SCIENCE CHINA
Information Sciences

• Supplementary File •

Expert policy support expansion: alleviating the overfitting issue of adversarial imitation learning for high-dimensional tasks

Fuguang HAN[1, 2†], Zhilong ZHANG[1, 2, 3†], Yichen LI[1, 2, 3], Feng CHEN[1, 2], Yang YU[1, 2, 3] & Zongzhang ZHANG[1, 2*]

[1]*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210023*, China*
[2]*School of Artificial Intelligence, Nanjing University, Nanjing* 210023*, China*
[3]*Polixir Technologies, Nanjing* 211106*, China*

Appendix A Implementation details

In our implementation, our code builds upon the open-source implementation of PatchAIL [1], which encompasses two baseline approaches: Shared-Encoder AIL and Independent-Encoder AIL. We select the Shared-Encoder AIL as the foundational AIL algorithm in our implementation. We maintain consistency with the original PatchAIL framework in terms of the network architectures and parameters for the actor, critic, encoder, and discriminator. Furthermore, we also integrate the Random Shift image augmentation technique into our training in line with PatchAIL.

Table A1 List of hyperparameters for common setting.

Method	Hyperparameter	Value
AIL	Replay buffer size	100000
	Learning rate	1e-4
	Discount γ	0.99
	Frame stack / n-step returns	3
	Action repeat	2
	Mini-batch size	256
	Agent update frequency	2
	Critic soft-update rate	0.01
	Feature dim	50
	Hidden dim	1024
	Optimizer	Adam
	Exploration steps	2000
	DDPG exploration schedule	linear(1, 0.1, 500000)
	Gradient penalty coefficient	10
	Default data augmentation strategy	random shift
e-RND	Training epoch	1000
	Extent of the estimated expert support α	0.98
p-RND	Intrinsic reward weight β schedule	linear(1, 0, 1000000)

* Corresponding author (email: zzzhang@nju.edu.cn)
† These authors contributed equally to this work.

The complete list of hyperparameters is detailed in Table A1. Following PatchAIL, we have implemented slight modifications to the original configurations for specific tasks within DMControl [2], particularly for Walker Stand/Walk/Run and the Humanoid Stand task. For the Walker Stand/Walk/Run task, the mini-batch size has been adjusted to 512, and the n-step return parameter is set to 1. In the case of the Humanoid Stand task, we have modified the learning rate to 8×10^{-5} and increased the feature dimension to 100. Furthermore, for the Walker Run and Humanoid Stand tasks, we directly utilize the output of the discriminator as the reward. Conversely, for all other tasks, the reward is derived from the normalized output of the discriminator."

Regarding the e-RND module, while we strive for consistent hyperparameter settings across all tasks, the number of training epochs is adaptively determined by the convergence behavior of the supervised learning process. Specifically, the required epochs depend on the input complexity, which is dictated by the feature dimensionality and the volume of expert demonstrations. For instance, in the DMControl suite, we found that the Mean Squared Error (MSE) stabilizes at 800 epochs for Walker Run and 2000 epochs for Hopper Stand. Conversely, for Meta-World [3] tasks where only a single expert trajectory is provided, the model reaches convergence more rapidly, requiring approximately 300 epochs for tasks like Hammer and Bin Picking. By monitoring the MSE and terminating training once it plateaus, we ensure that the e-RND sufficiently captures the expert support while maintaining a principled and automated training protocol, rather than relying on task-specific manual tuning. Regarding the extent of the estimated expert support α, we treat this as a task-agnostic parameter. Our configuration is designed to strike an optimal balance between precision and recall: a threshold that is too high would erroneously include low-quality policy samples as positives, while an overly restrictive (low) threshold would exclude genuine expert-like states.

Regarding the p-RND module, the intrinsic reward weight β is designed to balance exploration and exploitation. Theoretically, a larger weight facilitates exploration in complex environments with sparse rewards, while a smaller weight prioritizes the exploitation of established reward structures in simpler tasks. As demonstrated by the sensitivity analysis in Appendix B.3, we found that our method is insensitive to β. Consequently, we opted for a uniform hyperparameter setting across almost all tasks. For the sake of transparency and reproducibility, a comprehensive summary of the task-specific hyperparameter configurations for AIL, e-RND, and p-RND across all DMControl and Meta-World tasks is provided in Table A2.

Table A2 Detailed hyperparameter configurations across various DMControl and Meta-World tasks.

Hyperparameter	**Walker (Stand/Walk/Run)**	**Hopper Stand**	**Humanoid Stand**	**Others**
Learning rate	1×10^{-4}	1×10^{-4}	8×10^{-5}	1×10^{-4}
Mini-batch size	512	256	256	256
n-step	1	3	3	3
Feature dimension	50	100	50	50
Reward type	Original	Normalized	Original	Normalized
Training epoch of e-RND	800	2000	1200	[c]1000 (DMC)
300 (Meta)				
Expert support α	0.98	0.98	0.98	0.98
Intrinsic reward weight β	linear(1, 0, 1M)	linear(1, 0, 1M)	linear(1, 0, 5M)	linear(1, 0, 1M)

In the initial training stages, e-RND faces a limitation due to the sparse presence of expert-quality samples in the replay buffer, which in turn limits the effectiveness of e-RND. To overcome this challenge, our approach integrates Behavioral Cloning (BC) [4, 5]. This integration facilitates the development of a policy capable of producing higher-quality samples, surpassing those generated by a randomly initialized policy. As a result, this approach allows e-RND to filter a significant number of expert samples, effectively preventing the early onset of overfitting in the discriminator.

Appendix B Additional experimental results

Appendix B.1 Ablation experiments on DMControl tasks

In this section, we detail the outcomes of ablation studies conducted on eight DMControl tasks, as depicted in Figure B1.

This figure elucidates the roles of e-RND and p-RND within our framework. Our findings reveal that e-RND effectively counters discriminator overfitting by augmenting expert data, consistently outperforming DAC [6] across all DMControl tasks. On the other hand, p-RND shows comparatively less effective results than DAC in simpler tasks such as Finger Spin, Walker Stand, and Walker Walk. This can be attributed to the already rapid convergence of the vanilla AIL algorithm in these tasks. The intrinsic reward component in p-RND encourages exploration, which, while beneficial for complex tasks, results in slower convergence in simpler scenarios. Nevertheless, in more complex tasks, the exploration incentivized by p-RND's intrinsic reward mechanism leads to superior performance compared to DAC.

Appendix B.2 Visualization and analysis of sample distributions

To further investigate the mechanism of support expansion, we provide visual evidence at both the representation and decision levels.

We visualize the 50-dimensional representations extracted from the encoder using t-SNE [7] and analyze the discriminator's output distributions via Kernel Density Estimation (KDE) [8, 9] for three distinct sample categories: expert demonstrations, e-RND filtered expert-like positive samples, and remaining negative samples. As shown in Figure B2(a), the t-SNE visualization reveals that the

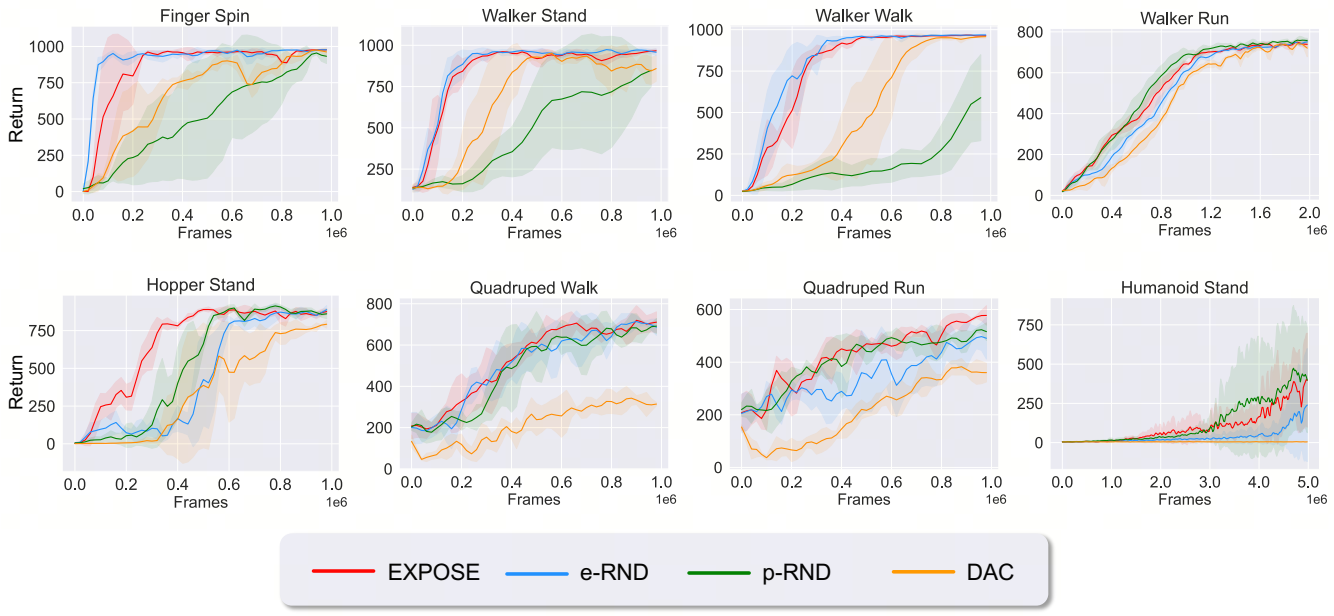
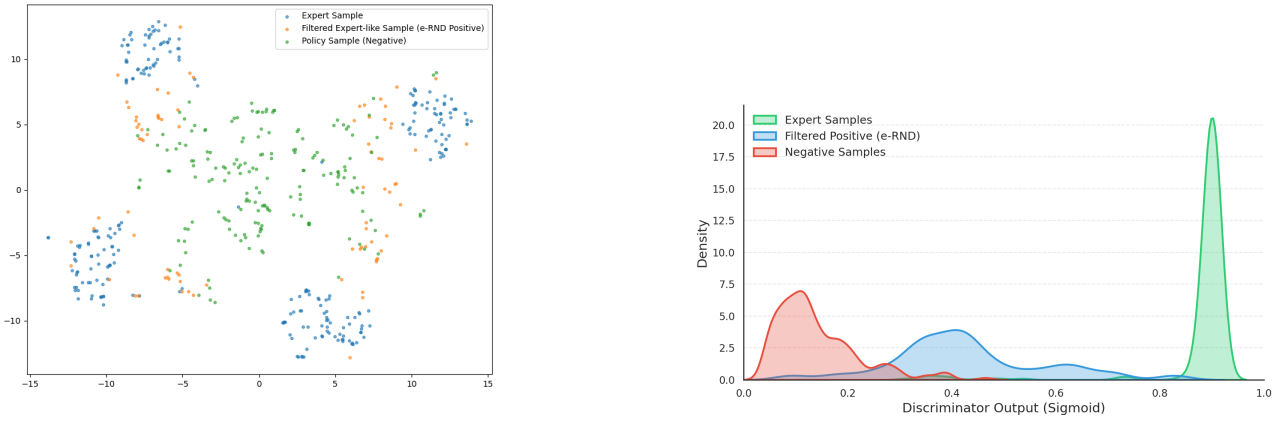


Figure B1 Ablation experiments on 8 DMControl tasks over 5 random seeds using 10 expert trajectories.

expert-like samples filtered by e-RND are predominantly situated in the regions surrounding the expert samples in the learned representation space. This proximity indicates that e-RND successfully identifies agent behaviors that are representationally similar to expert demonstrations.

Furthermore, the discriminator output distribution in Figure B2(b) shows that the filtered positive samples by e-RND receive intermediate scores that lie between those of the high-scoring expert samples and the low-scoring negative samples. This indicates that treating representationally similar samples as positives effectively prevents the discriminator from learning an overly sharp, binary-like decision boundary. By assigning transitional values to these samples, the discriminator learns a broader and more robust support for the expert distribution. These visual results suggest that our method facilitates a smoother reward landscape, which contributes to the improved training stability and generalization observed in our experiments.



(a) (b)

Figure B2 Visualization of the support expansion's effect. (a) t-SNE visualization of feature space; (b) distribution of discriminator output.

Appendix B.3 Parameter sensitivity study for e-RND

In our work, e-RND employs a sample filtration mechanism based on prediction error. Specifically, a sample is filtered if its prediction error is less than a threshold, defined as a predefined constant α multiplied by the maximum predicted error of e-RND over the expert dataset. The parameter α is pivotal as it dictates the extent of the estimated expert support, with a default setting of 0.98. Figure B3 illustrates the impact of varying α values on the experimental outcomes in the Hopper Stand and Quadruped Walk tasks. A lower α translates to a more stringent filter, allowing fewer expert samples to be considered as positive samples for training the discriminator. Conversely, a higher α tends to overestimate the support of the expert policy, potentially leading to the inclusion of numerous low-quality

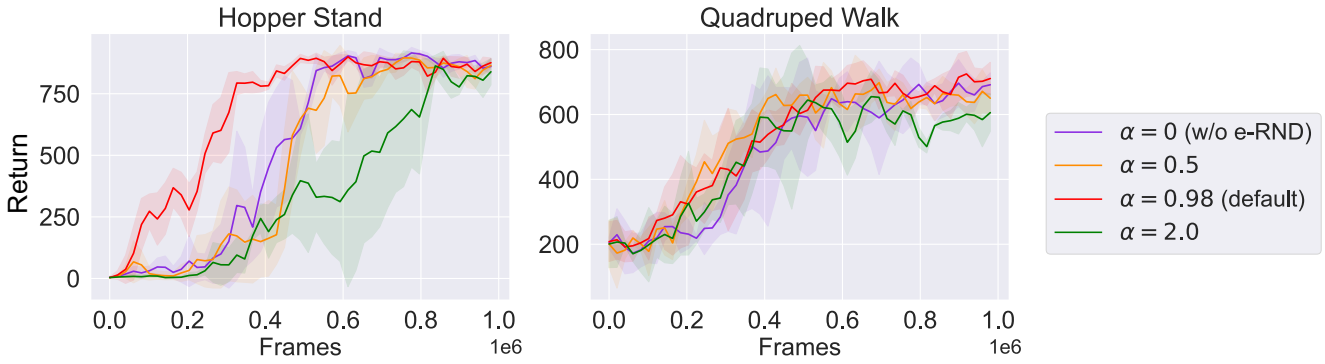


Figure B3 Parameter sensitivity study for α in e-RND on Hopper Stand and Quadrueped Walk.

samples in positive training samples. Our results indicate that an increased α value markedly diminishes performance. In the Hopper Stand task, our findings indicate that a lower α value in e-RND yields performance comparable to scenarios where e-RND is not utilized since a lower α values result in few samples can be filtered. However, in the Quadruped Walk task, performance appears to be less sensitive to variations in the α parameter of e-RND because employing p-RND alone suffices to attain commendable performance in this task.

Appendix B.4 Parameter sensitivity study for p-RND

In our study, we further examine the impact of exploration intensity of p-RND on our experimental outcomes, as shown in Figure B4. This investigation primarily revolves around modifying the initial value of β, which is programmed to decay to 0 by the end of the training process.

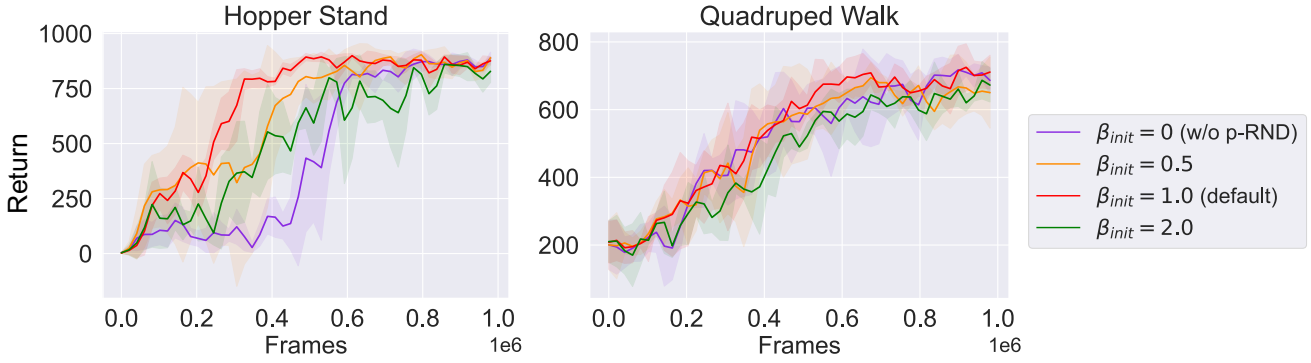


Figure B4 Parameter sensitivity study for β in p-RND on Hopper Stand and Quadrueped Walk.

In our study, we observe distinct effects of the β parameter on policy behavior. Specifically, a higher β value inclines the policy towards prioritizing exploration over exploitation. This tendency promotes a more extensive search through the state space but may delay the capitalization on known high-reward behaviors. In contrast, a lower β slows down the exploration of the policy, potentially hindering the discovery of optimal policy. Our experimental results highlight that a proper β strikes an effective balance between exploration and exploitation, not only ensuring a thorough exploration of policy space but also facilitates a faster convergence of the policy.

References

1 Liu M, He T, Zhang W, et al. Visual imitation learning with patch rewards. In: Proceedings of International Conference on Learning Representations, 2022
2 Tassa Y, Doron Y, Muldal A, et al. DeepMind control suite. arXiv preprint arXiv:1801.00690, 2018
3 Yu T, Quillen D, He Z, et al. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Proceedings of Conference on Robot Learning, 2020. 1094–1100
4 Pomerleau D A. ALVINN: An autonomous land vehicle in a neural network. In: Proceedings of Advances in Neural Information Processing Systems, 1988. 305–313
5 Torabi F, Warnell G, Stone P. Recent advances in imitation learning from observation. In: Proceedings of International Joint Conference on Artificial Intelligence, 2019. 6325–6331
6 Kostrikov I, Agrawal K K, Dwibedi D, et al. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In: Proceedings of International Conference on Learning Representations, 2019
7 Van der Maaten L, Hinton G. Visualizing data using t-sne. Journal of machine learning research, 2008, 9
8 Rosenblatt M. Remarks on some nonparametric estimates of a density function. The Annals of Mathematical Statistics, 1956, 27: 832–837
9 Parzen E. On estimation of a probability density function and mode. The Annals of Mathematical Statistics, 1962, 33: 1065–1076