

# STABLE BASIS DEEP NEURAL POLICY TRAINING

**Anonymous authors**

Paper under double-blind review

## 1 PERFORMANCE ANALYSIS RESULTS

Table 1 provides a detailed comparison of algorithmic properties. In the main body of the paper we have reported results for BC (Ross & Bagnell, 2010), GAIL (Ho & Ermon, 2016), SQIL (Reddy et al., 2020), vDICE (Kostrikov et al., 2020), inverse- $Q$  learning (Garg et al., 2021) and our proposed algorithm harmonic learning. These results can also be found in Table 2. Table 2 reports results with 50K environment interaction training. Inverse- $Q$  learning is the closest being able to learn a policy in a high-dimensional observation MDP, i.e. observations consisting of pixels as in the Arcade Learning Environment as also have been demonstrated in the inverse- $Q$  learning paper (Garg et al., 2021). These results once more demonstrate that harmonic learning results in substantial performance gains.

Table 1: Comparison of Algorithms with respect to properties of dynamics awareness, adversarial training, non restrictive rewards, and direct optimization.

Training Method	Reference	Dynamics Aware	Non-Adversarial Training	Non Restrictive Reward	Direct Optimization
GAIL	(Ho & Ermon, 2016)	✓	✗	✓	✗
SQIL	(Reddy et al., 2020)	✓	✓	✗	✓
vDICE	(Kostrikov et al., 2020)	✓	✗	✗	✗
Behaviour Cloning (BC)	(Ross & Bagnell, 2010)	✗	✓	✗	✓
Inverse $Q$ -learning	(Garg et al., 2021)	✓	✓	✓	✓
<b>Harmonic Learning</b>	<b>Ours</b>	✓	✓	✓	✓

Table 2: Performance analysis results for BC, GAIL, SQIL, vDICE, harmonic learning and inverse  $Q$ -learning. Table reports raw scores obtained by the policies in high-dimensional MDPs.

MDP	BC	GAIL	SQIL	vDICE	InverseQL	Ours
Pong	-11.00	-20.3 $\pm$ 0.008	-19.81 $\pm$ 0.012	-20.91 $\pm$ 0.001	8.0 $\pm$ 5.3814	<b>19.0<math>\pm</math>1.89736</b>
SpaceInvader	300	289 $\pm$ 24.1	218 $\pm$ 8.9	176 $\pm$ 12.8	470.5 $\pm$ 23.68	<b>609.0<math>\pm</math>14.5223</b>
Breakout	0.00 $\pm$ 0.00	0.5 $\pm$ 0.001	3.6 $\pm$ 0.021	4.2 $\pm$ 0.031	108.9 $\pm$ 29.72	<b>228.8<math>\pm</math> 35.4606</b>

## 2 EXPERT DEMONSTRATIONS AND HARMONIC LEARNING

Another important metric that we can measure and provide is the performance of the inverse- $Q$  learning algorithm and the harmonic learning algorithm when compared to the performance level of an expert in a data-limited setting.

Table 3 reports results of raw scores obtained by the harmonic learning policy, inverse  $Q$ -learning policy and the expert policy in Pong, Breakout, Seaquest, SpaceInvaders and BeamRider.

Table 3: Raw scores obtained by harmonic learning policy, inverse  $Q$ -learning policy and the expert policy in Pong, Breakout, Seaquest, SpaceInvaders and BeamRider.

Training Method	Harmonic Learning	Inverse $Q$ -learning	Expert Policy
Pong	19.0 $\pm$ 1.89736	8.0 $\pm$ 5.3814	21 $\pm$ 0.0
Seaquest	906.0 $\pm$ 53.2202	864.0 $\pm$ 42.0285	2393 $\pm$ 291.0
SpaceInvader	609.0 $\pm$ 14.5223	470.555 $\pm$ 23.6812	823.0 $\pm$ 272.0
BeamRider	1023.6 $\pm$ 140.974	909.6 $\pm$ 65.392	4295.0 $\pm$ 1173.0
Breakout	228.8 $\pm$ 35.4606	108.9 $\pm$ 29.7198	376.0 $\pm$ 34.0

Table 4 reports the percentage of the expert policy performance that harmonic learning and the inverse- $Q$  learning policy reach with only **50K environment interactions**. The percentage of the expert policy results once more demonstrate clearly the sample efficiency gains achieved by the harmonic learning algorithm.

Table 4: Percentage of the expert policy performance achieved by the harmonic learning policy and the inverse  $Q$ -learning policy in Pong, Breakout, Seaquest, SpaceInvaders and BeamRider.

Training Method	Harmonic Learning	Inverse Q-learning
Pong	<b>90.47%±9.03%</b>	38.09%±25.23%
Seaquest	<b>37.86%±2.22%</b>	36.10%±1.756%
SpaceInvader	<b>73.99%±1.764%</b>	57.16%±2.877%
BeamRider	<b>26.15%±3.282%</b>	21.17%±0.125%
Breakout	<b>60.64%± 9.43%</b>	28.71%±7.90%

### 3 CODE FOR HARMONIC LEARNING ALGORITHM

*Our algorithm does not require any gradient and function evaluations, and only takes 9 lines of code.*

```

for episode_step in range(EPISODE_STEPS):
    if steps < args.num_seed_steps:
        action = env.action_space.sample()
    else:
        with train_mode(agent):
            dim_obs = np.size(state)[0, :, 0]
            fourier_obs_initial = np.fft.fft2(np.array(state))
            LQ = np.random.randint(dim_obs/2, size=1)
            HQ = dim_obs-LQ-1
            fourier_obs_initial[:, LQ, LQ:HQ] = 0
            fourier_obs_initial[:, HQ, LQ:HQ] = 0
            fourier_obs_initial[:, LQ:HQ, LQ] = 0
            fourier_obs_initial[:, LQ:HQ, HQ] = 0
            state = np.fft.ifft2(fourier_obs_initial)
            action = agent.choose_action(state, sample=True)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        steps += 1

```

Figure 1: Harmonic learning code. Harmonic learning is easy to implement and only takes 9 lines of code with substantial sample-efficiency gains resulting in more generalizable and robust policies.

Figure 1 reports the code for harmonic learning. **Our algorithm does not require any gradient and function evaluations, and only takes 9 lines of code.** Harmonic learning is an extremely fast and efficient algorithm.

### 4 ADDITIONAL RESULTS AND STATE REPRESENTATIONS WITH SPECTRAL PROPERTIES

Figure 4 reports stable basis robustness analysis results for the inverse  $Q$ -learning policy and deep neural policies trained via harmonic learning policy in SpaceInvaders and Breakout. The results reported in Figure 4 once more demonstrate that harmonic learning yields learning of more robust policies. The results reported in Figure 5 of the main body of the paper further demonstrate that the

robustness and resilience properties of harmonic learning further extend substantially to distributional shift.<sup>1</sup> One of the concrete reasons for the level of achieved robustness by harmonic learning lies in Figure 4 and Table 2 of the main body of the paper. The results reported in Figure 4 from the main

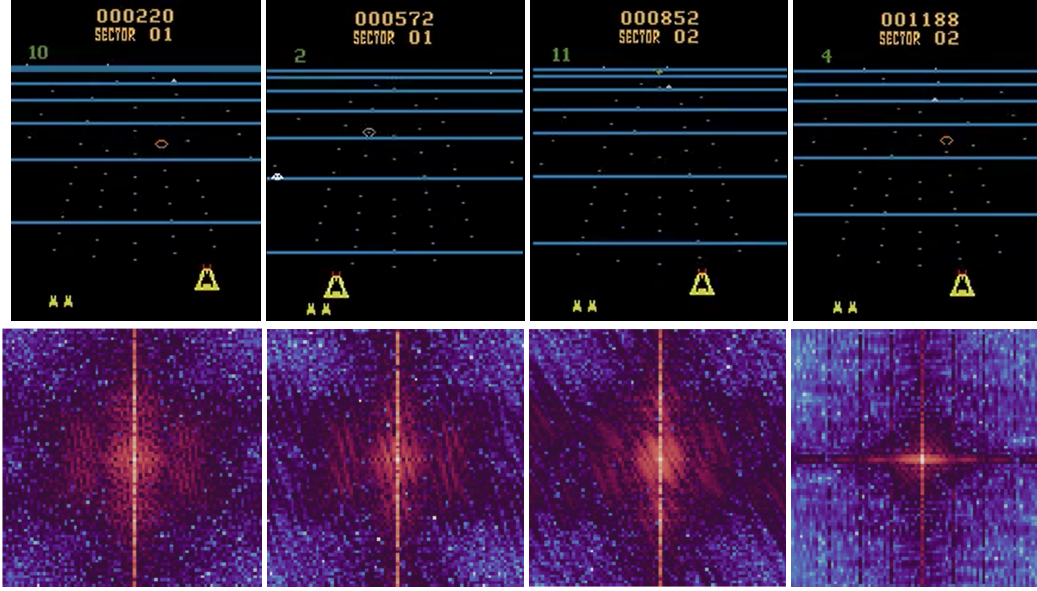


Figure 2: State observations of the high-dimensional state representation MDPs and the spectral representations in BeamRider.

body of the paper demonstrate that the inverse  $Q$ -learning policy further suffers from overfitting of the state-action value function, and harmonic learning addresses this problem resulting in learning a more correct estimation of the state-action values. More clearly, Figure 5 of the main body of the paper demonstrates the sensitivities of the deep inverse reinforcement learning policy, and further harmonic learning provides a concrete theoretically well-founded solution for these sensitivities and vulnerabilities of the deep inverse reinforcement learning policies. Also note that the methods focusing on robustification of the policies (i.e. adversarial training) result in **sample inefficiency**. However, our theoretically well-founded method demonstrates that while harmonic learning results in learning robust policies, it also further substantially increases the sample efficiency.

Figure 3 and Figure 2 demonstrate state observations and the corresponding spectral representations of these high-dimensional state representations.

## 5 HYPERPARAMETERS AND TRAINING SETUP

The hyperparameter settings are identical to the original algorithm setups that proposed these algorithms to provide a consistent and transparent analysis of comparison. In particular, for these settings BC is the identical setting to the original algorithm that proposed BC Ross & Bagnell (2010), GAIL is the identical setting to the original algorithm that proposed GAIL (Ho & Ermon, 2016), SQIL is the identical setting to the original algorithm that proposed SQIL (Reddy et al., 2020), vDICE is the identical setting to the original algorithm that proposed vDICE (Kostrikov et al., 2020), inverse- $Q$  learning is the identical setting to the original algorithm that proposed inverse- $Q$  learning (Garg et al., 2021). The results of the comprehensive comparison can also be found in Table 2. Table 5 further describes the hyperparameter settings for inverse- $Q$  learning and the harmonic learning algorithm. Note that the hyperparameters are fixed exactly to the inverse- $Q$  learning algorithm setting (Garg

<sup>1</sup>Some recent work also argued that state-of-the-art certified robustification methods for deep reinforcement learning policies results in learning policies that have worse reaction to distributional shift than straightforward vanilla trained deep reinforcement learning policies Korkmaz (2023). However, our proposed harmonic learning method demonstrates that deep neural policies trained via harmonic learning are more robust to both distributional shift and the overfitting problem, while further increasing the sample efficiency.

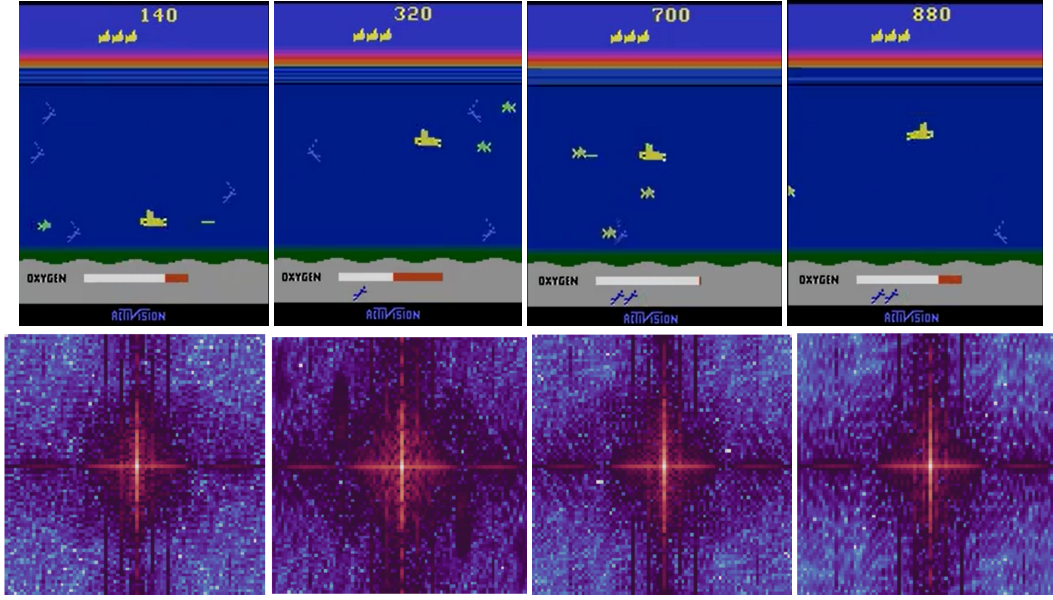


Figure 3: State observations of the high-dimensional state representation MDPs and the spectral representations in Seaquest.

et al., 2021) for transparency and fairness. The expert demonstrations are obtained from a DQN trained policy (Mnih et al., 2015) as also has been described in detail in (Garg et al., 2021). The activation functions for the connected layers are exponential linear units (ELU). Further note that all of the experiments conducted in our paper are in MDPs with high-dimensional state representations (i.e. Arcade Learning Environment) (Bellemare et al., 2013). Some of the expert demonstrations are obtained by using the (Raffin, 2020) pipeline as also has been described in detail in (Garg et al., 2021).

Table 5: Hyperparameter settings for the inverse- $Q$  learning and the harmonic learning algorithm.

Hyperparameters	Settings
Target Update Frequency	1000
Critic Learning Rate	$10^{-4}$
Initial Temperature	0.01
Critic $\tau$	0.1
Subsampling frequency	1
Replay Memory	150000
Initial Memory	5000
Demos	20
$\alpha$	0.5
$\epsilon$ steps	1000
$\epsilon$ window	100
Batch Size	64
Discount factor	0.99
Observation size	(84, 84)
Evaluation steps	5000
$Q$ -Network channels	32,64,64
$Q$ -Network filter size	$8 \times 8, 4 \times 4, 3 \times 3$
$Q$ -Network stride	(4, 4), (2, 2), (1, 1)
$Q$ -Network hidden units	512

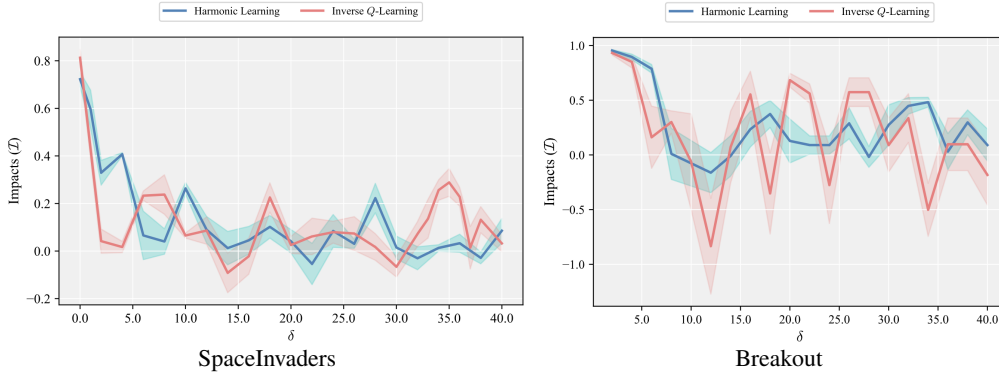


Figure 4: Stable Basis Robustness Analysis (SBRA) results for the inverse  $Q$ -learning policy and deep neural policies trained via harmonic learning in SpaceInvaders and Breakout.

## 6 THEORETICAL BASIS AND EMPIRICAL ANALYSIS

The uncertainty principle of Fourier transform states that a function and its Fourier transform can't be sharply concentrated at the same time: if one is very localized, the other must be spread out (Heisenberg, 1927; Benedicks, 1985). Thus, the removal of one element of the Fourier basis is spread out in the function without any semantic changes to the natural image. The optimal robust policy in high-dimensional MDPs should be robust to disturbances that have no true semantic meaning to the human eye, and hence should not have large dependence on any one part of the basis corresponding to a particular frequency.

Definition 3.1 introduces the notion of a stable basis, which is intuitively a basis in feature space such that the optimal policy depends approximately equally on each basis vector. Definition 3.4 extends the notion of stable-basis to the general function approximation setting, and Proposition 3.5 implies a natural method for testing if a given basis is a stable-basis by measuring the rewards obtained when removing the component of the observations along each stable basis vector. On the experimental side, the stable basis robustness analysis via Algorithm 3 is an operationalization of the test for a stable basis under general function approximation given by Proposition 3.5. In particular, by removing the observation components of basis vectors corresponding to particular frequencies. The results of stable basis robustness analysis in Figure 2 demonstrate that the vanilla trained policy is robust across the spectrum to the removal of these Fourier frequency components, and hence provides support to the claim that the Fourier basis is a stable-basis for the high-dimensional state observation MDPs considered. Furthermore, Proposition 3.3 implies that removing the observation components along randomly selected stable-basis vectors during training (i.e. Algorithm 2) corresponds to adding noise to the value function estimate proportional to a natural uncertainty measure given by Definition 3.2. Thus, training with Algorithm 2 has an interpretation that is analogous to randomized least-squares value iteration and other related methods for value-function randomization that yield provable regret bounds. Therefore, because the stable basis robustness analysis results indicate that the Fourier basis is a stable-basis for our setting of high-dimensional state observations, Proposition 3.3 provides theoretical motivation for the use of harmonic learning via Algorithm 3 to improve sample-efficiency. In connection to this the experimental results in Table 1 demonstrate notably improved sample-efficiency via harmonic learning, providing support for the theoretical justifications in Section 3.

## REFERENCES

- Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research.*, pp. 253–279, 2013.
- Michael Benedicks. On fourier transforms of functions supported on sets of finite lebesgue measure. *Journal of Mathematical Analysis and Applications*, 1985.

- Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Neural Information Processing Systems (NeurIPS) [Spotlight Presentation]*, 2021.
- Werner Heisenberg. On the intuitive content of quantum theoretical kinematics and mechanics. *Zeitschrift für Physik*, 1927.
- Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4565–4573, 2016.
- Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. *AAAI Conference on Artificial Intelligence*, 2023.
- Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, arc G Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Antonoglou, Helen King, Dharmashan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518: 529–533, 2015.
- Antonin Raffin. RL baselines3 zoo. <https://github.com/DLR-RM/rl-baselines3-zoo>, 2020.
- Siddharth Reddy, Anca D. Dragan, and Sergey Levine. SQIL: imitation learning via reinforcement learning with sparse rewards. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.
- Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In Yee Whye Teh and D. Mike Titterton (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pp. 661–668. JMLR.org, 2010.