DAC-Augmented AIRL

Li-Cheng Hsu
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
b11901158@ntu.edu.tw

Ming-Yang Wu
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
b09901045@ntu.edu.tw

Jun-Wei Ding
Department of Civil Engineering
National Taiwan University
Taipei, Taiwan
d135210232ntu.edu.tw

Yi-Lun Lin
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
b12901062@ntu.edu.tw

Abstract

Imitation Learning sometimes faces challenges in sample efficiency and domain shift, limiting its real-world applicability. To address these, we propose DAC-AIRL, a framework combining the sample efficiency of Discriminator Actor-Critic (DAC) with the robust reward modeling of Adversarial Inverse Reinforcement Learning (AIRL). DAC efficiently augments data in resource-constrained scenarios, while AIRL uses this data to infer adaptive policies resilient to domain shifts.

Using BipedalWalker-v3 from Open AI Gym, our experiments validate DAC's efficiency, AIRL's adaptability, and their combined performance. Results show that DAC-AIRL reduces data requirements while ensuring policy generalization, advancing RL for dynamic, data-scarce environments.

1 Introduction

Reinforcement Learning (RL) has established itself as a versatile framework for solving complex sequential decision-making problems across diverse domains, including robotics, game playing, and autonomous systems. However, despite its remarkable progress, two persistent challenges significantly limit the deployment of RL in real-world scenarios: sample efficiency and domain shift.

Sample efficiency is a critical factor in RL, as real-world environments often impose constraints on the amount of interaction data available for training. In high-stakes applications such as robotic manipulation and healthcare systems, where data collection can be costly or time-consuming, high data requirements render many RL approaches impractical. To address this limitation, various methods have been proposed. Model-based RL, for instance, leverages learned environment dynamics to reduce the need for extensive trial-and-error interactions. Off-policy algorithms such as Soft Actor-Critic (SAC) [1] and Deep Q-Networks (DQN) [2] improve data reuse through experience replay buffers, enhancing learning efficiency . Furthermore, meta-learning and few-shot learning have emerged as promising paradigms to enable RL agents to adapt to novel tasks with minimal data.

On the other hand, Domain shift arises when an RL agent trained in a source domain encounters discrepancies in the target domain, such as variations in system dynamics, sensor configurations, or operational conditions. This challenge is prevalent in real-world deployments, where the conditions during training and deployment are often mismatched. Addressing domain shift has been the focus of extensive research, with techniques such as domain adaptation, domain randomization, and transfer learning demonstrating efficacy. For example, adversarial training methods align feature distributions

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

between source and target domains, while policy adaptation strategies fine-tune pretrained policies with limited additional data in the target domain.

Although prior work has tackled sample efficiency and domain shift independently, the interplay between these two challenges remains largely underexplored. In practical settings, RL agents are often required to operate under data-scarce conditions while maintaining robustness and adaptability to varying and uncertain environments. Bridging this gap necessitates an integrated framework capable of simultaneously addressing sample efficiency and domain adaptability. In response to these limitations, this work proposes the development of DAC-AIRL, a novel algorithm that integrates the sample efficiency of Discriminator-Actor-Critic (DAC) [3] and the domain adaptability of Adversarial Inverse Reinforcement Learning (AIRL) [4]. Specifically, DAC-AIRL leverages DAC's efficient data utilization to address challenges in data-scarce environments and AIRL's robust domain modeling capabilities to handle domain shift.

To further enhance the applicability of DAC-AIRL in real-world scenarios, we unify these two methodologies into a cohesive framework that simultaneously improves sample efficiency and domain adaptability. This integration allows RL agents to learn effective policies with minimal data while maintaining robust generalization across diverse and dynamic deployment environments.

In summary, our contributions in the realm of RL algorithm development for resource-constrained and domain-shifting settings include:

- Proposing a DAC-based module for enhanced data efficiency, enabling RL agents to operate effectively in data-limited scenarios.
- Introducing an AIRL-based domain adaptation mechanism to ensure robustness and adaptability across varying environmental conditions.
- Developing a unified framework that combines these complementary strengths, setting a new benchmark for RL performance in dynamic and resource-constrained environments.

Through these contributions, DAC-AIRL addresses the dual challenges of sample efficiency and domain shift, paving the way for more robust and scalable RL frameworks in real-world applications.

2 Related Work

2.1 Techniques for Improving Sample Efficiency in RL

Sample efficiency is a critical bottleneck in reinforcement learning (RL), especially for applications that involve high-stakes decision-making or require extensive real-world interaction. Among the most prominent approaches, **latent-space learning** [5] has been explored to reduce the dimensionality of the state-action space, enabling more efficient exploration and policy learning. By compressing high-dimensional data into a lower-dimensional latent representation, these methods allow agents to focus on meaningful features for decision-making.

Another area of focus has been **reward shaping** [6], where additional reward signals or auxiliary tasks are introduced to accelerate learning. For example, curiosity-driven exploration methods encourage agents to seek novel states, significantly improving sample efficiency in sparse-reward environments. Similarly, self-supervised learning has been incorporated to pretrain models with unlabeled interaction data, which can then be fine-tuned for downstream RL tasks. Additionally, imitation learning has shown promise for improving sample efficiency by bootstrapping RL policies with expert demonstrations. Recent advancements include combining imitation learning with reinforcement learning in frameworks such as GAIL (Generative Adversarial Imitation Learning) [7] and DAC (Discriminator-Actor-Critic) [3], which enable faster convergence while maintaining the ability to explore beyond demonstrated behaviors.

2.2 Domain Shift Mitigation Strategies

Addressing domain shift is crucial for deploying RL agents in real-world applications, where training and deployment conditions often differ. One promising avenue is **representation learning**, which aims to extract invariant features that are robust to domain discrepancies. Techniques such as domain-invariant feature learning or contrastive learning have shown effectiveness in aligning feature spaces

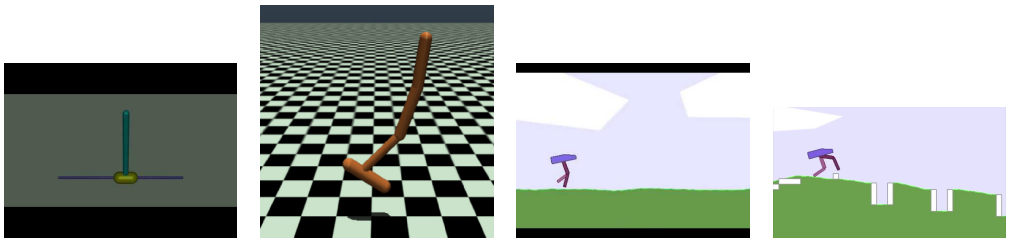
between source and target domains. Adversarial Inverse Reinforcement Learning (AIRL) [4] has also shown significant promise in addressing domain shift by learning robust reward functions that generalize across domains. Unlike traditional RL methods that rely on handcrafted rewards, AIRL employs adversarial training to infer underlying reward structures from expert demonstrations. This approach enables agents to adapt effectively to target domains by capturing task-specific objectives while mitigating the influence of domain-specific artifacts. Recent advancements in AIRL have demonstrated its ability to enhance policy transfer between simulation and real-world settings, making it a powerful tool for domain adaptation in RL. Recent studies [8] have also explored **meta-learning** for domain shift, where agents are trained to quickly adapt to new domains with minimal data. This has been particularly impactful in few-shot transfer scenarios, where the target domain offers limited interaction opportunities.

3 Environment

3.1 The Choice of Environment – Gym

Gym is a widely used platform in the context of reinforcement learning research. Developed by OpenAI, Gym provides a standardized interface for environments, which simplifies the process of benchmarking and developing reinforcement learning algorithms. Its popularity arises from its diverse collection of environments ranging from simple tasks like Inverted Pendulum to more complex simulations like BipedalWalker. The environments support both discrete and continuous action spaces, catering to a variety of RL tasks.

The consistent and modular design of Gym makes it an ideal choice for both academic and applied research. In our research, we mainly use mujoco environment (Hopper, InvertedPendulum) and Box2D environment.



(a) InvertedPendulum (b) Hopper (c) Bipedal (normal) (d) Bipedal (hardcore)

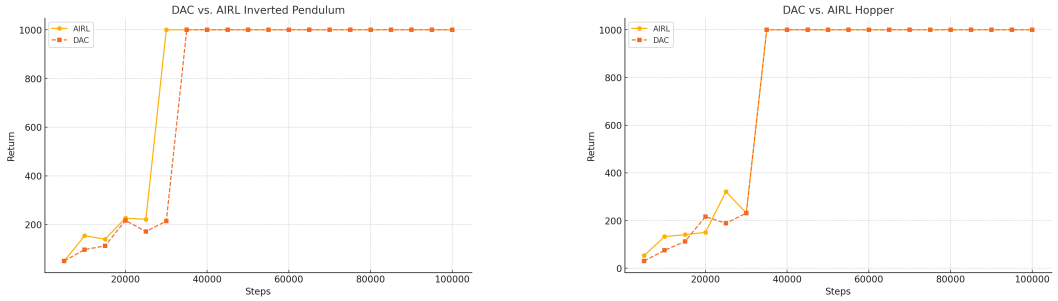
Figure 1: Gym environments used in this research.

3.2 InvertedPendulum-v2 and Hopper-v3

Both the InvertedPendulum-v2 (Figure 1a) and Hopper-v3 (Figure 9a) environments are relatively simple, making them unsuitable for distinguishing between high-performing algorithms such as DAC (Discriminator Actor-Critic) and AIRL (Adversarial Inverse Reinforcement Learning). For instance, in these environments, both algorithms achieve near-perfect performance (see Figure 2), akin to evaluating the abilities of two students when both score 100 on an easy test. While these tasks are useful for validating basic functionality, they fail to provide meaningful insights into the comparative strengths and weaknesses of advanced algorithms.

3.3 BipedalWalker-v3

The BipedalWalker-v3 environment offers a more balanced challenge for evaluating reinforcement learning algorithms. It includes both a normal (Figure 1c) and a hardcore (Figure 1d) version, enabling researchers to assess domain generalization and the ability to handle varying difficulty levels. This characteristic makes BipedalWalker-v3 particularly valuable for studying domain shift, where an agent trained on one version of the environment is tested on another. This nuanced difficulty allows



(a) DAC vs. AIRL Inverted Pendulum

(b) DAC vs. AIRL Hopper

Figure 2: As the result shows, both policy perform almost the same in these two environment. It is hard to judge which one is better under such condition. (expert data size : 50000)

for a more comprehensive evaluation of algorithmic performance, making it a preferred choice in reinforcement learning research. For those reason, we choose BipedalWalker-v3 as our main testing environment.

The reward function of the BipedalWalker-v3 is based on the distance the robot reaches without falling. The maximum reward is about 300. Once the robot falls, it would receive a -100 penalty and the episode would be terminate immediately. This is also why the result figure in the BipedalWalker-v3 experiment is sometime so turbulent since if the robot accidentally falls, even the policy is not that bad, the result would be bad as well.

4 Problem Formulation

As discussed in the introduction, DAC demonstrates strong sample efficiency, while AIRL excels in learning robust reward functions and adaptive policies. However, AIRL's reliance on extensive expert data due to its lack of sample efficiency and DAC's inability to produce adaptive policies limit their standalone applicability in real-world scenarios. To address these shortcomings, our goal is to design a unified framework, DAC-AIRL, that combines the strengths of both algorithms while mitigating their weaknesses.

Our experiments are structured into three parts. First, we validate the sample efficiency of DAC to confirm its capability in data-scarce settings. Second, we assess the adaptability of AIRL by evaluating its performance in handling domain shifts using HardcoreBipedalWalker-v3, a challenging environment designed to test generalization. Finally, we evaluate how DAC-augmented data enhances AIRL's performance, focusing on the interplay between the two methods. By analyzing the results and learning curves across these experiments, we aim to demonstrate the effectiveness and adaptability of the proposed DAC-AIRL framework.

5 Method

We aim to create a workflow (see Figure 3) that can produce robust policy, which is robust to variations of environment, via small amount of data. Our concept is to augment the data with an algorithm that is capable to train with small amount of data but can't produce a robust policy. After Augmentation, feed the augmented buffer to an algorithm that produce better policy but require sufficient expert data. In the end, we can get a robust policy, with small amount of expert data.

5.1 Data Augmentation Phase —- DAC

The Discriminator-Actor-Critic (DAC) algorithm is well-suited for augmenting expert data due to its design and output characteristics. Its structure overview is shown in Figure **??**. Below are the key reasons:

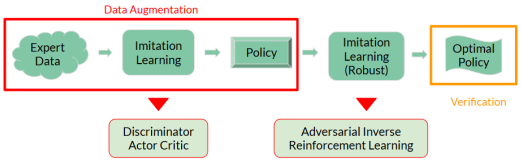


Figure 3: Workflow visualization

- **Discriminator as a Reward Signal**: The discriminator in DAC provides a reward signal by distinguishing between expert and policy-generated data. This signal evaluates the quality of trajectories, making it valuable for augmenting datasets with structured feedback.

- **Off-Policy Learning Efficiency**: The off-policy nature of DAC allows efficient reuse of generated data, enabling the creation of diverse and robust trajectories without excessive environment interactions.

- **Alignment with Downstream Algorithms**: DAC-augmented data aligns well with expert intent, making it suitable for downstream tasks such as:
  - Supervised learning models, which benefit from increased data diversity and quality.
  - Reinforcement learning algorithms, which can use the augmented data to initialize or warm-start training.

In summary, DAC enhances expert datasets by generating high-quality, diverse trajectories that align with expert behaviors, making them useful for downstream learning processes.

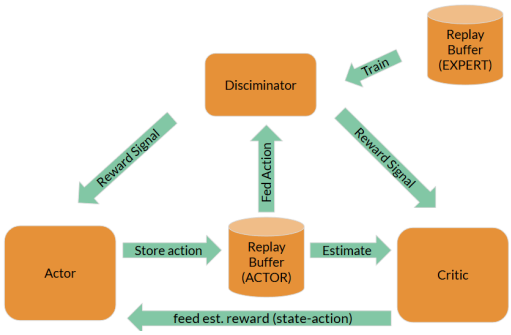


Figure 4: The structure of the DAC algorithm

5.2 Policy Phase —- AIRL

Adversarial Inverse Reinforcement Learning (AIRL) is an excellent downstream choice for processing DAC-augmented data due to its robust reward inference and policy generalization capabilities. Its structure overview is shown in Figure 5. Below are the key reasons:

- **Ability to Infer Underlying Reward Functions**: AIRL infers a reward function from data, which is particularly valuable when receiving DAC-augmented trajectories. These trajectories, generated to mimic expert behavior, provide a rich dataset for AIRL to extract meaningful and generalizable reward signals. This is what the DAC algorithm can't do.

- **Generalization and Robustness to Distribution Shifts**: AIRL leverages the high-quality DAC-augmented data to infer reward functions that generalize well beyond the original training scenarios. Its reward representation is invariant to changes in state distribution, ensuring that the augmented data does not introduce bias or instability, making it ideal for downstream applications.

- **Synergistic Combination with DAC**:
  - DAC enhances the dataset by generating expert-like, high-quality trajectories.
  - AIRL utilizes these trajectories to infer rewards and train policies that are robust, scalable, and generalizable.

  This combination allows for efficient transfer of expert behavior to downstream tasks with minimal additional expert data.

In summary, AIRL is highly suitable as a downstream algorithm for DAC-augmented data because it leverages the improved trajectories to infer robust reward functions, generalize policies across tasks, and mitigate biases introduced during data augmentation.

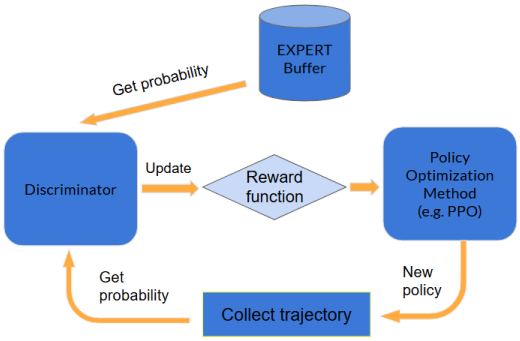


Figure 5: The structure of the AIRL algorithm

5.3 DAC-regulated AIRL

Furthermore, we try to integrate DAC into AIRL framework by using hyperparameter alpha to adjust the proportion of loss function from DAC and AIRL. Despite the fact that our entire structure is based on AIRL regulated by DAC, it still can be seen as a parallel combination of both methods since both methods share many similar elements. This concept is a reference to BC-regulated GAIL from Rohit et al.[4]. The formula on which we conduct policy gradient:

$$Loss = \alpha \times loss_{DAC} + (1 - \alpha) \times loss_{AIRL} \tag{1}$$

5.4 Verification Phase

Once the robust policy is generated by this DAC-AIRL workflow. We will test and compare it with the policy generated directly by AIRL in the BipedalWalker-v3 together. Aim to show that our workflow really enhance the capability of imitation learning.

This structured approach demonstrates how reinforcement learning can leverage both expert data and advanced techniques to generate robust and generalizable policies effectively.

6 Experiments

6.1 Experiment design

The initial input of expert data is retrieved using Soft-Actor-Critic(SAC) to train an agent in the normal BipedalWalker-v3 environment until convergence. We first demonstrate AIRL and DAC's difference in sample efficiency and adaptability to domain shift. Then we verify the effectiveness of DAC-augmented AIRL framework under domain shift and normal environment.

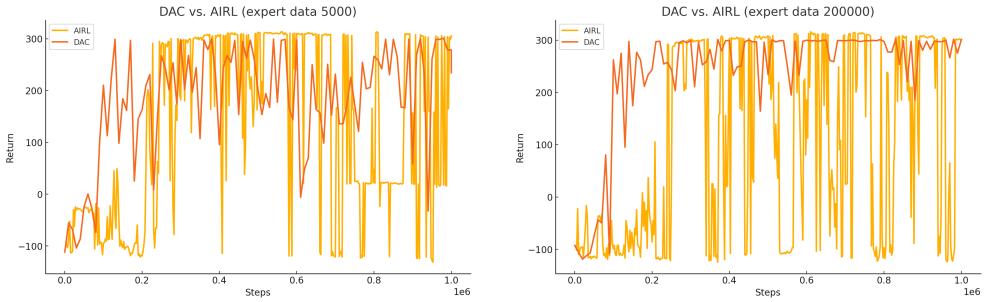
- **Sample Efficiency**: We see how quickly the policy will attain optimality using DAC and AIRL given the same expert data.

- **Domain Shift**: We want to see if the policy AIRL learns in training environment can generalize to drastically different environment. We compare the learning curves and the average return after 600k steps between AIRL and DAC.

By the way, we define the data size as the number of steps contained within the given policy.

6.2 Sample Efficiency

Using same expert data set, DAC obviously learns faster than AIRL in Figure 6. DAC's total reward reach 300(highest value) with fewer steps, and thus we can verify that DAC have higher sample-efficiency.

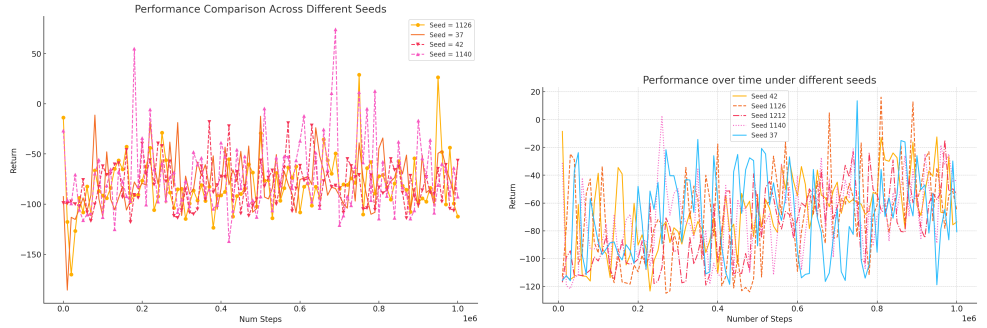


(a) Expert data size 5k (b) Expert data size 20k

Figure 6: Comparison of DAC (red line) and AIRL (orange line).

6.3 Domain Shift

We use train in normal BipedalWalker-v3 and test in Hardcore mode to evaluate the performance under domain shift. The environment can be randomly generated (within the constraint of hardcore/normal mode) using different seeds. AIRL shows to learn steadily in the hardcore environment (Figure 9b) while DAC struggles to do so (Figure 9b). And AIRL ends up doing better than DAC in terms of average return in various hardcore environment. We plot the average return after 600k steps in Table 2.



(a) DAC (b) AIRL

Figure 7: AIRL and DAC under domain shift

6.4 AIRL via Augmentation

We've demonstrated that DAC has sample efficiency and AIRL is adaptable to domain shift. Now we want to show that the sequential combination of both methods has better performance in normal and hardcore environment. We vary the scale of data augmentation and compare to AIRL-only method.

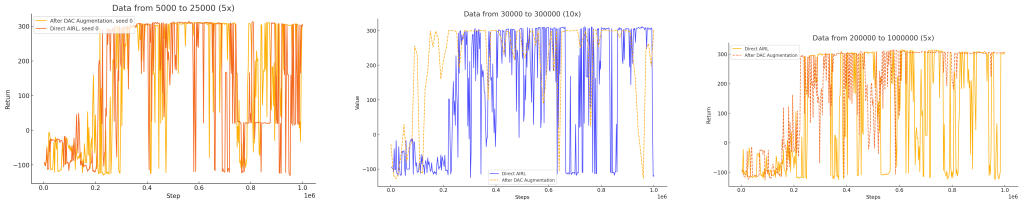
Table 1: Performance comparison (Average return after 600k steps) between DAC and AIRL

Seeds	DAC	AIRL
42	-75.76	-50.87
1126	-75.31	-56.68
1140	-80.60	-57.26
37	-63.78	-58.47
Average	**-77.2**	**-56.6**

6.4.1 Normal environment (no domain shift)

For data augmentation, we discuss three cases of augmentation scaling:

- **Small to small** : Find that what would happens if we just augment data a little.
  - Ex : Expert size from 5k -> 25k
- **Small to large** : Our main section, which is the problem we aim to solve.
  - Ex : Expert size from 30k -> 300k
- **Large to even larger** : Seeing if giving the system even more data would make the policy even better.
  - Ex : Expert size from 200k -> 1M



(a) 5k -> 25k (b) 30k -> 300k (c) 200k -> 1M

Figure 8: Comparison of performance before and after augmentation (no domain shift)

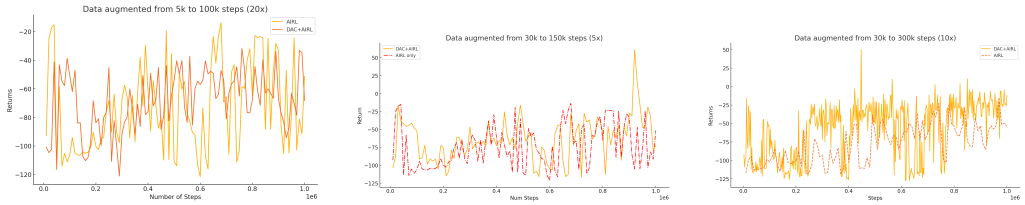
From Figure 8, it is not hard to find that regardless of the augmentation scale, the algorithm run with augmented algorithm always converges faster and reaches a stable return with few fluctuations. However, only 30k -> 300k (see Figure 9b) show significant improvement. After repeatedly run the experiment with different buffer size, we find that BipedalWalker-v3 achieve almost-the-same best policy after the buffer size reaches 200k 300k. We can conclude the observation for our method implemented in BipedalWalker-v3:

- Our augmentation method indeed is effective for all scale. However, the effectiveness depends on the scale of augmentation
- Augmentation work best for the augmentation from buffer size small to large
- It won't optimize the training too much if the original buffer already reaches size around 200k-300k

6.4.2 Hardcore Environment (domain shift)

For the sake of fairness, the above experiments for AIRL-only were run on the same condition with DAC-AIRL including input expert data size. Our experiments show that DAC-AIRL constantly outperform AIRL-only methods regardless of data augmentation approaches. We can conclude the following points based on Figure 9 and Table 2

- DAC-AIRL gain huge advantages over AIRL-only method through data augmentation, as AIRL-only method isn't as sensitive to expert data size as DAC-AIRL



(a) 5k -> 10k (b) 30k -> 150k (c) 30k -> 300k

Figure 9: Comparison of performance before and after augmentation (domain shift)

Table 2: Performance comparison (Average return after 600k steps) between DAC-AIRL and AIRL-only

Data augmentation	DAC-AIRL	AIRL only
5k-100k	-59.84	-62.07
30k-150k	-52.74	-60.58
30k-300k	-45.14	-57.26
Average	**-52.5**	**-60.0**

- Unlike normal environment, the data size after augmentation has more effect on performance than scaling of augmentation under domain shift.
- DAC-AIRL's return is generally more stable than AIRL-only method during learning.

6.5 DAC-regulated AIRL

We also test DAC-regulated AIRL to find out its adaptability to domain shift. Surprisingly, though DAC-regulated AIRL has a huge fluctuation at the start of the learning process, its return converges to somewhere near -40 (Figure 10), whose policy achieves better and more stable return than prior methods (sequentially running DAC and AIRL).

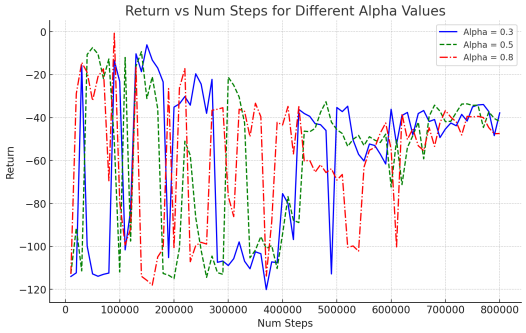


Figure 10: Return of DAC-regulated AIRL under different different α values

7 Conclusion

In this work, we presented DAC-AIRL, an integrated reinforcement learning framework designed to tackle the challenges of sample efficiency and domain shift. By combining Discriminator Actor-Critic (DAC) for efficient data augmentation with Adversarial Inverse Reinforcement Learning (AIRL) for robust policy learning, the framework achieves both efficient learning and policy generalization across diverse environments.

Through a systematic workflow, DAC serves as a data augmentation module that generates high-quality trajectories, while AIRL leverages these augmented datasets to infer reward functions and train robust policies. Through experiments in the BipedalWalker-v3 environment, we demonstrated that DAC-AIRL outperforms standard AIRL in generating policies with limited data, highlighting the synergy between the two methodologies. DAC enhances expert datasets with higher sample efficiency, while AIRL leverages these augmented datasets to infer reliable reward functions and train adaptive policies under domain shift in comparison with DAC. Furthermore, bottom-up integration such as DAC-regulated AIRL method also show great potential based on its policy stability and learning progress shown in our experiment.

This work underscores the potential for integrating data-efficient learning with domain-adaptive techniques to address real-world RL challenges. Future research could extend the framework to more complex environments and explore additional techniques to further improve adaptability and scalability in dynamic and resource-constrained settings.

References

[1] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. Retrieved from `https://arxiv.org/abs/1801.01290`.

[2] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with deep reinforcement learning. Retrieved from `https://arxiv.org/abs/1312.5602`.

[3] Kostrikov, I., Agrawal, K.K., Dwibedi, D., Levine, S., & Tompson, J. (2018). Discriminator-Actor-Critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. Retrieved from `https://arxiv.org/abs/1809.02925`.

[4] Fu, J., Luo, K., & Levine, S. (2018). Learning robust rewards with adversarial inverse reinforcement learning. Retrieved from `https://arxiv.org/abs/1710.11248`.

[5] Pathak, D., Agrawal, P., Efros, A.A., & Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *Advances in Neural Information Processing Systems 7*. Retrieved from `https://arxiv.org/abs/1705.05363`.

[6] Pathak, D., Agrawal, P., Efros, A.A., & Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Retrieved from `https://arxiv.org/abs/1705.05363`.

[7] Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pp. 4565–4573.

[8] Han, C., Fan, Z., Zhang, D., Qiu, M., Gao, M., & Zhou, A. (2021). Meta-Learning Adversarial Domain Adaptation Network for Few-Shot Text Classification. *arXiv preprint arXiv:2107.12262*. Retrieved from `https://arxiv.org/abs/2107.12262`

[9] Rohit, J., Changliu, L., Katia, S.(2021). Augmenting GAIL with BC for sample efficient imitation learning. Retrieved from `https://arxiv.org/abs/2001.07798`

A Pseudocode of DAC

This is the pseudocode of DAC from the work of Kostrikov et al[3].

Algorithm 1 Discriminative-Actor-Critic Adversarial Imitation Learning Algorithm

Require: Expert replay buffer $\mathcal{R}_E$
procedure WRAPFORABSORBINGSTATES(τ)
if s_T is a terminal state **then**
  $(s_T, a_T, \cdot, s'_T) \leftarrow (s_T, a_T, \cdot, s_a)$
  $\tau \leftarrow \tau \cup \{(s_a, \cdot, \cdot, s_a)\}$
end if
return τ
end procedure
Initialize replay buffer $\mathcal{R} \leftarrow \emptyset$
for $\tau = \{(s_t, a_t, \cdot, s'_t)\}_{t=1}^{T} \in \mathcal{R}_E$ **do**
  $\tau \leftarrow$ WRAPFORABSORBINGSTATES(τ) *# Wrap expert rollouts with absorbing states*
end for
for $n = 1, \ldots, N$ **do**
  Sample $\tau = \{(s_t, a_t, \cdot, s'_t)\}_{t=1}^{T}$ with π_θ
  $\mathcal{R} \leftarrow \mathcal{R} \cup$ WRAPFORABSORBINGSTATES(τ) *# Update Policy Replay Buffer*
  for $i = 1, \ldots, |\tau|$ **do**
    $\{(s_t, a_t, \cdot, \cdot)\}_{b=1}^{B} \sim \mathcal{R}$, $\{(s'_b, a'_b, \cdot, \cdot)\}_{b=1}^{B} \sim \mathcal{R}_E$ *# Mini-batch sampling*
    $\mathcal{L} = \sum_{b=1}^{B} \log D(s_b, a_b) - \log(1 - D(s'_b, a'_b))$
    Update D with GAN+GP
  end for
  for $i = 1, \ldots, |\tau|$ **do**
    $\{(s_t, a_t, \cdot, s'_t)\}_{b=1}^{B} \sim \mathcal{R}$
    for $b = 1, \ldots, B$ **do**
      $r \leftarrow \log D(s_b, a_b) - \log(1 - D(s_b, a_b))$
      $(s_b, a_b, \cdot, s'_b) \leftarrow (s_b, a_b, r, s'_b)$
    end for
    Update π_θ with TD3 *# Use current reward estimate*
  end for
end for

B Pseudocode of AIRL

This is the pseudocode of AIRL from the work of Fu et al[4].

Algorithm 2 Adversarial Inverse Reinforcement Learning

Obtain expert trajectories τ_i^E
Initialize policy π and discriminator $D_{\theta,\phi}$
for step $t \in \{1, \ldots, N\}$ **do**
  Collect trajectories $\tau_i = (s_0, a_0, \ldots, s_T, a_T)$ by executing π
  Train $D_{\theta,\phi}$ via binary logistic regression to classify expert data τ_i^E from samples τ_i
  Update reward $r_{\theta,\phi}(s, a, s') \leftarrow \log D_{\theta,\phi}(s, a, s') - \log(1 - D_{\theta,\phi}(s, a, s'))$
  Update π with respect to $r_{\theta,\phi}$ using any policy optimization method
end for