Extended Abstract

Motivation Human pianists rarely begin by practicing full musical passages. They practice scales, coordination drills, and simpler fragments before playing harder pieces. This project asks whether that intuition transfers to reinforcement learning agents in RoboPianist, a dexterous piano-playing benchmark that combines high-dimensional MuJoCo control with precise musical timing. The original hypothesis was that a human-inspired easy-to-hard curriculum would improve learning on a short two-hand chord progression. The final result is more nuanced: direct target training is the best specialist, while curricula learn broader component skills, and replay-buffer retention largely determines whether staged curricula help or forget.

Method I trained Soft Actor-Critic (SAC) agents on RoboPianist debug-suite tasks under matched 200k environment-step budgets. The target task was `CMajorChordProgressionTwoHands`. I compared four schedules: direct target training, uniform random task sampling, a human-inspired sequence from one-hand scales to a two-hand scale to the chord target, and a reverse schedule that starts with the target. Policies were evaluated with RoboPianist-style key precision, recall, and F1 on the target, an in-curriculum two-hand scale, and a held-out song. I then varied replay capacity across 50k, 100k, and 200k transitions to test whether off-policy replay explains the curriculum behavior.

Results Direct target training remains the strongest target specialist. With 50k replay, direct SAC reaches target F1 0.778, above random curriculum 0.523 and human curriculum 0.445. However, curricula learn component skills that direct target training never sees: under full replay, random and human curricula reach F1 0.453 and 0.413 on the two-hand scale, compared with 0.010 for direct training. The replay ablation reveals the main mechanism. Human target-last curriculum improves from 0.106 with full replay to 0.445 with 50k replay, while reverse target-first curriculum drops from 0.323 to 0.052. Smaller replay acts like target-focused fine-tuning when the target comes last, but causes forgetting when the target was practiced early.

Discussion and conclusion The final conclusion is a specialist-versus-generalist tradeoff. If the objective is a single target phrase, direct training is best. If the objective includes component skill coverage, curricula help, especially random sampling. Human ordering alone is not sufficient; off-policy curriculum design must jointly choose task order and replay retention. The current experiments do not show strong held-out song transfer, so the work should be read as a controlled analysis of curriculum and replay failure modes in RoboPianist rather than a final piano-playing system.

Replay-Aware Curriculum Learning for RoboPianist

Shekhar Sharma
Stanford University
shekhars@stanford.edu

Abstract

Curriculum learning is natural for human musical practice, but it is unclear whether human-like task orderings help off-policy reinforcement learning agents. I study this question in RoboPianist, a dexterous manipulation benchmark in which simulated hands must play piano notes with spatial and temporal precision. Using SAC, I compare direct target training, uniform random task sampling, a human-inspired easy-to-hard schedule, and a reverse-order ablation on short debug-suite RoboPianist tasks. Direct target training achieves the best target chord-progression F1, while curriculum agents learn component tasks such as two-hand scales that direct training does not acquire. A replay capacity ablation shows that this tradeoff is mediated by replay retention: smaller replay improves target-last fine-tuning for the human curriculum, but destroys target-first reverse performance by forgetting early target data. These results suggest that RoboPianist curricula must be designed jointly with off-policy replay and fine-tuning, rather than treated as a fixed easy-to-hard list of tasks.

1 Introduction

Dexterous piano playing is a useful stress test for reinforcement learning because it combines multi-finger manipulation, contact-rich physics, timing, and sparse musical success criteria. RoboPianist [Zakka et al., 2023] turns this into a simulated control problem: an agent controls anthropomorphic hands at a piano and is rewarded for matching a MIDI score. This project asks whether curriculum learning, inspired by how humans practice music, improves learning in this setting.

The starting intuition was simple. Human pianists often learn scales and drills before playing full phrases. I therefore expected an easy-to-hard curriculum to improve sample efficiency on a two-hand chord progression. The experiments contradict the simple form of that hypothesis. Direct target training remains the best target specialist under a fixed 200k-step budget. However, curricula learn broader component skills and their target performance changes substantially with replay capacity. This shifts the research question from "is a human curriculum better?" to "when does a curriculum help an off-policy dexterous control agent?"

This report makes three contributions. First, I build a reproducible RoboPianist curriculum harness that runs four SAC conditions in parallel on a headless 4xA100 pod and logs train/evaluation traces. Second, I compare direct, random, human-inspired, and reverse curricula under matched environment-step budgets. Third, I show that replay capacity is a key mechanism: target-last curricula benefit from smaller replay buffers, whereas target-first curricula require larger replay buffers to avoid forgetting.

2 Related Work

Curriculum learning was popularized as a way to order examples or tasks so that learning begins with easier instances and progresses toward harder ones [Bengio et al., 2009]. In reinforcement

learning, the design space is broader because a curriculum can vary initial states, goals, environments, rewards, or sampling probabilities [Narvekar et al., 2020]. This project focuses on task-order curricula: the agent sees different RoboPianist debug tasks over time.

The learning algorithm is Soft Actor-Critic (SAC), an off-policy actor-critic method that optimizes a maximum-entropy objective while reusing past transitions from replay [Haarnoja et al., 2018]. Replay is usually treated as a sample-efficiency mechanism, but in non-stationary or multi-task training it also determines which tasks remain represented in the update distribution. This makes replay relevant to curriculum design, not merely an implementation detail. Prioritized replay [Schaul et al., 2016] studies which transitions are sampled from replay; here I study how much old curriculum experience remains available at all.

RoboPianist provides the robotics domain and evaluation metrics [Zakka et al., 2023]. It builds on MuJoCo physics [Todorov et al., 2012] and evaluates policies with musical key-activation metrics. I use the debug task suite rather than the full repertoire to keep the final project compute bounded and to enable controlled ablations.

3 Method

3.1 Tasks and Metrics

The target task is `CMajorChordProgressionTwoHands`, a short two-hand chord progression. The main component task is `CMajorScaleTwoHands`, a two-hand C-major scale. I also evaluate on the held-out `TwinkleTwinkleLittleStar` task to test whether component-skill breadth transfers to an unseen song.

I report mean reward and MIDI-style key precision, recall, and F1. Precision measures how often pressed keys are correct, recall measures how many target notes are actually played, and F1 summarizes the two. This distinction matters because many policies learn to avoid wrong notes but omit target notes; such policies can have high precision and low musical completeness.

3.2 Training Schedules

All main runs use the same total training budget of 200k environment steps. I compare:

- **Direct**: train only on the target chord progression.
- **Random**: uniformly sample among four debug tasks.
- **Human**: C scale one-hand, D scale one-hand, C scale two-hand, then the target chord progression.
- **Reverse**: target chord progression first, followed by the scale tasks in reverse order.

The staged human and reverse curricula allocate approximately 50k steps to each of four tasks. Random curriculum receives a similar expected target exposure but interleaves tasks instead of using contiguous stages.

3.3 Replay Ablation

Because SAC is off-policy, the current task is not the only task used for learning: updates are sampled from a replay buffer. I therefore run the 200k-step matrix with replay capacities of 50k, 100k, and 200k transitions. A smaller replay buffer emphasizes recent tasks; a larger buffer preserves transitions from earlier curriculum stages. This ablation tests whether failures come from task ordering alone or from interaction between task ordering and replay retention.

4 Experimental Setup

Experiments use the RoboPianist debug suite with reduced action space, gravity compensation, control timestep 0.05, and n-step lookahead 10. SAC uses two hidden layers of width 128, batch size 128, discount 0.8, critic layer normalization, and critic dropout 0.01. Each run begins with 1000 warmstart steps and evaluates every 20k training steps. Final evaluation uses 5 episodes per seed.

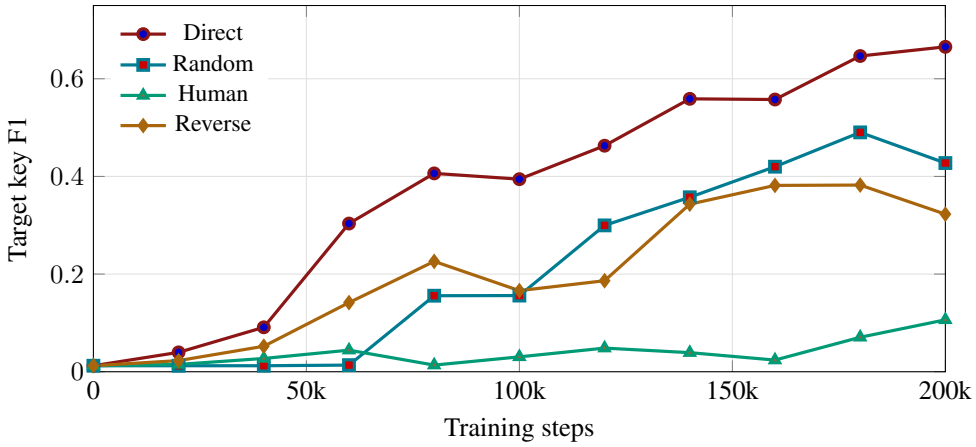

Figure 1: Target chord-progression F1 during 200k-step training with full 200k replay capacity. Curves show mean F1 across four seeds at each evaluation checkpoint.

Table 1: Final key F1 under full 200k replay. Curricula learn trained component skills that direct target training does not learn, but held-out song transfer remains weak.

Evaluation task	Direct	Random	Human	Reverse
Target chord	0.665	0.427	0.106	0.323
C-scale 2H	0.010	0.453	0.413	0.268
Held-out song	0.086	0.037	0.044	0.056

The main 200k replay and 50k replay runs use four seeds; the 100k replay midpoint also uses four seeds after the final overnight run. The 50k-total baseline uses two seeds.

All jobs ran on a headless 4xA100 pod using MuJoCo EGL rendering. I patched the RoboPianist environment setup by pinning a compatible MuJoCo version, initializing ShadowHand assets, avoiding audio-only dependencies during training, and removing a NumPy-2-incompatible wrapper from evaluation. These setup changes do not modify the task reward or the key-activation metrics.

5 Results

5.1 Direct Training Wins Target Mastery

Figure 1 shows target F1 during the full-replay 200k comparison. Direct training rises fastest and finishes highest. The strongest non-direct curriculum under full replay is random sampling, while human ordering performs poorly on the target.

5.2 Curricula Learn Component Skills

Direct training is a strong specialist but a poor generalist. Table 1 shows that direct target training nearly fails the two-hand scale under full replay (F1 0.010), while random and human curricula achieve 0.453 and 0.413. However, none of the methods transfers strongly to the held-out song: the best held-out F1 is only 0.086 in the full-replay condition.

5.3 Replay Capacity Controls the Curriculum Tradeoff

The central result is the replay-capacity ablation in Figure 2 and Table 2. Reducing replay capacity improves direct, random, and human target performance, but catastrophically hurts reverse. Human target-last curriculum rises from 0.106 with full replay to 0.445 with 50k replay. Reverse target-first curriculum moves in the opposite direction, falling from 0.323 to 0.052.

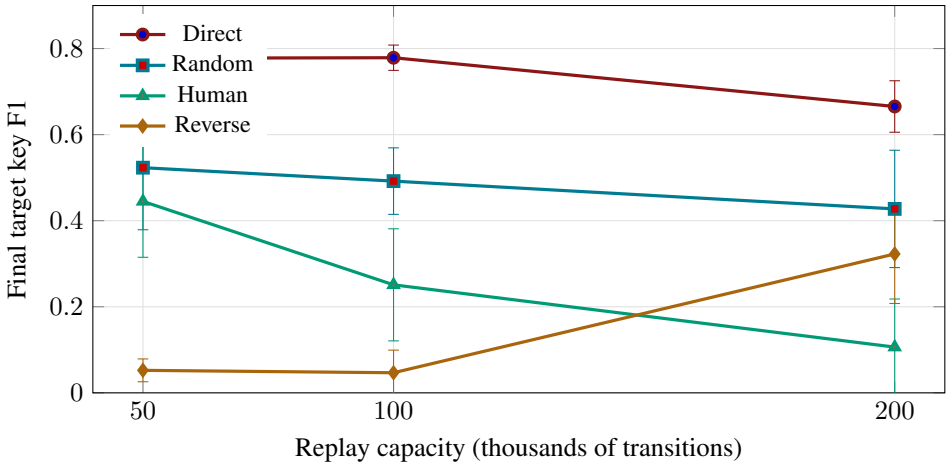

Figure 2: Replay capacity mediates the effect of curriculum order. Points show mean target F1 across four seeds; error bars show one standard deviation. Smaller replay emphasizes recent tasks, helping target-last human curriculum but forgetting the target in the reverse curriculum.

Table 2: Final target key F1 as replay capacity changes. All rows use 200k total training steps and four seeds.

Replay	Direct	Random	Human	Reverse	Seeds
50k	0.778 ± 0.091	0.523 ± 0.144	0.445 ± 0.130	0.052 ± 0.027	4
100k	0.779 ± 0.029	0.492 ± 0.077	0.251 ± 0.130	0.047 ± 0.052	4
200k	0.665 ± 0.060	0.427 ± 0.136	0.106 ± 0.112	0.323 ± 0.115	4

The interpretation is replay recency. For the human schedule, the target task comes last, so a smaller buffer behaves like target-focused fine-tuning. For the reverse schedule, the target task comes first, so the same smaller buffer flushes target experience before the final policy is evaluated. Random sampling is comparatively robust because all tasks continue to appear throughout training.

5.4 Target Exposure Efficiency

The results also separate target exposure from total environment budget. Table 3 compares direct target-only training for 50k steps with curricula that receive only about 50k target steps but 150k additional component-task steps. Random and human curricula with 50k replay exceed the 50k direct baseline, suggesting that component practice can improve target-exposure efficiency. Nevertheless, direct target training with the full 200k total budget remains the best target specialist.

6 Discussion

The original hypothesis was that human-style easy-to-hard ordering would directly improve RoboPianist learning. The experiments only partially support this. Human ordering is not enough under full replay, where old scale data remains mixed with final target data. However, the same order becomes much stronger with smaller replay capacity, which emphasizes recent target transitions. This suggests that curriculum design for off-policy RL should specify not only the order of tasks but also how old task data is retained, downweighted, or discarded.

The results also highlight a precision-recall failure mode. Most learned policies have high key precision, often above 0.95 on the target, but differ substantially in recall. Musically, this means policies avoid many wrong notes but still omit parts of the phrase. F1 and recall are therefore more informative than reward or precision alone.

There are several limitations. First, the experiments use RoboPianist debug tasks rather than the full 150-piece repertoire, so the results should not be interpreted as broad musical competence. Second,

Table 3: Target-exposure comparison on the chord task. Curricula with 200k total steps receive roughly 50k target steps but additional component-task experience.

Setting	Target F1	Target recall	Seeds
Direct, 50k total steps	0.265 ± 0.150	0.215	2
Random curriculum, 200k total / 50k replay	0.523 ± 0.144	0.435	4
Human curriculum, 200k total / 50k replay	0.445 ± 0.130	0.354	4
Direct, 200k total / 50k replay	0.778 ± 0.091	0.731	4

the held-out `TwinkleTwinkleLittleStar` evaluation remains weak, so component-task breadth did not yet become repertoire-level generalization. Third, the study is limited to SAC and a small set of replay capacities. More seeds, larger tasks, and explicit pretrain-then-fine-tune baselines would be natural next steps.

7 Conclusion

Direct target training is the best way to specialize on the RoboPianist chord target under the budgets tested here. Curricula, especially random interleaving, learn broader component skills. The main lesson is that replay capacity controls how staged curricula behave in off-policy learning: recent replay helps target-last fine-tuning, while full replay preserves early target knowledge. Future RoboPianist curricula should therefore jointly design task order, replay retention, and fine-tuning rather than relying on a human-inspired order alone.

8 Team Contributions

This was an individual project. Shekhar Sharma designed the project, implemented and debugged the RoboPianist setup, built the experiment launcher and summarization tools, ran all experiments, analyzed the results, and wrote the report.

9 AI Tools Disclosure

I used AI tools as an assistant for environment setup, shell-script debugging, LaTeX editing, and wording revisions. In particular, AI tools helped diagnose dependency issues on the headless GPU pod, debug experiment scripts, clean and organize scripts across experiments and organize the final report.

Changes from Proposal The proposal included an RL-generated adaptive curriculum as an ambitious goal. During implementation, I narrowed the project to a controlled comparison of direct, random, human-inspired, and reverse curricula, plus a replay-capacity ablation. This narrower scope produced a clearer final result: the key issue is replay-aware curriculum design, not simply whether a human ordering is better than random sampling.

References

Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48. ACM, 2009.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 2018.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

Kevin Zakka, Philipp Wu, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, and Pieter Abbeel. Robopianist: Dexterous piano playing with deep reinforcement learning. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2975–2994. PMLR, 2023.

A Additional Implementation Details

Each run writes a JSON configuration, JSONL event trace, training metrics CSV, and evaluation metrics CSV. The parallel launcher assigns one condition per GPU for each seed and then aggregates final rows into summary CSV files. The final-report tables and plot coordinates are generated from those CSV files by `build_report_data.py`.

B Additional Results

With 50k replay, cross-task F1 changes as follows:

Evaluation task	Direct	Random	Human	Reverse
Target chord	0.778	0.523	0.445	0.052
C-scale 2H	0.007	0.439	0.153	0.176
Held-out song	0.041	0.045	0.037	0.039