Learning Agile Skills via Adversarial Imitation of Rough Partial Demonstrations

Anonymous Author(s)
Affiliation
Address
email

Abstract: Learning agile skills is one of the main challenges in robotics. To this end, reinforcement learning approaches have achieved impressive results. These methods require explicit task information in terms of a reward function or an expert that can be queried in simulation to provide a target control output, which limits their applicability. In this work, we propose a generative adversarial method for inferring reward functions from partial and potentially physically incompatible demonstrations for successful skill acquirement where reference or expert demonstrations are not easily accessible. Moreover, we show that by using a Wasserstein GAN formulation and transitions from demonstrations with rough and partial information as input, we are able to extract policies that are robust and capable of imitating demonstrated behaviors. Finally, the obtained skills such as a backflip are tested on an agile quadruped robot called Solo 8 and present faithful replication of hand-held human demonstrations.

Keywords: Adversarial, Imitation Learning, Legged Robots

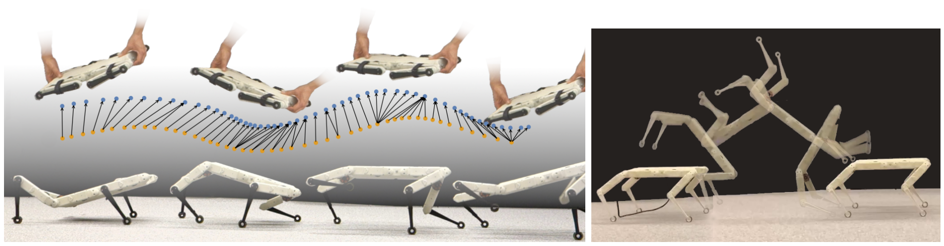


Figure 1: Our method (WASABI) achieves agile physical behaviors from rough (hand-held) and partial (robot base) motions. The illustrated performance measure is the Dynamic Time Warping distance of the base trajectories (left). A learned backflip policy is deployed on Solo 8 (right).

1 Introduction

Obtaining dynamic skills for autonomous machines has been a cardinal challenge in robotics. A primary shortage of motivating desired behaviors by reward engineering is the arduous reward-shaping process involved. Given the availability of some expert references, one possible solution is Imitation Learning (IL), which aims to mimic expert behaviors in a given task. In particular, Generative Adversarial Imitation Learning (GAIL) [1] draws a connection between IL and generative adversarial networks (GANs) [2], which train a policy to deceive a discriminator that constantly tries to distinguish state transitions generated between the policy and the reference data distribution. The output of the discriminator can then be used as a reward that encourages the learning agent to generate similar behaviors to the demonstration.

In this work, we present a novel adversarial imitation learning method named Wasserstein Adversarial Behavior Imitation (WASABI). We show that we are able to extract sensible task rewards from rough and partial demonstrations by utilizing adversarial training for obtaining agile skills in a sim-to-real

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

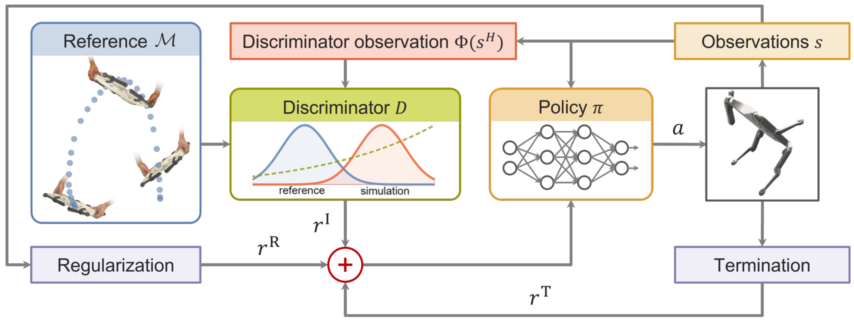


Figure 2: System overview. Given a reference dataset defining the desired base motion, the system trains a discriminator that learns an imitation reward for the policy training.

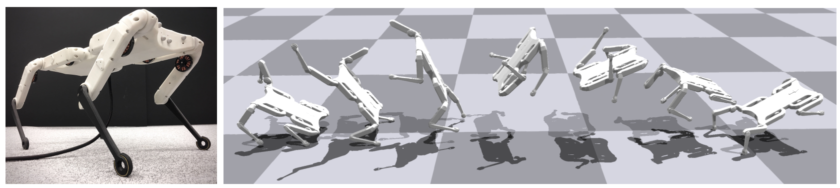


Figure 3: Solo 8 (left). Backflip motion in Isaac Gym (right).

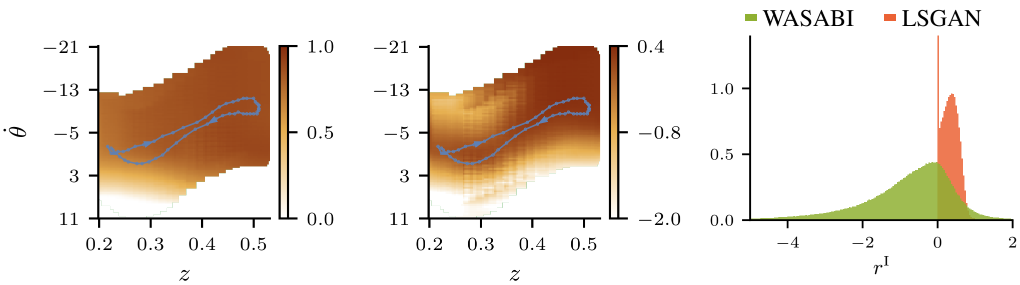
setting. In contrast to Peng et al. [3], our approach does not require any prior information about the task at hand in form of a specific reward function, but only reasonable task-agnostic regularization terms in addition to the adversarial reward that make the robot motion more stable. Most importantly, we achieve this without having access to samples from an expert policy, but rather hand-held human demonstrations that are physically incompatible with the robot itself. To the best of our knowledge, this is the first time that highly dynamic skills are obtained from limited reference information. In summary, our contributions include: **(i)** An adversarial approach for learning from partial, physically incompatible demonstrations. **(ii)** Analysis of the Least-Squares vs. Wasserstein GAN loss for reward inference. **(iii)** Experimental validation in simulation and on a quadruped robot. Figure 2 provides a schematic overview of our method. Supplementary videos for this work are available at `https://sites.google.com/view/corl2022-wasabi/home`.

2 Experiments

We evaluate WASABI on the Solo 8 robot, an open-source research quadruped robot that performs a wide range of physical actions [4], in simulation and on the real system (Fig. 3). For evaluation, we introduce 4 different robotics tasks. We provide *rough* demonstrations of these motions by manually carrying the robot through the motion and recording only the base information. The demonstrations are then used to infer an adversarial imitation reward for training a control policy that outputs target joint positions. In all of our experiments, we use Proximal Policy Optimization (PPO) [5] in Isaac Gym [6] and make use of domain randomization [7] for sim-to-real transfer.

2.1 Induced Imitation Reward Distributions

The LSGAN loss is proposed to alleviate the saturation problem that is encountered for the CEGAN loss. Yet, it does not directly yield a practical reward function. Peng et al. [3] remedy this by using $r^{\mathrm{I}} = \max\left[0,\ 1 - 0.25(D\left(\Phi(s), \Phi(s')\right) - 1)^2\right]$ to map the discriminator output to the imitation reward and bound it between 0 and 1. However, with the effective clipping at 0, information about the distance from the policy to the demonstration transitions is lost with discriminator prediction smaller than -1 (Fig. 4c). In addition, we show in Fig. 4a that the imitation reward learned using LSGAN yields a less informative signal for policy training, which is rather uniformly distributed across pitch rate $\dot{\theta}$ and base height z dimensions. In comparison, WASABI can use the discriminator output directly, learning a more characteristic reward function across the state space where reference trajectories are clearly outlined to yield high rewards in contrast to the off-trajectory states (Fig. 4b).



(a) LSGAN (b) WASABI (WGAN) (c) Distribution of r^{I}

Figure 4: Adversarial imitation rewards for SoloBackFlip. Imitation reward heatmap for LSGAN (a) and WASABI (b) around reference trajectories (blue) generated in varying pitch rate $\dot{\theta}$ and base height z. (c) Distribution of imitation rewards for LSGAN and WASABI during training. WASABI provides a more fine-grained reward function.

Method	SoloLeap	SoloWave	SoloStandUp	SoloBackFlip
WASABI	$\mathbf{131.70 \pm 16.44}$	$\mathbf{247.29 \pm 11.59}$	$\mathbf{351.13 \pm 88.60}$	$\mathbf{477.43 \pm 56.77}$
LSGAN	$\mathbf{155.31 \pm 18.10}$	$\mathbf{230.91 \pm 5.95}$	678.21 ± 6.71	813.76 ± 19.75
Stand Still	216.41	460.15	494.40	877.74

Table 1: Comparison of performances for LSGAN and WASABI trained with hand-held demonstrations in terms of **DTW distance** d^{DTW} (lower is better), successful runs are in **bold** font. As a reference, we provide also d^{DTW} of a constantly standing trajectory.

2.2 Learning to Mimic Rough Demonstrations

Since we record the base motion of the robot carried by a human demonstrator, we do not have access to a reward function evaluating learned behaviors or measuring the closeness between the demonstrated and the policy trajectories. In addition, these trajectories are largely misaligned. For this reason, we make use of Dynamic Time Warping (DTW) [8] with the L_2 norm metric for comparing policy trajectories and reference demonstrations. In Table 1 we compare performances in simulation for the different reference motions.

In order to confirm that WASABI is indeed able to extract a sensible reward function that motivates the desired motion, we compare the performance of LSGAN and WASABI in SoloStandUp and SoloBackFlip using an expert baseline that is trained on a handcrafted task reward for generating demonstrations in simulation. The learned policies are evaluated with the same task rewards that are used to obtain the expert policies. A comparison of training performance curves in terms of the corresponding handcrafted task rewards is detailed in Fig. 5. In Table 2 we show the performance evaluation of the best runs.

2.3 Evaluation on Real Robot

To evaluate our method on real system, we trained policies for sim-to-real transfer with WASABI for the SoloLeap, SoloWave and SoloBackFlip. During deployment, we recorded the robot

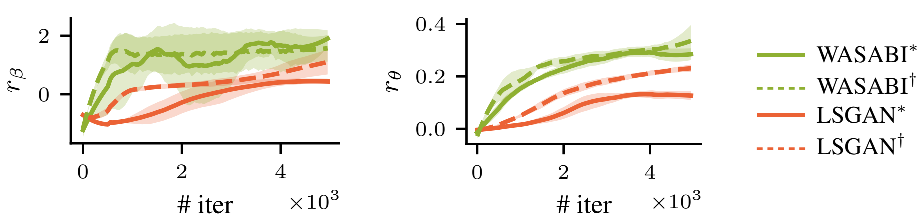


Figure 5: Performance of WASABI and LSGAN in terms of the handcrafted task reward for SoloStandUp (left) and SoloBackFlip (right). Dashed lines indicate partial information (†).

Method	SoloStandUp†	SoloStandUp*	SoloBackFlip†	SoloBackFlip*
WASABI	$\mathbf{1.54 \pm 0.51}$	$\mathbf{1.68 \pm 0.51}$	$\mathbf{0.36 \pm 0.05}$	$\mathbf{0.28 \pm 0.02}$
LSGAN	1.07 ± 0.5	0.44 ± 0.14	0.12 ± 0.01	0.06 ± 0.01
Handcrafted	$\mathbf{2.24 \pm 0.05}$		$\mathbf{0.77 \pm 0.04}$	

Table 2: Performance comparison in terms of handcrafted **task reward** (higher is better). We denote with $*$ where the full robot configuration is given to the discriminator and $\dagger$ where only base information is given. Successful runs are in **bold** font. Std-dev. is over 5 independent random seeds.

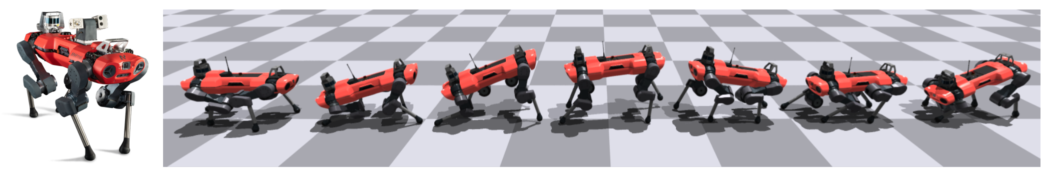


Figure 6: ANYmal C (left). Wave motion in Isaac Gym (right).

base information for evaluation by d^{DTW}. The resulting performance on the real system, as shown in Table 3, resembles the performance obtained in simulation.

	SoloLeap	SoloWave	SoloBackFlip
WASABI (Real)	153.64 ± 7.08	215.38 ± 21.82	504.26 ± 18.90
WASABI (Sim)	131.70 ± 16.44	247.29 ± 11.59	477.43 ± 56.77

Table 3: Sim-to-real performance on the Solo 8 in terms of DTW distance (lower is better). Values are computed from the recorded data of the learned policies with respect to the reference trajectories.

2.4 Cross-platform Imitation

As the reference motion in WASABI contains only base information, it does not restrict itself to be obtained only from any specific robotic platform. This provides the possibility of cross-platform imitation. Using the reference trajectories recorded from Solo 8, we apply WASABI to ANYmal [9], a four-legged dog-like robot for research and industrial maintenance (Fig. 6). To confirm that WASABI applies to cross-platform imitation, we define ANYmalWave and ANYmalBackFlip tasks for the corresponding wave and backflip motions learned by ANYmal, yet from the reference data recorded from Solo 8. The performance in terms of the DTW distance is detailed in Table 4.

Method	SoloWave	ANYmalWave	SoloBackFlip	ANYmalBackFlip
WASABI	$\mathbf{247.29 \pm 11.59}$	$\mathbf{193.08 \pm 14.52}$	$\mathbf{477.43 \pm 56.77}$	$\mathbf{572.60 \pm 12.18}$
Stand Still	460.15		877.74	

Table 4: Performance of cross-platform imitation of ANYmal using WASABI trained with hand-held demonstrations from Solo 8 in terms of **DTW distance** d^{DTW}, successful runs are in **bold** font.

3 Conclusion

In this work, we propose an adversarial imitation method named WASABI for inferring reward functions that is capable of learning agile skills from partial and physically incompatible demonstrations without any a priori known reward terms. Our results indicate that WASABI allows extracting robust policies that are able to transfer to the real system and enables cross-platform imitation. For highly agile or incompatible motions which initially seem beyond the robot's capability, WASABI outperforms LSGAN by successful and faithful replication of roughly demonstrated behaviors.

References

[1] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems*, volume 3, 06 2014.

[3] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4):1–20, 2021.

[4] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols, J. Fiene, A. Badri-Spröwitz, and L. Righetti. An open torque-controlled modular robot architecture for legged locomotion research. *IEEE Robotics and Automation Letters*, 5(2):3650–3657, 2020. doi:10.1109/LRA.2020.2976639.

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

[6] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[7] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017.

[8] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD Workshop*, 1994.

[9] Anymal c – the next step in robotic industrial inspection, August 2019. URL `https://www.anybotics.com/the-next-step-in-robotic-industrial-inspection/`.