during training.

In contrast, POIL adapts these techniques to the offline IL setting by introducing a framework that directly compares the agent's actions to expert actions without the need for a discriminator or a reference model, as illustrated in Figure 1. By eliminating the need for preference datasets and reference models, POIL simplifies the learning process, avoids adversarial training instabilities, and enhances computational efficiency. This approach allows POIL to effectively overcome key constraints of existing methods in offline IL while improving overall performance.

We evaluate POIL on standard MuJoCo control tasks, including *HalfCheetah*, *Hopper*, and *Walker2d*, as well as on complex Adroit dexterous manipulation tasks (Rajeswaran et al., 2017). POIL consistently delivers superior or competitive results compared to state-of-the-art methods, particularly excelling in data-scarce scenarios, e.g., a single demonstration or small fractions of the dataset.

Our contributions are summarized as follows:

- We introduce POIL, a novel offline imitation learning method that eliminates the need for adversarial training, preference datasets, and reference models by directly comparing agent and expert actions.
- We provide empirical evidence on MuJoCo and Adroit tasks showing that POIL achieves superior or competitive performance against state-of-the-art methods in the past, especially in data-limited scenarios and complex manipulation tasks.

- We conduct ablation studies to analyze the impact of some key hyper-parameters on POIL's performance, providing insights into its robustness and applicability.

These results suggest that preference optimization techniques from LLM alignment are effectively adapted to offline imitation learning, opening new avenues for research and applications in control and robotics.

2. Related Work

2.1. Offline Imitation Learning

Offline imitation learning methods (Zare et al., 2024) learn from static datasets without needing interaction with the environment. An early approach, behavior cloning (Pomerleau, 1991), learns directly from expert demonstrations but has issues with compounding errors and distribution shift, especially with limited data.

To overcome these issues, the DICE (DIstribution Correction Estimation) family of algorithms provides improvements. ValueDICE (Kostrikov et al., 2019) minimizes the KL divergence between stationary distributions, while SoftDICE (Sun et al., 2021) employs the Earth-Mover distance for distribution matching. DemoDICE (Kim et al., 2022) can use demonstrations of varying quality, and ODICE (Mao et al., 2024) adds orthogonal-gradient updates to handle conflicting gradients in learning. Other methods include adaptations of *inverse reinforcement learning* for offline use (Zolna et al., 2020; Yue et al., 2023), energy-based models (Jarrett et al., 2020), and OTR (Luo et al., 2023). Despite these advances, challenges remain. Some methods, such as DemoDICE and SMODICE (Ma et al., 2022), need extra data beyond expert demonstrations. Others struggle with limited datasets or single demonstrations.

2.2. Preference-based Reinforcement Learning

Preference-based RL (PbRL) (Wirth et al., 2017) has emerged as a promising approach to address the challenges of reward function design in traditional RL by incorporating human preferences into the learning process. Early work by Christiano et al. (2017) demonstrated the potential of PbRL using deep learning techniques. This pioneering work opened new avenues for tackling challenging domains but relied heavily on external preference feedback. Subsequent research focused on reducing this dependency. Lee et al. (2021) introduced PEBBLE, which improved feedback efficiency through experience relabeling and unsupervised pre-training. Park et al. (2022) further improved this with SURF, a semi-supervised approach leveraging data augmentation, yet neither fully eliminated the need for human feedback.

A significant advancement in the field came with the inverse preference learning (IPL) proposed by Hejna and Sadigh (2024). IPL represents a novel approach specifically designed for learning from offline preference data. It leverages the key insight that for a fixed policy, the Q-function encodes all necessary information about the reward function. Using the Bellman operator, IPL eliminates the need for an explicit reward model, thereby simplifying the algorithm and improving parameter efficiency. This innovation marks a crucial step towards more efficient and scalable PbRL methods. However, IPL still requires an external preference dataset and remains a value-based method, leaving room for further improvements in data efficiency and algorithmic approach.

2.3. Alignment Techniques in Large Language Models

Reinforcement learning from human feedback (RLHF) (Kaufmann et al., 2023; Wang et al., 2024) has emerged as a powerful approach for aligning large language models (LLMs) with human preferences. Pioneered by InstructGPT (Ouyang et al., 2022) and further developed by Bai et al. (2022), RLHF involves training a reward model based on human preference data and then optimizing the policy using reinforcement learning guided by this reward model. While effective, RLHF is computationally intensive and requires careful hyper-parameter tuning.

To address these challenges, a class of methods known as DPO-like methods has been proposed as simpler alternatives that bypass explicit reward modeling. DPO (Rafailov et al., 2024) directly optimized the policy to match preference data using a classification loss. The standard DPO loss function is defined as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\left(log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right)\right], \tag{1}$$

where π_θ is the agent's policy parameterized by θ, π_{ref} is the reference model, $\sigma(z) = 1/(1+e^{-z})$ is the sigmoid function, β is a scaling factor, $\mathcal{D}$ is the dataset of preference pairs (x, y_w, y_l), and y_w and y_l denote the preferred and less preferred responses given a prompt x, respectively.

Several variants of DPO, collectively referred to as DPO-like methods, have been developed to improve performance or address specific issues. For instance, identity preference optimization (IPO) (Azar et al., 2024) aimed to mitigate overfitting in preference learning, DPO-positive (DPOP) (Pal et al., 2024) introduced additional regularization to prevent reward degradation, and Kahneman-Tversky optimization (KTO) (Ethayarajh et al., 2024) incorporated insights from prospect theory to better model human decision-making.

Recent researches have focused on developing reference-free DPO-like methods to eliminate the need for a fixed reference model. Simple preference optimization (SimPO) (Meng et al., 2024) introduced a loss function with length normalization and a reward margin, enabling reference-free optimization while addressing response length control. CPO (constrastive preference optimization) (Xu et al., 2024) showed that when the reference model perfectly aligns with the true data distribution of preferred data, the DPO loss is upper-bounded by a simpler loss function without a reference model by assuming a uniform distribution U. The preference part of the CPO loss function is given by:

$$\mathcal{L}_{\text{CPO}}^{\textit{Preference}}(\pi_\theta) = \mathcal{L}_{\text{DPO}}(\pi_\theta; U) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\left(\log\pi_\theta(y_w|x) - \log\pi_\theta(y_l|x)\right)\right)\right] \tag{2}$$

These DPO-like methods not only enhance computational and memory efficiency but also retain comparable optimization performance to the standard DPO.

Most closely related to our work is SPIN (self-play fine-tuning) (Chen et al., 2024), which employed a self-play mechanism to improve an LLM without additional human preference data iteratively. SPIN generated its training data and refines itself by distinguishing between current and previous outputs, continuously updating its reference model.

Our research bridges the gap between language model alignment and imitation learning, demonstrating how DPO-like alignment techniques, originally developed for text generation problems in LLMs, can be successfully adapted to control problems in reinforcement learning. This cross-domain application opens up new possibilities for improving imitation learning in complex, real-world tasks, and highlights the potential of adapting DPO-like methods.

3. Preference Optimization for Imitation Learning

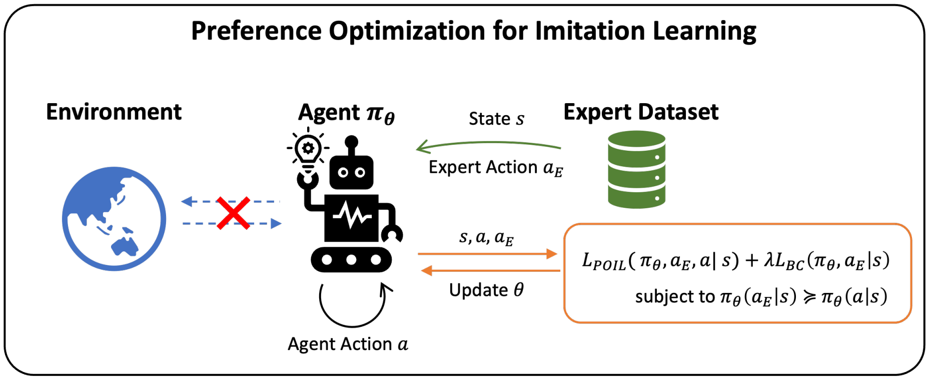
Preference Optimization for Imitation Learning
Environment
Agent π_θ
State s
Expert Dataset
Expert Action a_E
s, a, a_E
Update θ
$L_{POIL}(\pi_\theta, a_E, a| s) + \lambda L_{BC}(\pi_\theta, a_E|s)$
subject to $\pi_\theta(a_E|s) \succcurlyeq \pi_\theta(a|s)$
Agent Action a

Figure 1: Process overview of preference optimization for imitation learning (POIL). The agent, guided by an expert dataset, compares the agent's actions with expert actions and computes the preference loss for updating the policy parameters. This process does not require environmental interaction, as the red cross indicates.

3.1. Adapting DPO-like Methods for Imitation Learning

Our approach leverages the strengths of DPO-like methods and, namely, combines the self-play mechanism from SPIN with the reference-free optimization from CPO, to enhance the imitation learning process. In this method, we eliminate the need for a reference model in the self-play setup by adopting CPO's reference-free loss function and allowing us to apply self-play in offline imitation learning without the computational overhead of maintaining a reference model. Specifically, we focus on iteratively refining the agent's policy by directly comparing its actions to those of experts, thus enabling the model to align more closely with expert behavior while reducing computational complexity.

In SPIN, a model generates synthetic data and refines itself by distinguishing between current and previous outputs using a continuously updated reference model. However, maintaining and updating this reference model adds computational complexity. To address this, we adapt CPO's reference-free optimization, eliminating the need for a reference model while retaining the benefits of self-play.

In our approach, expert demonstrations serve as positive samples ($y_w = a_E$), while the model's own actions during training are treated as negative samples ($y_l = a$), where

a denotes the agent's action and a_E denotes the expert's action. This direct comparison enables the model to prioritize actions that align more closely with expert behavior. By progressively learning from its own generated data in relation to expert demonstrations, the agent refines its policy, achieving a more stable and efficient imitation learning process without relying on predefined rewards or reference models.

3.2. POIL Objective

Our goal is to adapt DPO-like methods to the offline IL setting by eliminating the need for a reference model and incorporating a BC regularization term to enhance performance. Inspired by the findings of CPO as described in Subsection 2.3, we consider the expert's policy as the true data distribution of preferred actions. First, we obtain a reference-free loss function for imitation learning as follows.

$$\mathcal{L}_{\text{POIL}}(\pi_\theta) = -\mathbb{E}_{(s,a_E,a)\sim\mathcal{D}_E}\left[\log\sigma\left(\beta\left(\log\pi_\theta(a_E|s) - \log\pi_\theta(a|s)\right)\right)\right], \tag{3}$$

This loss function is designed to achieve two main objectives:

1. Align with expert behavior. By maximizing $\log\pi_\theta(a_E|s)$, we encourage the agent to assign higher probabilities to the expert's actions.

2. Discourage sub-optimal actions. By minimizing $\log\pi_\theta(a|s)$, we encourage the agent to move away from sub-optimal (agent's) behaviors.

The loss design of POIL (Equation 3) is to minimize the divergence between the expert's behavior and the agent's behavior while maximizing the preference of expert actions over the agent's current actions.

To further encourage the policy to closely mimic expert actions, we incorporate a BC (behavior cloning) regularization term, similar to the approach in Xu et al. (2024). Specifically, we add the negative log-likelihood of expert actions under the agent's policy:

$$\mathcal{L}_{\text{BC}}(\pi_\theta) = -\mathbb{E}_{(s,a_E)\sim\mathcal{D}_E}\left[\log\pi_\theta(a_E|s)\right]. \tag{4}$$

Our overall augmented POIL loss function then combines the preference optimization and the BC regularization:

$$\mathcal{L}^{\text{aug}}_{\text{POIL}}(\pi_\theta) = \mathcal{L}_{\text{POIL}}(\pi_\theta) + \lambda\cdot\mathcal{L}_{\text{BC}}(\pi_\theta), \tag{5}$$

where λ is a hyper-parameter that balances the trade-off between preference optimization and behavior cloning.

By incorporating λ as a tunable parameter, we allow for greater flexibility in balancing the influence of the BC regularization term, which is crucial in scenarios with varying quality or quantity of expert data.

Our approach builds upon the proof provided by Xu et al. (2024), adapting it to the offline imitation learning setting and introducing λ to enhance the method's adaptability. This results in a loss function that effectively guides the policy towards aligning with the expert data distribution without the need for a reference model.

3.3. Algorithm

The POIL algorithm, detailed in Algorithm 1, iteratively refines the agent's policy to better align with expert behavior through preference-based optimization and behavior cloning regularization. As shown in Figure 1, the process begins by initializing the policy parameters randomly. During each iteration, the algorithm samples state-action pairs from the expert demonstrations, which serve as the basis for learning.

Algorithm 1 Preference Optimization for Imitation Learning (POIL)

```
Require: Expert dataset $\mathcal{D}_E = \{(s_i, a_{E,i})\}_{i=1}^N$, scaling factor $\beta$, regularization coefficient
    $\lambda$, batch size $m$, learning rate $\eta$, number of iterations $T$.
Randomly initialize policy parameters $\theta$
for iteration $= 1$ to $T$ do
    Sample a batch of expert state-action pairs $\{(s_j, a_{E,j})\}_{j=1}^m$ from $\mathcal{D}_E$
    Sample agent actions $a_j$ from $\pi_\theta(a|s_j)$ for all $j$
    for each $(s_j, a_{E,j}, a_j)$ in batch do
        Compute the POIL loss: $\mathcal{L}_{\text{POIL}} = -\log\sigma\left(\beta\left(\log\pi_\theta(a_{E,j}|s_j) - \log\pi_\theta(a_j|s_j)\right)\right)$
        Compute the BC regularization term: $\mathcal{L}_{\text{BC}} = -\log\pi_\theta(a_{E,j}|s_j)$
        Combine the losses: $\mathcal{L}_{\text{POIL}}^{\text{aug}} = \mathcal{L}_{\text{POIL}} + \lambda \cdot \mathcal{L}_{\text{BC}}$
        Update policy parameters: $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{\text{POIL}}^{\text{aug}}$
    end for
    return $\theta$
end for
```

In Algorithm 1, $\mathcal{D}_E$ represents the dataset of expert demonstrations, and N is the total number of expert state-action pairs. The regularization coefficient λ controls the weight of the BC regularization relative to the preference-based loss, while the scaling factor β, learning rate η, and number of iterations T are hyper-parameters that control the optimization process. The batch size m determines how many samples are used in each iteration to compute the gradient.

The algorithm proceeds by sampling batches of expert data and generating corresponding agent actions. The total loss $\mathcal{L}_{\text{total}}$ is computed for each sample in the batch, balancing between preference optimization and behavior cloning. The policy parameters θ are then updated using gradient descent to minimize the total loss, thereby improving the agent's policy to better match the expert's behavior.

4. Experiments

In this section, we evaluate the performance of POIL on control tasks in the MuJoCo environment (**?**) and Adroit dexterous manipulation tasks from the D4RL benchmark (Fu et al., 2020). We conduct several experiments to demonstrate the effectiveness of POIL under different settings and compare it against some baseline methods, which include those state-of-the-art in the past.

We compare the performance of POIL against several baseline methods, including BC (Pomerleau, 1991), IQ-Learn (Garg et al., 2021), MCNN (Sridhar et al., 2023) and O-DICE

Environment	Traj.	BC	IQ-Learn	MCNN
HalfCheetah	traj1	2775.39 ± 296.23	3416.38 ± 285.19	4435.17 ± 310.87
	traj2	3031.12 ± 727.32	3938.39 ± 352.01	4186.35 ± 121.60
	traj3	2927.38 ± 967.15	3405.82 ± 681.51	3590.01 ± 537.49
Hopper	traj1	1548.32 ± 108.74	3497.65 ± 80.21	1327.03 ± 230.00
	traj2	1687.93 ± 70.75	**3332.26 ± 186.25**	2783.80 ± 228.24
	traj3	1850.54 ± 870.87	2860.63 ± 498.47	1553.76 ± 409.65
Walker2d	traj1	881.23 ± 68.44	618.00 ± 255.96	1288.02 ± 887.30
	traj2	1023.85 ± 124.71	873.80 ± 64.50	2198.08 ± 741.59
	traj3	935.63 ± 76.50	924.57 ± 230.85	1127.44 ± 405.91

Environment	Traj.	O-DICE	POIL$_{\lambda=1}$	POIL$_{\lambda=0}$
HalfCheetah	traj1	1532.63 ± 716.74	3843.75 ± 379.47	**4628.88 ± 183.47**
	traj2	1801.12 ± 697.67	2228.04 ± 797.59	**4833.13 ± 30.95**
	traj3	1032.74 ± 1104.10	2645.44 ± 876.50	**4309.46 ± 247.95**
Hopper	traj1	1672.45 ± 441.29	2426.22 ± 622.98	**3501.67 ± 469.42**
	traj2	1748.63 ± 204.60	2534.35 ± 398.42	3066.00 ± 158.89
	traj3	1797.43 ± 145.06	3372.09 ± 52.41	**3499.08 ± 58.08**
Walker2d	traj1	1543.67 ± 445.67	2645.29 ± 840.42	**4722.79 ± 521.90**
	traj2	1392.54 ± 1106.87	1826.08 ± 490.14	**5233.93 ± 234.21**
	traj3	1483.52 ± 716.27	2493.53 ± 729.79	**3745.50 ± 445.90**

Table 1: performance of various methods trained with a single expert demonstration on MuJoCo tasks. The results are averaged over 3 different runs, each using a unique random seed, and the scores represent the average over the last ten epochs. (The bold numbers represent the best, while the underscored numbers are the second best. Note that the scores are not normalized to expert data because we cannot directly get the expert scores from this dataset (Kostrikov et al., 2019)).

Task	BC	IQ-Learn	ODICE	MCNN	POIL$_{\lambda=1}$	POIL$_{\lambda=0}$
Pen	2633	662.81 ± 881.71	2712.92 ± 200.53	3405 ± 328	3036.04 ± 180.60	**4077.04 ± 66.77**
Hammer	16140	18.97 ± 96.21	10921.48 ± 829.98	**16387 ± 682**	15904.91 ± 173.60	16295.94 ± 49.67
Door	969	1950.80 ± 730.36	3003.24 ± 61.04	3035 ± 7	2438.69 ± 576.73	**3040.88 ± 12.33**
Relocate	4289	32.86 ± 49.95	4429.50 ± 29.50	4566 ± 47	4090.77 ± 157.97	**4606.51 ± 45.64**

Table 2: Performance comparison on Adroit tasks using the full expert dataset (5,000 demonstrations). The results are averaged over three runs with different random seeds. (The bold numbers indicate the best performance, while the underscored numbers are the second best.)

(Mao et al., 2024), many of which were state-of-the-art in offline imitation learning in the past. For fair comparison, we use the same neural network architecture for all methods. The policy network consists of two fully connected layers, each with 256 units, with ReLU activation functions applied after each layer. All models are trained for 100k timesteps. We use the Adam optimizer (**?**) for optimization with default parameters. The experiments are conducted on a system equipped with 4 NVIDIA RTX A6000 GPUs, 128GB of RAM, and an AMD Threadripper PRO 5965WX processor featuring 24 cores and 48 threads.

4.1. Single Demonstration Learning

We conduct experiments on standard MuJoCo control tasks, specifically `HalfCheetah-v2`, `Hopper-v2`, and `Walker2d-v2`. These tasks are widely used as benchmarks in reinforcement learning, requiring agents to learn complex locomotion behaviors in high-dimensional state and action spaces. For the single demonstration experiments, we utilize one expert trajectory per task, sourced from the same dataset as used in ValueDICE (Kostrikov et al., 2019). These expert trajectories generated by well-trained policies present a challenging setting for imitation learning due to the limited data available.

In this experiment, we evaluate the ability of POIL to learn effective policies from a single expert trajectory. The hyper-parameter β is set to 0.2 in this experiment. This experiment tests the data efficiency of imitation learning methods when only minimal expert data is available.

As shown in Table 1, POIL with $\lambda = 0$ achieves the best performance on eight out of nine trajectories across all tasks, only falling behind IQ-Learn on trajectory 2 of the `Hopper-v2` task. This demonstrates POIL's ability to effectively utilize limited expert data and suggests superior data efficiency compared to other methods. More discussion about λ is given in Subsection 4.3.2.

4.2. Adroit Dexterous Manipulation

In this experiment, we focused on the expert datasets from D4RL within the Adroit tasks, following the same settings as in MCNN (Sridhar et al., 2023) to ensure consistency and allow for a direct comparison between POIL and other methods under similar conditions. Adroit tasks have been considered challenging benchmarks in the field of robotics and reinforcement learning, involving dexterous manipulation with high-dimensional human-like five-finger hands. These tasks simulate complex real-world scenarios, requiring precise control and coordination of multiple joints. The complexity of these tasks makes them particularly demanding for imitation learning algorithms, providing a rigorous test of an algorithm's ability to learn and replicate sophisticated motor skills.

Specifically, the experiment is conducted for the following tasks: `pen-expert-v1`, `hammer-expert-v1`, `door-expert-v1`, and `relocate-expert-v1`. For our POIL, we empirically set the the scaling factor $\beta = 1$. For baseline methods, we obtained the results of O-DICE and IQ-Learn by running their official implementation, while the results of other methods are directly inherited from MCNN. Namely, the one for MCNN is the best one with tuned hyperparameters, denoted by MCNN+MLP in Sridhar et al. (2023). As shown in Table 2, POIL outperforms all methods for all cases except for the case `hammer-expert-v1` for MCNN, where POIL with $\lambda = 0$ achieves comparable performance with only a slight difference.

4.3. Ablation Study

4.3.1. Component Analysis of POIL

Recent theoretical analysis (Swamy et al., 2025) has provided important insights into the relationship between POIL and behavioral cloning. Specifically, Theorem 2.4 in that work demonstrates that under idealized conditions—infinite sampling and linear loss function—POIL should converge to the behavioral cloning gradient. The same work also argues from an information-theoretic perspective (Section 3.1) that on-policy data is redundant via the data-processing inequality, questioning the value of stochastic sampling in preference optimization. However, our empirical results consistently show POIL outperforming BC across all experimental settings, motivating a systematic investigation into which specific components drive this performance advantage.

To understand the contribution of each component in POIL, we conduct a controlled ablation study examining two key design choices: the sigmoid nonlinearity $\log \sigma(\cdot)$ in the loss function and the stochastic sampling strategy for generating negative examples. We evaluate variants of POIL$_{\lambda=0}$ on the Adroit manipulation tasks, where $\lambda = 0$ is selected based on our findings in Subsection 4.3.2.

We test the complete POIL method, a linear version without sigmoid nonlinearity (”POIL w/o sigmoid”), a variant using deterministic policy mean $\mu_\theta(s)$ instead of stochastic sampling (”POIL w/o stochastic”), and a version combining both modifications (”POIL w/o both”). The linear variant follows the form:

$$\mathcal{L}_{\text{linear}}(\pi_\theta) = -\mathbb{E}_{(s,a_E,a)\sim\mathcal{D}_E}\left[\beta\left(\log \pi_\theta(a_E|s) - \log \pi_\theta(a|s)\right)\right] \tag{6}$$

Task	BC	POIL	POIL w/o sigmoid	POIL w/o stochastic	POIL w/o both
Pen	2633	$\mathbf{4077.04 \pm 66.77}$	3149.52 ± 32.37	3091.31 ± 44.35	253.26 ± 2.30
Hammer	16140	$\mathbf{16295.94 \pm 49.67}$	15402.48 ± 455.17	15486.26 ± 199.52	-238.37 ± 20.21
Door	969	$\mathbf{3040.88 \pm 12.33}$	2265.42 ± 435.28	1653.31 ± 476.53	-52.73 ± 3.04
Relocate	4299	$\mathbf{4606.51 \pm 45.64}$	4377.92 ± 15.88	4374.64 ± 21.17	-4.78 ± 1.86

Table 3: Component analysis of POIL$_{\lambda=0}$ variants compared with BC baseline on Adroit tasks. Results are averaged over three runs with different random seeds.(The bold numbers indicate the best performance.)

Table 3 presents the experimental results compared with the BC baseline. The original POIL substantially outperforms BC across all tasks. Individual component removals (”POIL w/o sigmoid” and ”POIL w/o stochastic”) show comparable performance to BC in most cases, while the combined removal (”POIL w/o both”) leads to catastrophic failure with performance dramatically worse than BC.

These results illuminate the gap between theoretical predictions and practical performance in continuous control settings. While theory suggests linear POIL should converge to BC and that self-sampling provides no additional information, our experiments demonstrate that both components are essential for maintaining POIL's performance advantages. The catastrophic failure when both are removed reveals that POIL's effectiveness stems from synergistic component interactions that emerge under practical constraints rather than the idealized conditions assumed in theoretical analysis.

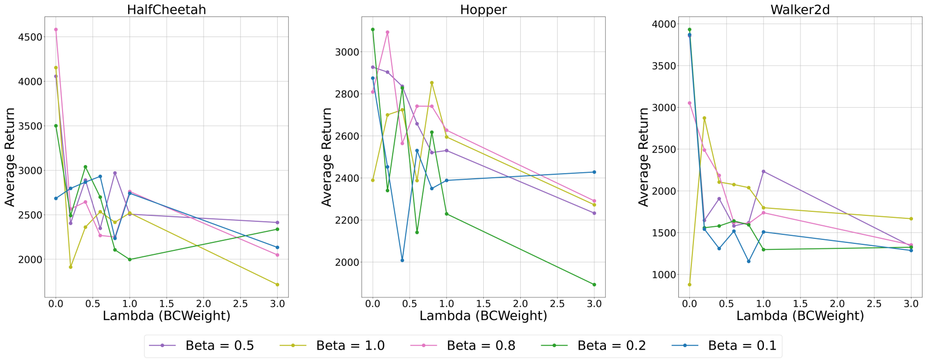

Figure 2: Performance comparison between different λ and β values on the `HalfCheetah-v2`, `Hopper-v2`, and `Walker2d-v2` tasks.

4.3.2. Impact of Hyperparameters β and λ

We conduct a comprehensive ablation study to analyze the impact of both the scaling factor β and the BC coefficient λ on POIL's performance. These experiments were conducted using only a single expert demonstration for each of the three environments: `HalfCheetah-v2`, `Hopper-v2`, and `Walker2d-v2`, with results averaged over three different random seeds to ensure robust evaluation.

Figure 2 illustrates the performance of POIL across the three tasks as λ varies from 0 to 3, specifically $\lambda = [0, 0.2, 0.4, 0.6, 0.8, 1.0, 3.0]$, for five different $\beta = [0.1, 0.2, 0.5, 0.8, 1.0]$. Our findings reveal several important insights about both hyperparameters.

Impact of λ (BC Coefficient): Our results consistently show that smaller values of λ lead to higher performance across all three tasks for most β values. In particular, $\lambda = 0$ yields the highest returns for most β configurations in all three environments. As λ increases, performance decreases notably, especially for larger values such as 3.0, where performance deteriorates significantly across the board. This suggests that the preference-based loss alone is sufficient for effective policy learning from demonstrations, and the additional BC regularization term may unnecessarily constrain the policy optimization process.

Impact of β (Scaling Factor): The scaling factor β also demonstrates significant influence on performance. Smaller values of β, particularly $\beta = 0.2$, consistently yield better performance across most tasks and λ settings. This suggests that choosing an appropriate scaling factor effectively controls the sharpness of the preference function in the loss, influencing how strongly the model distinguishes between expert and agent actions. A smaller β value tends to smooth the preference function, leading to more stable gradients and improved training dynamics.

The interaction between β and λ reveals that the combination of $\lambda = 0$ and $\beta = 0.2$ provides the most consistent performance improvements across all evaluated environments. While $\beta = 0.2$ performed well in most cases, different environments may require different scaling factors based on their complexity and the nature of the expert data. The consistent observation that smaller values for both hyperparameters yield better performance suggests that a smoother preference function with minimal BC regularization is generally beneficial in offline imitation learning with POIL.

4.3.3. Comparison Between Varying Numbers Of Demonstrations

Task	**Demos**	**MCNN+MLP**	**POIL**
Door	100	2725 ± 139	**3015 ± 3**
	500	2931 ± 32	**3025 ± 1**
	1000	2992 ± 18	**3027 ± 7**
	2000	3017 ± 10	**3028 ± 22**
	4000	3025 ± 3	**3033 ± 3**
	5000	3035 ± 7	**3041 ± 12**
Pen	100	—	**4146 ± 104**
	500	3712 ± 32	**4127 ± 8**
	1000	3808 ± 6	**4141 ± 38**
	2000	3858 ± 29	**4197 ± 136**
	4000	3934 ± 42	**4172 ± 113**
	5000	4051 ± 195	**4078 ± 66**

Table 4: Performance comparison on Adroit tasks with varying numbers of demonstrations. Results are averaged over three runs with different random seeds. (The bold numbers indicate the best performance for each dataset size)

To assess POIL's performance across varying dataset sizes, we conducted experiments on datasets of different scales. The experimental setup follows that of MCNN, with data directly sourced from the MCNN paper to ensure result comparability. Table 4 demonstrates that POIL consistently achieves superior performance across all tasks and dataset sizes. Notably, POIL shows strong performance even with a limited number of demonstrations (e.g., 100 demos), highlighting its data efficiency and robustness in data-scarce scenarios.

5. Discussion

In this paper, we introduce POIL, a novel method inspired by preference optimization techniques from large language model alignment. POIL eliminates the need for adversarial training by directly comparing agent actions to expert actions. Through extensive experiments on MuJoCo tasks, and Adroit manipulation tasks, we demonstrate that POIL performs best or competitively against state-of-the-art methods, particularly in data-scarce settings.

Our study reveals interesting insights about the regularization coefficient λ. While it plays a crucial role in CPO, we found that in our imitation learning context, setting $\lambda = 0$ often leads to better performance across various tasks and dataset sizes. This suggests that POIL can effectively learn from expert demonstrations without additional regularization, highlighting the method's robustness in capturing expert behavior.

Notably, our experiments on the Adroit manipulation tasks showcase POIL's exceptional performance in complex, high-dimensional control problems. These tasks involve dexterous manipulation of objects, presenting a significant challenge in robotics and control. POIL consistently outperformed state-of-the-art methods in these tasks, demonstrating its ability to handle intricate action spaces and learn sophisticated behaviors.

Overall, POIL offers a robust solution for offline imitation learning, especially when expert data is limited or challenging to learn from. Its flexibility in adapting to different dataset sizes and task difficulties makes it a promising direction for future research in imitation learning and related fields.

Acknowledgments

This work was partially supported by the National Science and Technology Council (NSTC) of Taiwan under contract NSTC 114-2221-E-A49-005, 114-2221-E-A49-006, 114-2622-E-A49-017.

References

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

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

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.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602 (7897):414–419, 2022.

Jonas Eschmann. Reward function design in reinforcement learning. *Reinforcement Learning Algorithms: Analysis and Applications*, pages 25–33, 2021.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34:4028–4039, 2021.

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

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. Strictly batch imitation learning by energy-based distribution matching. *Advances in Neural Information Processing Systems*, 33:7354–7365, 2020.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022.

Bahare Kiumarsi, Kyriakos G Vamvoudakis, Hamidreza Modares, and Frank L Lewis. Optimal and autonomous control using reinforcement learning: A survey. *IEEE transactions on neural networks and learning systems*, 29(6):2042–2062, 2017.

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

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*, 2018.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. *arXiv preprint arXiv:1912.05032*, 2019.

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*, 2021.

Yicheng Luo, Zhengyao Jiang, Samuel Cohen, Edward Grefenstette, and Marc Peter Deisenroth. Optimal transport for offline imitation learning. *arXiv preprint arXiv:2303.13971*, 2023.

Ma X. Wan L. Liu R. Li X. Lu Z. Lyu, J. Seabo: A simple search-based method for offline imitation learning. *ArXiv Preprint ArXiv:2402.03807*, 2024.

Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pages 14639–14663. PMLR, 2022.

Liyuan Mao, Haoran Xu, Weinan Zhang, and Xianyuan Zhan. Odice: Revealing the mystery of distribution correction estimation via orthogonal-gradient update. *arXiv preprint arXiv:2402.00348*, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

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.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

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.

Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al.

Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, James Weimer, and Insup Lee. Memory-consistent neural networks for imitation learning. *arXiv preprint arXiv:2310.06171*, 2023.

Mingfei Sun, Anuj Mahajan, Katja Hofmann, and Shimon Whiteson. Softdice for imitation learning: Rethinking off-policy distribution matching. *arXiv preprint arXiv:2106.03155*, 2021.

Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

Gokul Swamy, Sanjiban Choudhury, Wen Sun, Zhiwei Steven Wu, and J. Andrew Bagnell. All roads lead to likelihood: The value of reinforcement learning in fine-tuning, 2025. URL https://arxiv.org/abs/2503.01067.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*, 2024.

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

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.

Sheng Yue, Guanbo Wang, Wei Shao, Zhaofeng Zhang, Sen Lin, Ju Ren, and Junshan Zhang. Clare: Conservative model-based reward learning for offline inverse reinforcement learning. *arXiv preprint arXiv:2302.04782*, 2023.

Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2024.

Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyu Wang, Yusuf Aytar, Misha Denil, Nando de Freitas, and Scott Reed. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020.