Extended Abstract

Motivation Robotic piano-playing is a challenging task that requires high levels of spatial and temporal coordination. Prior work has demonstrated how a virtual robot can be trained using soft actor-critic methods to play a variety of short pieces. However, it remains unknown how the skills acquired in one piece may be transferred to another, potentially accelerating the learning process. In this paper, we examine how a variety of pretraining curricula drawn from typical human exercises impact training. We also explore how adding a bonus for correctly timed key presses affects the accuracy of the model.

Method This project makes two primary contributions to the existing RoboPianist model. First, we propose a modified reward function that includes a bonus for when a key is pressed close to the expected onset of the note in the score, as opposed to the existing implementation which instead only accounted for the amount of overlapping timesteps that a note was pressed. The reward bonus is Gaussian-shaped, decaying to zero as the timing error increases. We compare the performance of the runs with this onset alignment reward included to runs that use the existing reward function. Second, we explore how structured curricula can be used for pretraining to improve performance. Specifically, we experiment with using scales and arpeggios in the same key as the target piece and in a different key. We perform multiple two-by-two experiments to compare the efficacy of curricula type and key with and without onset reward included. A post-hoc analysis also investigates whether key-matched pretraining benefits transfer more when the target piece is scale-heavy.

Implementation We adopt the model architecture from the original RoboPianist paper by Zakka et al. (2023) which uses DroQ, a model-free RL algorithm with a randomized ensembled double Q-learning with dropout. We use the Chopin Nocturne in E♭ Major as our primary target piece, with post-hoc analyses using Mozart's Sonata in C Major as an alternative. For runs including the curriculum, 100k steps are allotted for pretraining and 400k steps are used for finetuning on the target piece, whereas in non-curriculum runs, 500k steps are used to train on the target piece. In later runs on the Sonata, we use 100k pretraining steps and 900k finetuning steps for a total of one million training steps. Performance is evaluated using the F1 score, a harmonic mean between precision and recall over key activations per timestep. The environment simulates a pair of Shadow Dexterous Hands playing a piano in MuJoCo.

Results Across both target pieces, the onset alignment reward improved performance while curriculum pretraining reduced it in most cases. On the Nocturne, the onset-only condition achieved the best F1 score (0.589), roughly 10% above the baseline, while on the Sonata the baseline and onset-only conditions were equal within noise. Curriculum pretraining alone yielded only a marginal gain on the Nocturne and degraded performance on the Sonata. Combining curriculum with onset reward was the worst configuration for both pieces. Key-matching scale curricula provided no advantage, whereas key-matched arpeggios did improve performance when paired with onset timing. Decomposing F1 into precision and recall revealed that all configurations reached very high precision, so performance differences were driven almost entirely by recall, and curriculum pretraining reduced that recall.

Discussion Our results suggest that curriculum pretraining is beneficial only when the curriculum is relevant to the downstream task. The consistent precision-recall tradeoff indicates that curriculum pretraining results in a conservative key-pressing strategy which maximizes precision at the expense of recall. By contrast, the onset alignment reward helped most on the simpler Nocturne, and was relatively neutral on the more complex Sonata. Together, these findings point to reward shaping rather than curriculum design as the more promising avenue for improvement in this setting.

Conclusion Our findings demonstrate a method for leveraging skills learned across multiple pieces. First, we show that adding curriculum pretraining reduces model performance, perhaps due to shortcomings in multi-task generalization as referenced by Zakka et al. (2023). Second, we found that adding an onset alignment reward generally improved the F1 score. Future work may run the same experiments but with longer runs, in order to observe how the model behaves at convergence. Future work may also experiment with increasing the number of trainable parameters in the critic networks, which could potentially improve the model's capacity to learn diverse skills which could be applied to a range of pieces.

Scale-Based Curriculum Pretraining for Robotic Piano Performance

Justin Choo
Department of Computer Science
Stanford University
justinchoo@stanford.edu

Eric Martz
Department of Computer Science
Stanford University
emartz@stanford.edu

Anna Fisher Lopez
Department of Computer Science
Stanford University
afishpez@stanford.edu

Abstract

Robotic piano playing is a challenging form of dexterous hand manipulation which requires precise coordination across many degrees of freedom. Prior work has demonstrated how a virtual robot can be trained using RL to learn a variety of target piano pieces, but it remains unknown how curriculum-based learning might impact learning in this setting. In this project, we examine the effects of pretraining an actor model with two forms of curricula (scales and arpeggios) on its performance on two target pieces (Chopin's Nocturne in E♭-Major and Mozart's Piano Sonata No. 16 in C Major). We also examine how the addition of a key onset alignment reward impacts the accuracy of the model. We find that the onset alignment reward boosts F1 score for most conditions, while pretraining on a curriculum of scales or arpeggios reduces performance. Future work may increase the number of trainable parameters in the critics and execute longer training runs in order to examine behaviors at model convergence.

1 Introduction

Dexterous manipulation remains one of the most difficult tasks in reinforcement learning (RL) due to coordination across many degrees of freedom, large search spaces, and the need for spatial and temporal precision. Piano-playing is a task which embodies all of these challenges, and therefore serves as a strong benchmark for dexterous manipulation.

Zakka et al. (2023) introduced RoboPianist, an end-to-end system for training a virtual robot to play piano pieces via deep RL. They designed a simulated environment, including an 88-key piano with spring weights and a pedal, as well as two robotic hands drawn from the MuJoCo Menagerie. The model was individually trained on 150 pieces and exhibited variety of emergent capabilities such as simultaneous coordination of both hands, playing chords, and playing trills. However, when trained on 16 of these pieces simultaneously, the F1 score of the model dropped to "almost 0." This raises the question of how a model might be trained to transfer learning across multiple pieces, that is, how prior experience on one set of pieces might improve model performance on another set.

In this project, we explore a subset of this question: how can pretraining a model on a human-inspired piano curriculum affect downstream performance on a target piece? Human students are often taught a variety of basic skills, such as scales, arpeggios, and etudes, in order to build up a level of technical competency that can be applied toward more difficult pieces. Here, we present the model with a

variety of pretraining conditions, including scales in various keys, as well as arpeggios, to determine what kind of pretraining scheme is maximally beneficial for downstream model performance.

We also explore how the reward function for the model might be modified in order to generate more realistic playing. While the original paper uses F1 score as a reward function, we noticed that performance on some pieces sounded jumbled, or out of time. Although the model generally plays the correct keys at roughly the right time, it often presses those keys slightly too early or too late. In piano playing, because each note is most audible at its onset, it is critical for the start of each note to be in rhythm. In fact, it is arguably more important that each key is pressed down at the correct time than sustained for its exact duration in order to produce a qualitatively convincing performance. Therefore, we experiment with adding a bonus to the reward function for every key pressed at the correct time, in order to observe how this incentive might affect downstream performance.

2 Related Work

Zakka et al. (2023) incorporated a number of design choices that were crucial to the success of their model. They demonstrate that assigning specific finger labels to each key press substantially reduces the exploration burden, which they find is necessary for generating a successful policy. They also underactuate their hand models, which further helps to constrain the action space and accelerates learning. For their critic, they use a DroQ model (randomized ensemble Double-Q learning, with dropout) which exhibits relatively stable learning over complex reward structure (see Methods section for more details). They also modify the reward function to encourage fingers to be spatially close to their target keys and to minimize energy expenditure, with the goal of smoothing out erratic behaviors.

Our investigation into the impact of curriculum pretraining arises from the field of curriculum learning, introduced by Bengio et al. (2009) and extended to RL by Portelas et al. (2020), which establishes that the order and difficulty progression of training samples meaningfully affects final performance. Across a range of machine learning tasks, presenting easier examples first consistently improves both generalization and convergence speed. More recent work has shown that teaching a model tasks within its "Zone of Proximal Development," i.e. tasks that are not too easy and not too hard, can accelerate training across a variety of tasks (Tzannetos et al., 2023).

3 Methods

3.1 Reward Functions

The original paper defines its reward as a weighted sum of five components: key-press, fingering, forearm collision penalty, sustain-pedal usage, and energy penalty.

Table 1: Reward function components from the original RoboPianist task.

Term	Description	Range	Threshold
r_{key}	Press correct keys; penalise false positives	$[0, 1]$	$\delta_k = 0.05$
$r_{\text{fingering}}$	Fingertip distance to target key	$[0, 1]$	$\delta_f = 0.01$ m
r_{forearm}	No collision between forearm geoms	$\{0, 0.5\}$	—
r_{sustain}	Sustain pedal matches score	$[0, 1]$	$\delta_k = 0.05$
r_{energy}	Penalise actuator power ($\lambda = 5 \times 10^{-3}$)	$(-\infty, 0]$	—
r	Sum of all active terms	up to ≈ 3.5	

Interestingly, the key-press reward component rewards whether the correct key is pressed at the given timestep, but it does not take into account when the note should begin. In other words, a key pressed in the middle of a long sustained note would receive the same key-press reward as a key pressed at the beginning, as long as they were held for the same number of timesteps.

In real piano playing, the onset of the key press is the most important for melodic accuracy and interpretability. As a result, we modify the existing reward function to incentivize the key being pressed at the beginning of the note.

To do so, we add an onset alignment bonus r_{onset} to the reward:

$$r_{\text{onset}} = \frac{1}{|O_t|} \sum_{k \in O_t} \exp\left(-\frac{\Delta t_k^2}{2\sigma^2}\right) \tag{1}$$

where O_t is the set of keys newly pressed at step t (i.e. transitions from inactive to active), Δt_k is the distance in timesteps between step t and the nearest expected onset for key k in the MIDI score, and $\sigma = 2$ steps is the temporal tolerance. The bonus lies in $[0, 1]$ and is weighted by a scalar α, giving a modified total reward $r' = r + \alpha \cdot r_{\text{onset}}$. We also experiment with different weightings of the onset alignment bonus which can be found in Table 7.

3.2 Structured Curricula

The original paper revealed that pretraining on many pieces actually worsened performance on a target piece, which leads us to wonder what pretraining data can be better transferred. We hypothesize that scales and arpeggios, similarly to human learning, can provide a general baseline dexterity that is more widely applicable and simpler than training on other pieces. As a result, we experiment with two types of pretrain curricula, scales and arpeggios, each in the same key as the target piece and in a different key than the target piece. The goal is to better understand what matters in transfer learning.

3.3 Experimental Conditions

In order to understand which types of curricula transfer best, we run multiple two by two comparisons including a combination of factors. We examine whether including the onset alignment reward improves performance compared to the original reward function. We also vary our curricula by type (scales vs. arpeggios) and key (C/D major vs. E♭/F major).

A summary of all experimental conditions can be found below:

Table 2: Experimental conditions. The target piece (Nocturne in E♭ major) is held fixed across all conditions. "Same key" pretraining uses E♭/F major; "different key" uses C/D major. All curriculum runs use 100k pretrain + 400k finetune steps; no-curriculum runs use 500k steps direct. $\alpha = 0.1$, $\sigma = 2$ steps where onset reward is active.

		Onset Alignment Reward	
Pretraining	**Key match**	**No** ($\alpha = 0$)	**Yes** ($\alpha = 0.1$)
None (baseline)	—	`baseline`	`onset-only`
Scales	Different (C/D)	`curriculum`	`curriculum+onset`
Scales	Same (E♭/F)	`eb-curriculum`	`eb-curriculum+onset`
Arpeggios	Different (C/D)	`arpeggio-cd`	`arpeggio-cd+onset`
Arpeggios	Same (E♭/F)	`arpeggio-eb`	`arpeggio-eb+onset`

Lastly, our post-hoc analysis examines whether a different piece affects the performance of the scale curriculum. We examine the Mozart Sonata in C Major to test whether a piece that includes more scales performs better.

4 Experimental Setup

4.1 Environment and simulation

We use the RoboPianist environment (Zakka et al., 2023), built on MuJoCo, where a pair of Shadow Dexterous Hands must press piano keys to match a MIDI score. The simulation runs at a control timestep of 0.05 seconds. The agent observes the current and upcoming 10 timesteps of the MIDI score as lookahead, as well as joint positions and velocities, fingertip positions, and key activation states. The primary target piece is Chopin's Nocturne in E♭ major, and we use a Mozart Sonata in C major for post-hoc analysis.

4.2 Model architecture

We adopt Zakka et al. (2023)'s open-source code for our policy optimizer: DroQ, a model-free RL algorithm that uses a randomized ensembled double Q-learning with dropout. This is a variant of SAC where regularization is applied to the critic network. The actor and critic each use a three-layer MLP with hidden dimensions $(256, 256, 256)$ and GeLU activations. Full hyperparameters are reported in Table 9.

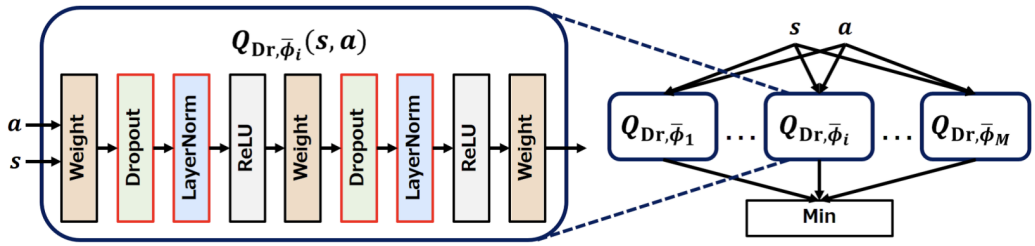

Figure 1: DroQ critic architecture. Each Q-network applies alternating dropout and layer normalization after every linear layer, and then the minimum is taken across critics to reduce overestimation bias.

4.3 Training Procedure

For the runs with curriculum learned, we trained the agent on 100k steps in a scale or arpeggio environment, followed by 400k steps of finetuning on the target piece for a total of 500k steps. For the runs without curriculum learning, all 500k steps were used to train on the target piece. For Sonata runs which were run for a total of one million steps, curriculum runs were trained on 100k of an arpeggio curriculum and then had 900k finetuning steps. All runs were executed on Modal cloud infrastructure using NVIDIA A10G GPUs or T4 GPUs.

4.4 Evaluation

The agent is evaluated every 10,000 training steps for one episode. Performance is reported as the F1 score over key activations, which is the harmonic mean of precision (fraction of pressed keys that should be pressed) and recall (fraction of required keys that are pressed) averaged across all timesteps in the episode. F1 ranges from 0 to 1, with 1 indicating perfect note accuracy. We report the final F1 at 500k steps as our primary metric.

4.5 Reproducibility

All runs use a fixed random seed of 42. We acknowledge the constraint of using only one seed for all runs, as it limits the conclusions we can draw from the results. Due to preemption challenges with these runs that lasted up to 20 hours, we also add checkpoints at intervals throughout the training process to allow for a run to be resumed midway through its training if it was preempted by another process.

5 Experiments & Results

5.1 Impacts of Basic Curriculum and Onset Timing

Across both target pieces, the onset alignment reward improved performance and the curriculum reduced performance. This contradicts our hypothesis that scale and arpeggio pretraining would transfer general knowledge and dexterity to the target piece in the manner of human practice; however, it aligns with the results from Zakka et al. (2023) which showed that pretraining the agent on 16 other piano pieces reduced F_1 scores to nearly zero.

In the Nocturne, we found that the best performance over the baseline was the model with onset timing but no curriculum, yielding an F_1 score of 0.589 which is about 10% better than the baseline.

Table 3: F_1 scores for Nocturne and Sonata across experiments. Note that for Nocturne, the curriculum is scales, and for Sonata the curriculum is arpeggios.

Method	**Nocturne F_1 (500k steps)**	**Sonata F_1 (1M steps)**
Baseline	0.533	**0.627**
Curriculum Only	0.547	0.600
Onset Only	**0.589**	0.622
Onset + Curriculum	0.477	0.593

On the Sonata, the baseline actually performed the best, although its F_1 score is within 0.004 of that of the onset timing only F_1 score, making it equal to the onset timing within noise-induced errors. We believe this difference is due to the different structure of the two pieces: the Nocturne's sustained melodic lines depend on precise note onsets for rhythmic clarity, whereas the Sonata is faster, with complicated scalar passages that constrain timing because of their complexity which leaves less room for the onset bonus to improve alignment.

Curriculum pretraining alone yielded a marginal gain against the baseline on the Nocturne (F_1 scores of 0.547 vs. 0.533) but degraded performance on the Sonata (F_1 scores of 0.600 vs. 0.627). Combining curriculum with the onset reward was the worst configuration for both pieces (a 10% and 5.4% reduction from the baseline for Nocturne and Sonata respectively), indicating that the two interventions interfere rather than compound. We hypothesize that the general reduction in performance from curriculum pretraining comes from the policy being too small, and the curriculum biasing the policy toward behaviors that do not generalize to the target piece. In an attempt to mitigate this, we trained Nocturne on a curriculum matching the key of the piece.

5.2 Key-matching Curriculum Pretraining Performance

Training the agent playing Nocturne on a curriculum of the E♭/F scales instead of the C and D scales was not completely conclusive, but we find that it does not improve performance; however, training on a E♭/F arpeggios does improve performance when combined with onset timing.

Table 4: F_1 scores for Nocturne after retraining on E♭ and F scales

Curriculum (Scales)	**Onset F_1**	**No Onset F_1**
C/D Scales	0.477	**0.547**
E♭/F Scale	0.485	0.492

Table 5: F_1 scores for Nocturne after retraining on arpeggio curricula.

Curriculum (Arpeggios)	**Onset F_1**	**No Onset F_1**
C/D Arpeggios	0.533	0.519
E♭/F Arpeggios	**0.572**	0.465

Comparing the two scale-based curricula on the Nocturne (see table 4), we find that without the onset reward, different-key scales outperformed same-key scales (0.547 vs. 0.492), contradicting our expectation that pretraining in the target key would transfer most effectively. With the onset reward active, the two were comparable (0.477 vs. 0.485). This suggests that key-matching scale pretraining has no advantage, and might even be a disadvantage because it trains the agent on an unrelated objective to the piece (scales vs. playing Nocturne) using notes which appear often in the piece instead of notes that the agent is less likely to encounter in the piece, thereby increasing ambiguity about what is supposed to be done with those specific notes.

Arpeggios, on the other hand, do yield a performance improvement when key-matched, though they still don't outperform the best F_1 of onset timing and no curriculum. We hypothesize that this is because arpeggios contain a wider structure that is more similar to what an agent would encounter in

a piece like Nocturne, whose melodies more similarly resemble arpeggios than scales.

5.3 Comparing Precision and Recall

Table 6: Performance comparison for Sonata across different model configurations.

Method	F_1 Score	Precision	Recall
Baseline	**0.627**	0.985	**0.569**
Onset Only	0.622	0.985	0.561
Curriculum Only	0.600	0.992	0.541
Curriculum + Onset	0.593	**0.999**	0.534

Decomposing the Sonata F1 scores into precision and recall reveals that all configurations achieved very high precision (0.985–0.999), meaning performance differences were driven almost entirely by recall. Curriculum + Onset attained the highest precision (0.999) but the lowest recall (0.534), while the baseline had the lowest precision (0.985) and the highest recall (0.569). In other words, curriculum pretraining produces a more conservative policy that presses fewer keys but is more correct when it does press a key, at the cost of missing more notes. This precision–recall tradeoff is the most consistent signature of curriculum pretraining in our experiments, and can be seen more clearly in 2.

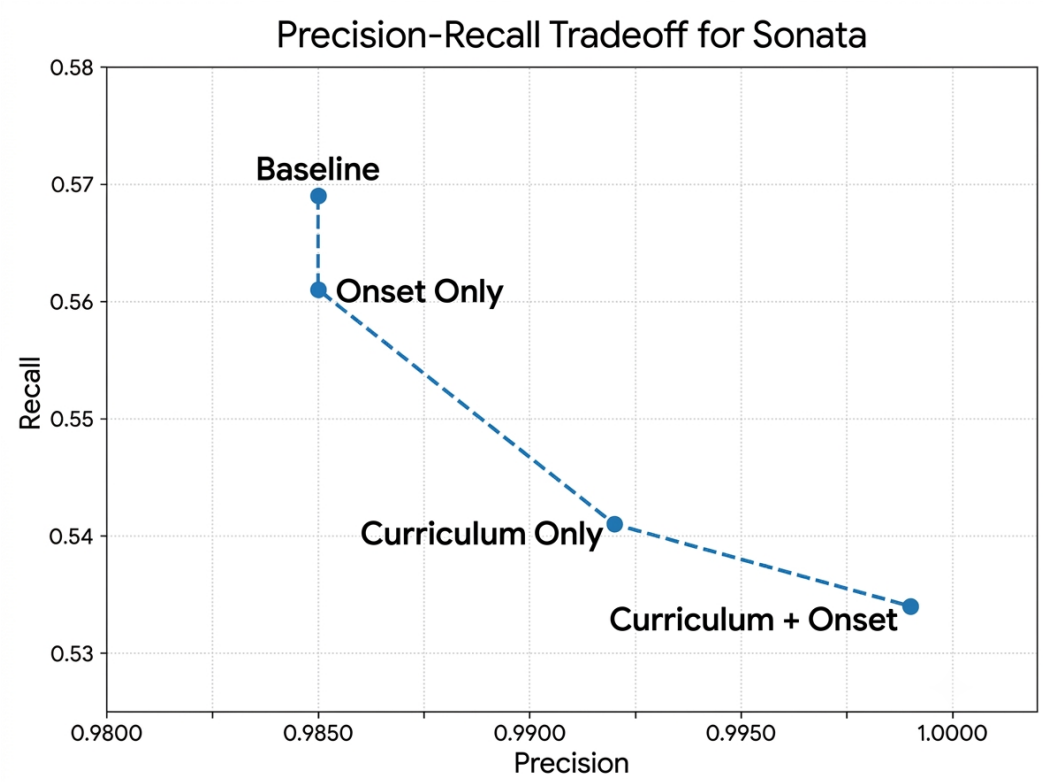

Figure 2: Precision (how correct are pressed keys) increases and Recall (how many correct keys are pressed) decreases with curriculum and training.

5.4 Qualitative Analysis

Listening to the performance videos gives us expected quantitative trends. Models trained with the onset reward produce playing that is more rhythmically accurate, with note onsets falling closer to

their intended positions in the score. Without onset timing, the piece feels more awkward, with the model pressing correct keys but slightly early or late. The conservative behavior of the curriculum-pretrained models was also audible: these policies tended to skip notes in more complex passages rather than risk incorrect presses, consistent with their high precision and reduced recall.

6 Discussion

Our work contains two main findings. First, we observed that pretraining on a structured curriculum reduces performance, regardless of the key of the curriculum or whether it is scales or arpeggios, except for the case of key-matched arpeggios. The precision–recall tradeoff that comes from curriculum training shows that curriculum pretraining appears to instill a conservative key-pressing strategy that maximizes precision at the expense of recall. This mirrors the original RoboPianist observation that multi-piece pretraining degraded single-piece performance, and suggests that the difficulty of transfer in this setting is not specific to full pieces but extends even to simpler scale and arpeggio curricula.

Second, we found that adding an onset alignment reward, which incentivizes keys to be pressed at the correct times, improved the F1 score. Its benefit on the Nocturne and its relative neutrality on the Sonata suggests that temporal rewards help most when a piece's melody is sparser, and is less effective when note density already enforces timing.

These findings resolve gaps in prior literature. While the authors of the original RoboPianist paper found that simultaneously training a model on several pieces at once yielded poor performance, we show that segmenting the learning process into pretraining and post-training phases with simple curricula sometimes only minimally reduce model performance (Zakka et al., 2023). We also show that the choice of pretraining curriculum is critical in setting the foundation for future learning, suggesting that both the selection, and the order of acquisition, of foundational skills are important. Finally, we introduce a new component of the reward structure, onset alignment, that can be used to accelerate learning.

7 Conclusion

We investigated whether pretraining a robotic piano policy on scale and arpeggio curricula improves downstream performance on target pieces, and whether an onset alignment reward yields more rhythmically accurate playing. We found that curriculum pretraining did not transfer as hypothesized and produces a more conservative, high-precision and low-recall policy that underperformed direct training. Onset reward improved performance on a piece with clear melodic lines like Nocturne, but effects were more modest in a more complex piece like Sonata. These results show that transfer difficulty in dexterous piano playing persists even for simple, human-inspired curricula. Addionally, we identify reward shaping rather than curriculum design as a more promising opportunity for improvement in this setting.

Future work may run the same experiments but with longer runs, in order to observe how the model behaves at convergence. Training curves suggest that the models were still improving when we ended training, so it is unclear what optimal model performance would look like under this training scheme. Future work may also experiment with increasing the number of trainable parameters in the critic networks, which could potentially improve the model's capacity to learn diverse skills which could be applied to a range of pieces.

8 Team Contributions

- **Anna:** Designed the onset alignment reward, implemented it, and conducted the parameter sweep.
- **Eric:** Designed the scales curriculum, and ran the two experiments associated with it (100K + 400K steps vs. 100K + 500K steps).
- **Justin:** Designed the arpeggios curriculum and added the second target piece, and ran those experiments.

All three members contributed equally to the writing and editing of the final report.

Changes from Proposal We made several additions to our project outline that expand upon our original proposal. First, we decided to test pretraining the model on scales in different keys. Observing that pretraining on C/D-Major scales did not improve performance on the target piece, we hypothesized that pretraining on a scale in the key of the piece might be more beneficial. Second, we pretrained the model on arpeggios. Observing later than no scale improved downstream performance, we thought that pretraining on arpeggios, which more closely reflect the skills required by the piece, would be useful. Third, as a sanity check, we tested the model on a second target piece (the Mozart Sonata) which contains more scale-like passages than the Nocturne, and so we were curious whether the scale pretraining would have more impact here. Finally, we conducted a hyperparameter sweep of the onset alignment reward.

We eliminated a few aspects of our proposal as well. Initially, we suggested training the model on an unstructured curriculum (i.e., pretraining on a random piece), but because the original paper conducted a similar experiment, and because each training run was quite expensive, we decided to eliminate this experiment. We also had planned to test using audio embeddings as a reward function. However, we realized that the model would not receive enough concrete, short-term rewards in this paradigm, and therefore that learning would be quite difficult, if not impossible. We also decided not to pursue teaching the model to learn fingerings on its own. As we started training the model, we observed how large the search space was, and we recognized that learning fingerings from scratch would be an intractable problem.

9 AI Disclosure

Claude Code was used for debugging purposes, especially with Modal entrypoints, library dependency resolution, and checkpointing to prevent runs from being preempted. Claude was also used to aid in table generation and design for this report, such as when reporting the full hyperparameters in an appropriate format.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML '09)*. Association for Computing Machinery, New York, NY, USA, 41–48. doi:10.1145/1553374.1553380

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Automatic Curriculum Learning For Deep RL: A Short Survey. arXiv:2003.04664 [cs.LG] https://arxiv.org/abs/2003.04664

Georgios Tzannetos, Bárbara Gomes Ribeiro, Parameswaran Kamalaruban, and Adish Singla. 2023. Proximal Curriculum for Reinforcement Learning Agents. arXiv:2304.12877 [cs.LG] https://arxiv.org/abs/2304.12877

Kevin Zakka, Philipp Wu, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, and Pieter Abbeel. 2023. RoboPianist: Dexterous Piano Playing with Deep Reinforcement Learning. *Conference on Robot Learning (CoRL) 2023* (2023).

A Implementation Details

Table 7: Onset alignment reward component and α sweep. $\sigma = 2$ steps throughout. $r_{\text{onset}} \in [0, 1]$ is triggered only on key-press transitions (0→1).

α	**Onset bonus range**	**Bonus as % of max base reward**	**Notes**
0.01	$[0,\ 0.01]$	$\approx 0.3\%$	Near-negligible shaping
0.05	$[0,\ 0.05]$	$\approx 1.4\%$	Weak shaping
0.10	$[0,\ 0.10]$	$\approx 2.9\%$	Originally used
0.25	$[0,\ 0.25]$	$\approx 7.1\%$	Moderate shaping
0.50	$[0,\ 0.50]$	$\approx 14.3\%$	Strong onset emphasis

Table 8: α sweep results: final evaluation F1 scores at 500k steps for onset-only conditions (no curriculum) on the Nocturne in E♭ major. Seed 42, $\sigma = 2$ steps.

α	**Onset bonus range**	**F1 (500k steps)**
0.01	$[0,\ 0.01]$	0.505
0.05	$[0,\ 0.05]$	0.556
0.10	$[0,\ 0.10]$	**0.589**
0.25	$[0,\ 0.25]$	0.490
0.50	$[0,\ 0.50]$	0.547

Table 9: Full hyperparameter settings.

Category	**Hyperparameter**	**Value**
SAC / DroQ	Actor learning rate	3×10^{-4}
	Critic learning rate	3×10^{-4}
	Temperature learning rate	3×10^{-4}
	Optimizer	Adam
	Hidden dimensions	$(256, 256, 256)$
	Activation function	GeLU
	Number of Q-networks	2
	Soft update coefficient τ	0.005
	Discount factor γ	0.99
	Initial temperature	1.0
	Target entropy	$-0.5 \times \lvert\mathcal{A}\rvert$
	Backup entropy	True
Replay & Batching	Replay buffer capacity	1,000,000
	Batch size	256
	Pretrain warmstart steps	5,000
	Finetune warmstart steps	1,000
Training Schedule	Pretrain steps	100,000
	Finetune steps	400,000
	Total steps (no curriculum)	500,000
	Scale switch interval	50,000
	Clear replay on finetune	True
Environment	Control timestep	0.05 s
	Lookahead steps	10
	Evaluation interval	10,000 steps
Onset Reward	Weight α (default)	0.1
	Tolerance σ	2 steps (0.1 s)