RoboPianist: High-Dimensional Robotic Control of Bi-Dexterous Shadow Hands

Pankhuri Aggarwal Shelly Goel Yoko Nagafuchi

Stanford University CS224R Deep Reinforcement Learning Spring 2023

Objective

Apply two off-policy methods to train robotic control for bi-dexterous piano-playing tasks.

Zakka et al. introduced a training policy for dexterous high-dimensional control in precision, co-ordination, and planning. Our model integrates the Robopianist and NanoRL implementations,

- Two-hand C major scale, Nocturne Rousseau, and Twinkle Twinkle Little Star
- Hyperparameter tuning: pitch shifts and note stretches
- Model trains one song at a time

The simulation consisted of Piano, Hand, and Sound models:

- Standard 88-key digital piano with 52 white and 36 black keys.
- Left and Right Shadow Dexterous Hands from Mujoco Menagerie [53].
- Musical Instrument Digital Interface (MIDI) to represent pieces and synthesize sounds.

Models

Soft-Actor-Critic (SAC)

SAC is an off-policy algorithm that can be used for environments with continuous action spaces. The SAC algorithm focuses primarily on entropy regularization. "The policy is trained to maximize a trade-off between expected return and entropy, a measure of randomness in the policy. This has a close connection to the exploration-exploitation trade-off: increasing entropy results in more exploration, which can accelerate learning later on. It can also prevent the policy from prematurely converging to a bad local optimum" (OpenAI, 2018).

SAC concurrently learns a policy π_θ and two Q-functions Q_{ϕ_1} and Q_{ϕ_2}.

The loss functions for the Q-networks in SAC are:

$$L(\phi_i, \mathcal{D}) = \mathop{\mathrm{E}}_{(s,a,r,s',d)\sim\mathcal{D}} \left[\left(Q_{\phi_i}(s,a) - y(r,s',d) \right)^2 \right]$$

Twin Delayed Deep Deterministic Policy Gradients (TD3)

TD3 is an off-policy algorithm for environments with continuous action spaces. It is a successor to the Deep Deterministic Policy Gradient (DDPG) and addresses the issue of policy breaking due to overestimation of Q-values, which is prevalent in DDPG. To achieve this, it introduces 3 features namely, clipped double-Q learning, delayed policy updates and target policy smoothing. They have shown to substantially improve performance over a DDPG (OpenAI, 2018).

TD3 concurrently learns two Q-functions, Q_{ϕ_1} and Q_{ϕ_2}, by mean square Bellman error minimization. The policy is learned by maximizing Q_{ϕ_1}:

$$\max_\theta \mathop{\mathrm{E}}_{s\sim\mathcal{D}} \left[Q_{\phi_1}(s, \mu_\theta(s)) \right]$$

SAC and TD3 are a state-of-the-art methods and we experiment with them to understand and compare their impact on the training process of RoboPianist.

Experimental Results: Baseline

Song	Model Type	Return	Q-Value	Actor Loss	Critic Loss
C Major Scale	SAC	266.02	248.51	-250.89	2.90
	TD3	419.56	214.61	-215.33	0.46
Nocturne Rousseau	SAC	1028.93	257.36	-261.45	8.04
	TD3	1617.39	190.03	-190.77	0.39
D Major Scale	SAC	248.34	211.20	-211.90	0.77
	TD3	421.63	258.49	-261.35	5.93

Figure 1. Comparison of Training Scores of SAC and TD3 Models for C Major Scale and Nocturne Rousseau (No temporal/pitch shifts).

Hyperparameter Tuning Results: Temporal and Pitch Shifts

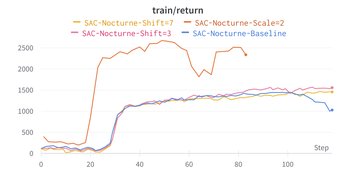
Song	Model Type	Pitch Shift	Temporal Shift	Return	Q-Value	Actor Loss	Critic Loss
C Major Scale	SAC	0	1.0	266.02	248.51	-250.89	2.90
	SAC	0	2.0	452.30	267.54	-271.01	11.37
	SAC	3	1.0	293.78	295.99	-297.59	2.55
	SAC	7	1.0	232.76	282.90	-286.01	9.79
	TD3	0	1.0	419.56	214.61	-215.33	0.46
	TD3	0	2.0	850.83	213.31	-214.09	0.52
	TD3	3	1.0	413.40	212.96	-213.70	0.48
	TD3	7	1.0	421.75	215.44	-216.12	0.47
Nocturne Rousseau	SAC	0	1.0	1028.93	257.36	-261.45	8.04
	SAC	0	2.0	2336.82	322.24	-325.33	3.20
	SAC	3	1.0	1558.17	226.89	-227.97	0.48
	SAC	7	1.0	1456.17	218.11	-219.10	0.44
	TD3	0	1.0	1617.39	190.03	-190.77	0.39
	TD3	0	2.0	2992.35	183.05	-183.78	0.38
	TD3	3	1.0	1615.95	192.09	-192.74	0.09
	TD3	7	1.0	1572.72	183.52	-184.20	0.1497

Figure 2. Comparison of Training Scores of SAC and TD3 Models for C Major Scale and Nocturne Rousseau with temporal and pitch shifts.

Final Model and Results

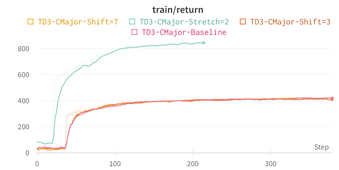
- The Nocturne Rousseau task is more complex than the C Major Scale task.
- Unexpectedly, we had a higher reward return with the Nocturne Rousseau task compared to the C Major Scale task.
- TD3 was more effective than SAC based on the baseline results in Figure 1, due to 1. less exploration as much as SAC and more focus on targeted exploitation, and 2. TD3's delayed policy updates for a more complex task like the nocturne.
- Hyperparameter tuning with pitch shifts: modifying the pitch of a note without modifying the duration by lowering it or raising the pitch by transposing a note to a different key. Pitch shifts didn't yield noticeable improvements in the reward return, perhaps due to the same relative motion of the hands.
- Hyperparameter tuning with temporal shifts: altering the timing of the song by speeding or slowing down the music. Temporal shifts almost doubled the reward return. This can be explained by how slowing the song by a factor of two allows our policy to spend more time per note, capturing keyboard and note representations.

Performance per Model / Task

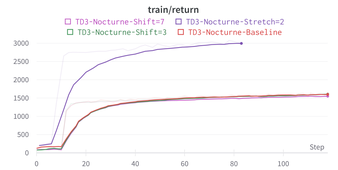


(a) SAC - C Major Scale

(b) SAC - Rousseau Nocturne



(c) TD3 - C Major Scale



(d) TD3 - Rousseau Nocturne

Key Findings

- TD3 was more effective than SAC, as seen in Figure 1 comparing the baselines.
- Nocturne Rousseau was a more complex task than the C Major scale task, but surprisingly yielded a higher return.
- Adding temporal shifts was more effective, perhaps due to more time spent per note and more capturing of the feature representations; however, adding pitch shifts didn't improve the model performance.

Current Steps

- Reducing the parameter space by modifying the key representations from 88 to 36 keys for faster training. Applying relative key representation (for instance using a key representation for 10 fingers instead and incorporating the relative distance of the key from the hand).

Future Directions

- Multitask behavior cloning by using demonstrations and training policy on a variety of songs, which could help them learn shared features such as finger/hand positioning and movement patterns. This could potentially be used to generalize the learned policy on songs not explicitly trained on but with similar underlying features.

References

MuJoCo Menagerie Contributors. 2022. MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo.

Kevin Zakka, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, and Pieter Abbeel. 2023. Robopianist: A benchmark for high-dimensional robot control.

pankhuri@stanford.edu, shelly23@stanford.edu and yokongf@stanford.edu