Extended Abstract

Motivation Every pianist uses curricula to learn. They practice finger exercises before pieces, and they progress from easy repertoire to harder repertoire over the course of years. We ask whether these habits help a Soft Actor-Critic agent control simulated Shadow Hands in RoboPianist (Zakka et al., 2023; Haarnoja et al., 2018), where a MIDI score tells the robot which piano keys to press and when. In the first of three experiments, the agent practices Hanon finger exercises (a standard amongst pianists for hundreds of years) before a piece both using vanilla SAC as well as using tree-based exploration. In the second, the agent follows a hand-designed repertoire sequence ordered by difficulty. In the third, Prioritized Level Replay picks the next piece from the agent's recent learning progress.

Method For the Hanon experiment, we compare a hanon-pretrained agent and a pretrainingless agent on technical one-voiced right handed melodies in and out of sample. Beyond the first simple comparison, we add two controls to the environment. A weight λ_{fing} scales the fingering reward from full strength to zero, and a switch replaces the per-note fingering field in the observation with a constant while keeping the observation shape fixed, so we can load checkpoints from runs that used the field. Training agents from random weights and from Hanon-pretrained weights under these settings lets us tell apart a policy that reads the fingering field from a motor skill the weights store (aka "intuition"). In the second experiment, we construct a hand-designed curriculum by ranking RoboPianist repertoire according to trajectory-derived difficulty features, including polyphony, note density, and hand-motion complexity. The agent trains sequentially on increasingly difficult pieces (*Twinkle Twinkle Little Star* $\rightarrow$ *Reverie* $\rightarrow$ *Clair de Lune* $\rightarrow$ *Maple Leaf Rag*) with checkpoint transfer between stages. In the third experiment, we replace this fixed ordering with Prioritized Level Replay, which adaptively selects pieces according to recent learning progress. We also recover human fingerings from piano video with a five-stage perception pipeline and substitute them for the optimal-transport (OT) fingering.

Implementation All three experiments share the same RoboPianist environment and a general Soft Actor-Critic backbone. We score timing with "note-onset" F1, counting a note correct when the agent strikes it within 50 ms of its target time, and we attribute every key contact to the nearest fingertip so we can read which finger presses each key. Across experiments, we mainly vary the practice material, its ordering, and the reward pacing, allowing us to isolate the effects of curriculum learning (potentially coupled with other exploration techniques) on robotic piano performance.

Results With the fingering reward enabled, the "pretrainingless" agent reaches the same note-onset accuracy as the Hanon-pretrained agent at every tempo we tried, up to roughly twice the written speed. As the fingering reward decays, the pretrained agent gains most of its lead by reading the fingering field, and once we hide that field the two agents differ by about 0.1 in accuracy, inside single-run noise. However, they greatly differ in out-of-sample technique; after we remove the fingering reward, the pretrainingless agent plays the phrase with only two fingers while the pretrained agent keeps using all five, even though both play nearly the same notes accuracy-and-timing-wise. In the hand-designed curriculum involving both hands and more complex pieces, transfer improves note-onset F1 on both *Clair de Lune* and *Maple Leaf Rag*, indicating better timing-sensitive performance than training directly on the target piece. The gains are modest for *Clair de Lune* but become larger on the more difficult *Maple Leaf Rag*, where curriculum transfer also improves overall return and learning efficiency. The adaptive Prioritized Level Replay curriculum raises paired Onset F1 over the no-curriculum baseline on a 75-piece held-out corpus while note selection stays flat, discovering its ordering from the agent's own learning signal rather than a difficulty list. Substituting human fingerings recovered from video for the optimal-transport reward adds a further timing gain concentrated on octave runs and hand crossings.

Discussion In our series of repertoire-curriculum experiments, an agent trained on a hand-designed or adaptive ordering of pieces competes against an agent trained directly on the target piece. The Hanon agent already reaches similar reward by two routes, the five-finger technique that Hanon prescribes or the two-finger play it settles into when nothing rewards the recommended finger. We hypothesize that a five-finger agent would start to win in longer runs that penalize fatigue, yet our runs at this scale stay too short for any such advantage to surface. Across both repertoire tests in Experiment 2, curriculum learning primarily improves Onset F1 rather than note-selection accuracy alone, suggesting that prior repertoire experience transfers timing and coordination skills. These benefits become more pronounced as repertoire difficulty increases, consistent with the way human pianists build technique through progressively harder pieces. Prioritized Level Replay shows the same split without any difficulty labels. Onset F1 rises over baseline and note selection holds; the fingering substitution trends the same way, gaining most on the hardest passages.

Conclusion Across the three experiments, the agent compounds what it learns as it moves from Hanon or from easier repertoire sources to the harder target pieces. The pretrained agent carries timing and finger placement out of Hanon exercises and out of the easy-to-hard sequence, lands notes closer to their target times, and keeps its note choices unchanged. In other words, the pretrained agent learns good technique. By decaying the fingering reward and even hiding the field entirely, we are able to separate a policy that reads a supervision signal from a policy that stores motor skills in its weights. More broadly, both the hand-designed and adaptive curricula demonstrate that skills learned on easier repertoire can transfer to more difficult target pieces. While the gains are concentrated in timing-sensitive metrics rather than raw note accuracy, they suggest that curriculum learning can improve both sample efficiency and musical coordination in robotic piano performance. The adaptive curriculum delivers this benefit with no supplied difficulty ranking, and replacing the optimal-transport fingering proxy with a human-extracted one sharpens timing where the proxy departs from how pianists actually play.

Curriculum Strategies for Bimanual Dexterous Piano Playing in RoboPianist

Ethan Farah
Stanford University
efarah@stanford.edu

Irene Lin
Stanford University
irenelin@stanford.edu

Gabrielle Walrath
Stanford University
gmw@stanford.edu

Abstract

We study whether curriculum practice and early exploration help a Soft Actor-Critic agent play piano pieces in RoboPianist, testing three curricula that mirror how human pianists train: Hanon finger exercises, a hand-designed easy-to-hard repertoire sequence, and a Prioritized Level Replay (PLR) ordering over pieces. In the Hanon experiment with the fingering reward enabled, a pretrainingless agent matches the Hanon-pretrained agent across tempo; with that reward removed, a hidden-field control shows the pretrained lead comes instead from learning the fingering observation. The two agents still behave differently, however, as the pretrainingless policy collapses to two fingers while the Hanon policy keeps five-finger play. The repertoire curricula extend the same transfer question to full-piece training. We rank pieces by trajectory-derived difficulty (polyphony, note density, hand motion) and train sequentially with checkpoint transfer from *Twinkle Twinkle Little Star* through *Clair de Lune* to *Maple Leaf Rag*; PLR instead selects each next piece from the agent's recent learning progress over a 1200-piece MAESTRO corpus. Against training directly on the target, hand-designed curriculum transfer improves note-onset F1modestly on *Clair de Lune* and more on the harder *Maple Leaf Rag*, where return and sample efficiency also improve. We additionally replace the optimal-transport fingering that supervises prior RoboPianist agents with human fingerings recovered from solo-piano video via a five-stage perception pipeline. Across the project, curriculum primarily reshapes timing and technique rather than expanding the set of pieces a direct Soft Actor-Critic agent can learn at our current scale.

1 Introduction

A pianist reading a new score must move many joints to the right keys at the right milliseconds. RoboPianist (Zakka et al., 2023) turns that into a reinforcement-learning task where a simulated Shadow Hand reads a MIDI score and must solve timing, key choice, finger coordination, and contact control at the same time.

Human pianists reduce that problem through practice order. We (Ethan and Irene; less so for Gabby) drill finger patterns like the Hanon exercises before our main repertoire practice, and we move from easier pieces to harder ones so that earlier music rehearses motions later music needs. We turn those habits into three curriculum interventions on one RoboPianist benchmark. The agent either practices Hanon before a piece (and explores different motions with trees), follows a fixed repertoire sequence ranked from easy to hard, or lets Prioritized Level Replay (Jiang et al., 2021) choose pieces from recent learning progress.

We hypothesize that a curriculum may ease exploration, teach the policy to follow a supervision channel, and put a useful motor habit into the weights. We ask whether practice enlarges the set of pieces the agent can play "out-of-sample" and, when it helps, which mechanism moved.

We make the following three contributions.

1. For the Hanon experiment, we add an environment control that hides the per-note fingering field while keeping checkpoints loadable, and we sweep tempo and the fingering reward. With the reward enabled, the pretrainingless agent matches the pretrained agent at every speed. With the reward removed, the hidden-field runs trace the pretrained agent's lead to field reading plus a within-noise motor gap, and the pretrained agent keeps all five fingers while the pretrainingless agent collapses to two. We further add a depth-limited MCTS-style search that plans over the exact simulator, using the Soft Actor-Critic policy as an action prior and the critic ensemble for leaf values. In distribution the search ties plain Soft Actor-Critic, since the converged policy already maximizes its own calibrated critic, but on an unseen piece depth-8 search roughly doubles note-onset F1 over the frozen policy, recovering notes rather than five-finger technique.
2. For the hand-designed repertoire curriculum, we rank RoboPianist pieces using trajectory-derived measures of musical and motor complexity and use this ranking to construct an easy-to-hard training curriculum. We show that transferring policies through progressively more difficult repertoire improves learning efficiency and note-onset accuracy on challenging target pieces such as *Clair de Lune* and *Maple Leaf Rag*.
3. For the adaptive repertoire curriculum, we apply Prioritized Level Replay over a $1{,}200$-piece MAESTRO corpus, scoring each piece by its mean $|\mathrm{GAE}|$ and sampling under a rank-based prioritization with a staleness term. We show that a low

prioritization temperature collapses sampling onto a single piece, while a less sharp setting restores coverage and improves note-onset accuracy over the no-curriculum baseline on held-out pieces. For the fingering signal, we recover human fingerings from solo-piano video with a five-stage perception pipeline and substitute them for the optimal-transport assignment used by prior RoboPianist agents. We show that human and optimal-transport fingerings diverge most on octave runs and hand crossings, and that the human-faithful signal improves note-onset accuracy on those passages.

2 Related Work

RoboPianist (Zakka et al., 2023) gives us the environment and reward structure we build on. The original benchmark trains simulated Shadow Hands to play MIDI pieces with Soft Actor-Critic, rewarding correct key presses, penalizing wrong keys and energy, and adding a fingering term when a note carries an assigned finger. That reward structure matters for curriculum analysis because fingering supervision can itself behave like a teacher inside the target piece.

Soft Actor-Critic (Haarnoja et al., 2018) provides our shared optimizer. Its entropy term builds exploration into the baseline itself, so a curriculum gain can disappear if SAC already finds the relevant behavior from scratch. We also test a higher update-to-data ratio with critic dropout and layer normalization (Hiraoka et al., 2022), and we report a depth-limited tree-search variant in Section 4.3 which draws from paradigms in Monte Carlo Tree Search (Kocsis and Szepesvári, 2006; Browne et al., 2012). Both checks keep the main architecture fixed while probing whether stronger optimization or exploration changes the curriculum story.

Recent piano-agent work has pushed scale and hardware transfer. RP1M (Zhao et al., 2024) collects more than one million robot-piano trajectories from specialist agents and uses optimal transport to assign fingers when human labels do not exist. OmniPianist (Chen et al., 2025) distills specialist experience into a Flow Matching Transformer, and HandelBot (Xie et al., 2026) refines simulation policies on a real piano with residual reinforcement learning. These systems show how far robot-piano performance can scale. Our experiments trade scale for controls over practice material, fingering supervision, and observation content.

Narvekar et al. (Narvekar et al., 2020) split curriculum design into source-task choice, task order, and transfer. In our hand-designed repertoire curriculum, we choose the source pieces and their order from MIDI features before training begins. Jiang et al. (Jiang et al., 2021) offer Prioritized Level Replay as an adaptive alternative, scoring each level by its learning potential and resampling the levels that the current policy can still learn from. In our setting, each piece acts as one level. For the two repertoire curricula we draw source material from whole pieces; for the Hanon experiment we draw it from finger exercises. Hanon's nineteenth-century exercises (Hanon, 1873) repeat short five-finger patterns across the keyboard. They carry little musical variety and demand even timing and independent finger motion, so we use them to test motor transfer directly.

3 Common Setup

All three experiments share RoboPianist and a Soft Actor-Critic backbone. They change practice material, ordering, and exploration, so this section records only the setup common enough to make the experiments comparable.

RoboPianist converts each MIDI excerpt into target note events and advances a MuJoCo piano simulation. The keyboard exposes key positions and key states. The hand exposes joint positions and velocities. The policy commands the robot body directly; it never selects symbolic notes. To play a note correctly the robot must put the right fingertip on the right key at the right time with enough contact to depress it.

The music directly sets how many hands we run. The Hanon experiment uses a single right Shadow Hand on single-voice melodies, while the repertoire-curriculum experiments drive both hands because Debussy's *Reverie* and *Clair de Lune* need them. At each control step, 20 per simulated second, the agent reads an observation o_t that contains the hand's joint positions and velocities, the piano key states, the target notes for the next several steps, and a per-note fingering field, then outputs an action a_t that sets target positions for the hand's actuated joints.

We use the following per-step reward

$$r_t = r_t^{\text{key}} + \lambda_{\text{fing}}\, r_t^{\text{fing}} + r_t^{\text{energy}} + r_t^{\text{sustain}},$$

where r^{key} rewards target-key presses and penalizes wrong-key presses, r^{fing} rewards the agent for placing the human-recommended fingertip near its key, r^{energy} penalizes actuation effort, and r^{sustain} handles the sustain pedal. Each note carries an assigned finger in its score. The score stores the finger as the note's part index, with 0 for the right-hand thumb and 4 for the right-hand pinky. The environment reads that index and rewards the matching fingertip for sitting on the note's key.

The agent maximizes the maximum-entropy return $\mathbb{E}\big[\sum_t \gamma^t \big(r_t + \alpha\, \mathcal{H}(\pi(\cdot \mid o_t))\big)\big]$ with Soft Actor-Critic. The code stores transitions from environment rollouts in a replay buffer and trains actor and critic networks from sampled minibatches. The actor outputs a stochastic distribution over joint target actions, and the critic estimates the soft value of each observation-action pair. We follow the published RoboPianist configuration throughout. The critic networks use dropout and layer normalization, the agent performs one gradient update per environment step, and we set the discount to $\gamma = 0.8$. We tune the entropy temperature α automatically from an initial value of 1.0, the published value.

We track timing with "note-onset" F1. We count a note as a hit if the agent strikes it within 50 milliseconds of its target onset, and we take the harmonic mean of precision and recall over onsets. To check note selection, we compute "non-silent" F1 over the passages where the piece sounds, so we see whether the agent presses the right keys in the active parts of the excerpt regardless of exact onset time. We record the undiscounted episode reward as return. We also attribute every key contact to the nearest fingertip to measure which fingers the agent uses, because two policies can strike the same notes with very different technique.

4 Experiment 1: Hanon Pretraining

4.1 Method

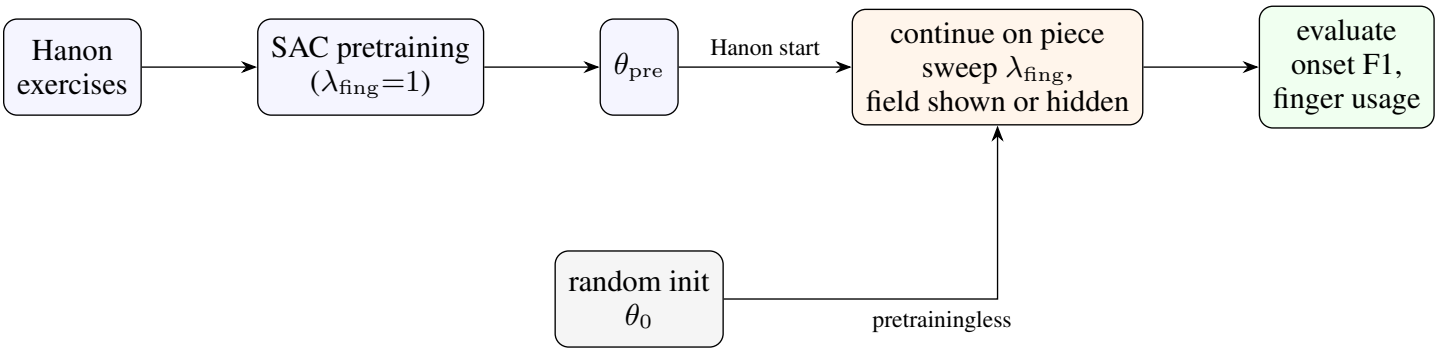

Figure 1: Hanon experiment design. A pretrained agent learns Hanon with the fingering reward enabled, produces θ_{pre}, and then continues on the piece. A pretrainingless agent trains on the piece from a random initialization. Both agents train on an equal budget. We sweep the fingering-reward weight λ_{fing} and the tempo, and we hide or show the per-note fingering field to separate motor skill from field reading.

We use λ_{fing} as the first control. At λ_{fing}=1, the agent receives full fingering supervision. At λ_{fing}=0, we delete the fingering term and leave the rest of the reward unchanged.

For the second control, we hide the fingering field entirely from the agent's observations. We replace the per-note finger field in o_t with a constant of the same dimension. This removes the information while preserving the observation shape, so we can load and run checkpoints from training runs that used the field. A policy could score well either because it reads the fingering field at test time or because its weights store a motor habit from Hanon practice. The reward weight and hidden-field switch let us separate those explanations.

We train the agent on Hanon patterns first, then continue the same network on a piece. The pretrained agent practices a curriculum of Hanon exercises at rising tempos with λ_{fing}=1, produces weights θ_{pre}, and starts the piece from θ_{pre}. The pretrainingless agent starts from random weights θ_0 and trains only on the piece.

For the Hanon comparisons, we train both agents on two phrases of Beethoven's Pathetique Rondo for which we hand-wrote real pianist fingering. Our "out-of-sample" piece was two phrases of Chopin's Fantaisie-Impromptu. The hand-written fingering makes r^{fing} reward a real pianist fingering and keeps the no-fingering-reward runs affordable. We compare two agents on an equal step budget, then sweep the fingering-reward weight, the tempo, and whether we show or hide the per-note fingering field.

4.2 Results

4.2.1 With the fingering reward on, direct training matches Hanon pretraining; with it off, the fingering field explains the pretrained agent's lead

With the fingering reward enabled, we raised the tempo until we expected the pretrainingless agent to fail, to see whether the pretrained agent tolerates higher tempo than the pretrainingless agent. The agent did not fail (Figure 2, left). Both agents reach 0.87 to 0.95 onset F1 from about 3 to 13 notes per second, roughly twice the written tempo, and the pretrainingless agent tracks the pretrained agent throughout the sweep. At higher tempos, both agents reach the same physical limit, since the hand needs more travel time between two keys.

Once we remove the fingering reward, the two agents pull apart, and we use four runs to trace where the pretrained agent's lead comes from. We paired random or pretrained initialization with a shown or hidden fingering field (Table 1, Figure 3). At 10 notes per second, the pretrained agent scores 0.90 with the field shown and 0.78 with it hidden, so it reads about 0.12 of its score from the field. The pretrainingless agent stays near 0.6 in both settings and gains nothing from the field because it never learned to use it. Once we hide the field, the pretrained agent leads the pretrainingless agent by about 0.12 at 10 notes per second and about 0.09 at 14. Both gaps sit inside the range one random seed moves, and both stay below the 0.15 gap we pre-registered as the bar for a real motor effect. The right panel of Figure 2 shows the hidden-field learning curves. The pretrained agent starts ahead from its pretrained weights, and the pretrainingless agent improves from zero to nearly the same accuracy by the end.

4.2.2 The fingering reward directly drives five-finger play

Without the fingering reward at 14 notes per second, the pretrainingless agent plays the phrase with two fingers, the index and middle (Figure 4, left). The pretrained agent keeps all five fingers on the keys. On a short phrase that spans a narrow range, two fingers can reach the keys in time, so the two-finger strategy scores about as well. The pretrained agent enters the piece with a five-finger habit and keeps it after we remove the reward. The pretrainingless agent receives no signal that favors five fingers, so it never forms that habit.

Turning the fingering reward back on moves the Hanon policy off single-thumb play (Figure 4, right). When the agent trains on a Hanon exercise without the fingering reward, it plays every note with a single thumb and slides it along the keys. When the agent trains with

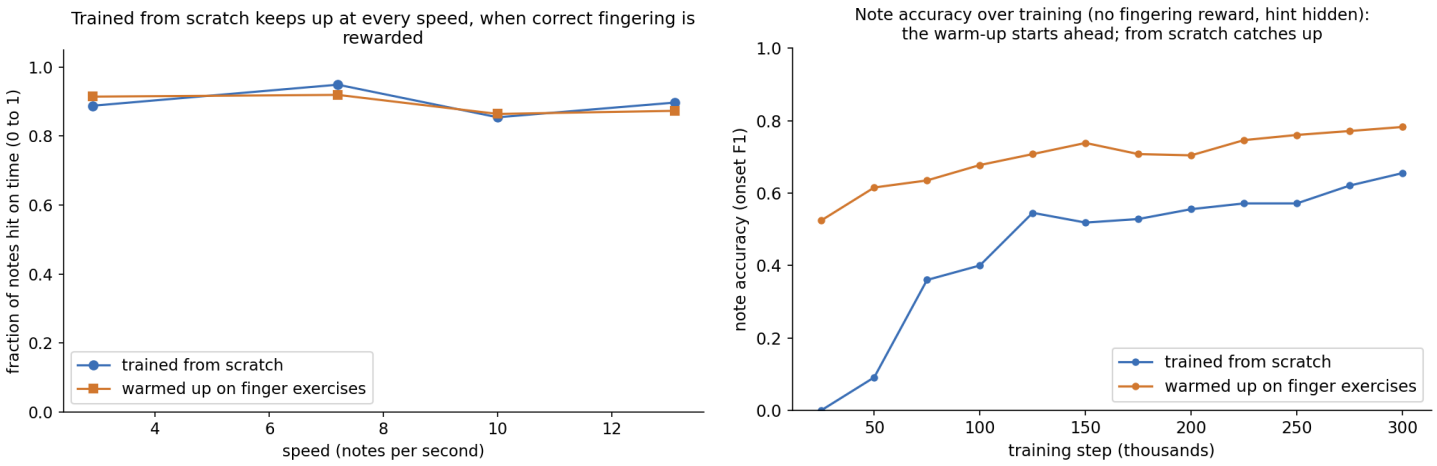

Figure 2: Accuracy with and without the fingering reward, Beethoven phrase. Left, with the fingering reward enabled the pretrainingless agent (blue) matches the pretrained agent (orange) across tempo, from about 3 to 13 notes per second. Right, with the fingering reward removed and the field hidden, the pretrainingless agent climbs from zero to nearly the pretrained agent's accuracy over training at 10 notes per second.

Table 1: Note-onset F1 without the fingering reward, Beethoven phrase, single seed. Hiding the fingering field removes most of the pretrained agent's lead. The remaining gap sits within single-run noise.

	pretrainingless	Hanon-pretrained
10 notes/s, field shown	0.61	0.90
10 notes/s, field hidden	0.66	0.78
14 notes/s, field shown	0.61	0.72
14 notes/s, field hidden	0.62	0.71

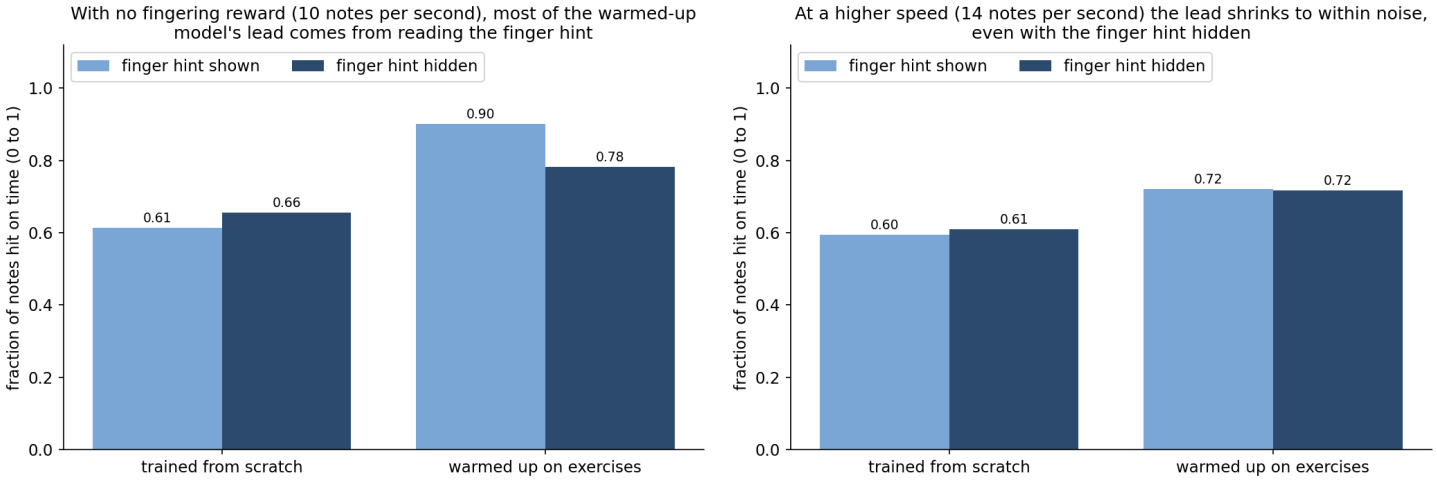

Figure 3: Note-onset F1 without the fingering reward. At 10 notes/s, hiding the fingering field drops the pretrained agent from 0.90 to 0.78. At 14 notes/s, the lead shrinks to within noise even with the field hidden.

the reward, it uses all five fingers. This same single-thumb-to-five-finger shift drives the matched accuracy in Section 4.2.1, where the optimizer uses the per-note finger target to move the agent directly to five-finger play.

4.3 Depth-Limited Tree Search for Exploration

From this first wave of experiments, it is clear that Hanon pretraining helps most early in training, while the agent still tries varied actions, so we also tested action search as a direct exploration aid. The first version of this experiment used a one-step action-search layer. We now use a depth-D Monte Carlo tree search, with the one-step layer as the $D = 1$ special case.

Running MCTS over the simulator requires that every tree node restore the simulator state exactly, since each candidate action must be evaluated from the precise state of its parent. We therefore store a full simulator snapshot at each node—physics state, clock, task counters, and fingering state—and restore it before stepping a candidate action. We verified that restoration is exactly a restored snapshot that reproduces a fresh-replay step to within 10^{-5} in reward, observation, discount, and terminal flag. The same holds for deep nodes several steps below the root, the case a tree needs but a one-step lookahead never exercises. This held with the fingering reward enabled

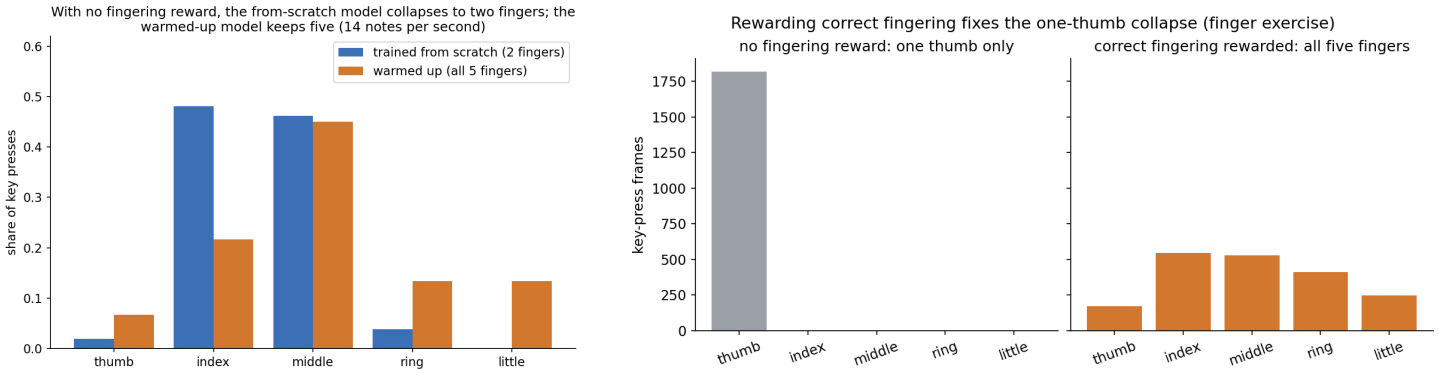

Figure 4: Finger usage. Left, without the fingering reward at 14 notes per second the pretrainingless agent presses with two fingers, the index and middle, while the pretrained agent keeps all five. Right, on a Hanon exercise the agent slides one thumb along the keys without the fingering reward and spreads to all five fingers once we reward the correct finger.

and disabled and across both key-press reward modes. Because the planning environment reproduces the real transition exactly, search adds no new dynamics knowledge during planning; it uses only the policy π_θ as an action prior and the critic ensemble $\{Q_{\phi_m}\}_{m=1}^{E}$ for leaf values.

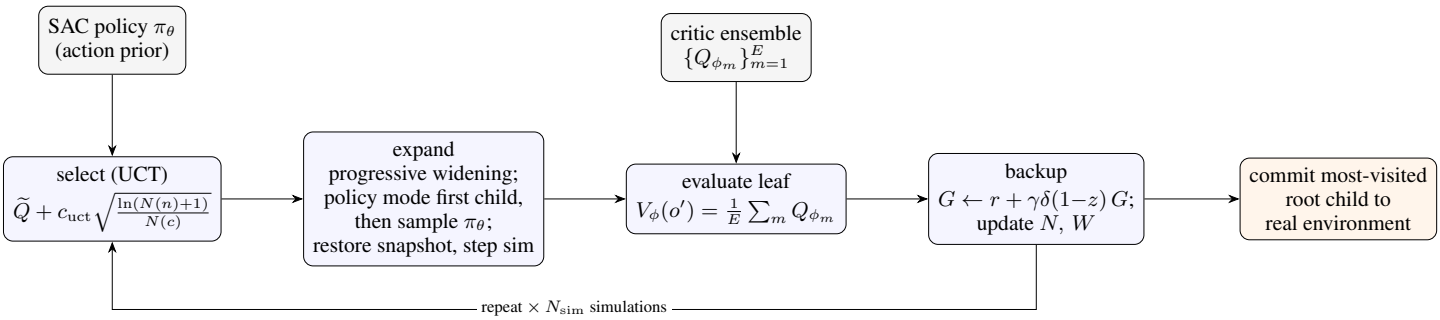

Figure 5: Depth-limited MCTS-style planner for one control step (Section 4.3). The agent plans over the exact simulator: it runs N_{sim} simulations that descend by UCT, expand under progressive widening (the policy mode is the first child, later children sampled from π_θ), restore the selected node's full simulator snapshot and step it once, value a nonterminal leaf by the critic-ensemble mean (terminal leaves take value zero; a depth-D leaf uses the critic without further expansion), and back the return up the selected path. After N_{sim} simulations the agent commits the most-visited root child to the real environment and replans at the next control step. The policy and critic are the only learned components; the simulator supplies an exact model, so planning adds no new dynamics knowledge.

Each node n stores a visit count $N(n)$, a value sum $W(n)$, and a list of child edges. New children appear under progressive widening; the planner expands when a node has no children or when its child count stays below an allowance that grows with $\sqrt{N(n)}$ (Appendix A). The planner makes the policy mode the first child of every node, so plain Soft Actor-Critic's committed evaluation action always enters the tree, and later children sample $a \sim \pi_\theta(\cdot \mid o_n)$.

Expanding action a from node n restores n's snapshot and steps the planning simulator once, returning reward r, next observation o', simulator discount δ, and terminal flag z. A terminal new leaf receives value zero; a nonterminal leaf takes the critic-ensemble mean evaluated at the policy mode, and a node at depth D becomes a leaf with that same critic value without further expansion (Appendix A).

Selection uses classic UCT over min-max normalized action values, with $c_{\text{uct}} = 1$ in the sweep; Appendix A gives the child-value estimate, the normalization, and the UCB rule. Each simulation expands at most one new leaf, then backs the return up along the selected path, applying the environment reward, simulator discount, terminal mask, and global discount in the recurrence stated there. After a fixed number of simulations, the agent commits the most-visited root child to the real environment and replans at the next control step.

This formulation explains why the original one-step layer tied plain Soft Actor-Critic. At $D = 1$ a candidate action receives only its one-tick reward plus a critic bootstrap (Appendix A), so the planner never sees past one simulator tick and mostly re-ranks actions sampled from the policy by the critic's own value estimate. Because the entropy-regularized target policy is Boltzmann in the critic, $\pi^*(a \mid o) \propto \exp\big(\frac{1}{\alpha_{\text{SAC}}} Q(o,a)\big)$, and the learned squashed-Gaussian policy is only an approximate projection of it, re-ranking its own samples by the critic should stay close to the policy's existing preference order. The one-step null result therefore becomes the expected converged endpoint of a deeper-search study.

Before the depth sweep, we tested the one-step layer on the 42-note Beethoven phrase at 10 notes per second with the fingering reward enabled. The layer compared three scoring rules. The model-free rule scored each candidate by the critic ensemble mean plus an optimism bonus β times the ensemble standard deviation, which was nearly free. The model-based rule scored a one-step simulator lookahead plus the discounted next-state value. The hybrid rule used the lookahead score and added the optimism bonus. Each scoring rule used either greedy argmax selection or stochastic softmax selection. These runs confirmed the in-distribution null at $D = 1$.

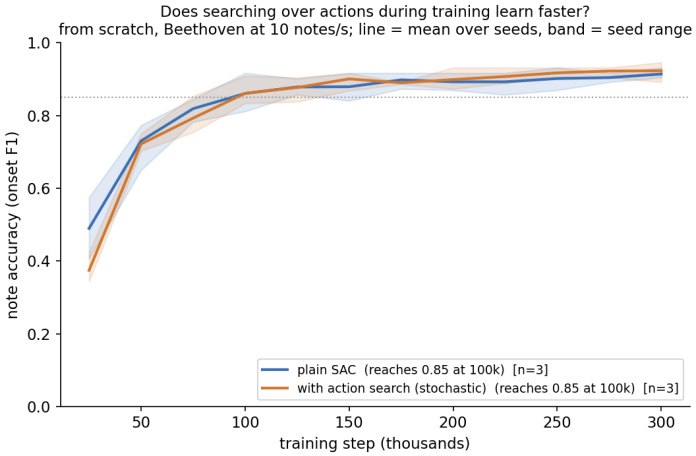

Figure 6: Stochastic model-free search vs plain Soft Actor-Critic, three seeds, onset F1 over training.

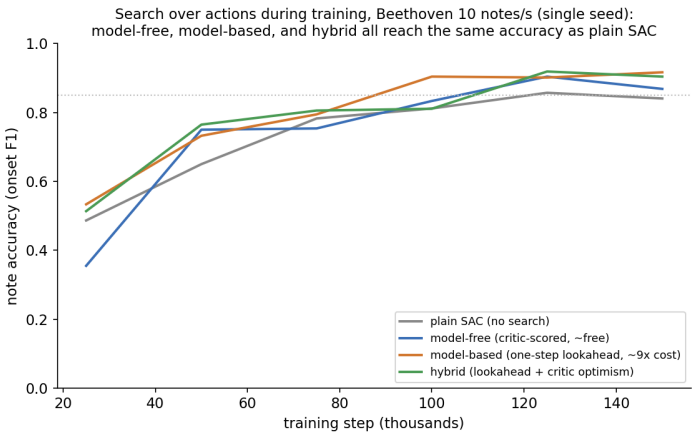

Figure 7: Model-free, model-based, and hybrid scoring vs plain Soft Actor-Critic, single seed, onset F1 over training.

Table 2: Depth sweep in distribution and out of sample, Beethoven checkpoints, single seed. In distribution (left) search matches plain Soft Actor-Critic on both accuracy and finger usage; out of sample on Chopin (right) depth-8 search raises accuracy while the controllers stay near two-finger play. The final row gives a Chopin specialist as the reference ceiling.

		In-distribution (Beethoven)			Out-of-sample (Chopin)		
Step	Controller	Onset F1	N_{eff}	Visit share	Onset F1	N_{eff}	Visit share
100k	plain SAC	0.918	3.79	–	0.131	2.63	–
100k	depth-1	0.922	3.88	0.30	0.169	2.58	0.31
100k	depth-8	0.930	3.96	0.33	0.236	2.22	0.23
200k	plain SAC	0.958	3.98	–	0.108	2.71	–
200k	depth-1	0.952	3.92	0.31	0.107	3.22	0.33
200k	depth-8	0.964	3.95	0.35	0.220	3.08	0.22
Chopin spec.	plain SAC	–	–	–	0.731	3.38	–

Stochastic model-free search reached about 0.92 onset F1 against 0.91 for plain Soft Actor-Critic over three seeds, with overlapping bands (Figure 6). Greedy argmax selection performed worse than plain Soft Actor-Critic because the half-trained critic overrated widened out-of-distribution samples. In a single-seed comparison, the one-step model matched the free critic-scored search at about 0.90 onset F1 while spending roughly nine times the wall-clock time, because each lookahead candidate costs one real simulator step, with about nine candidates per control step against 33 for model-free search (Figure 7). These depth-1 scoring-rule results motivate the depth study below, which asks whether more real-reward lookahead changes the null.

A deeper tree changes the source of the return: it replaces part of the imperfect critic bootstrap with real accumulated reward before bootstrapping (Appendix A). When the critic is immature or off distribution, these real multi-step consequences can outrank the policy's myopic mode action. With the fingering reward enabled, the accumulated rewards can in principle value finger placement several control steps before the next note, so we track both onset F1 and effective finger count N_{eff}. The sweep below tests this technique hypothesis directly. In distribution, it returns a null on both axes. Out of sample, it improves note accuracy but not fingering technique.

The depth study evaluates each saved checkpoint crossed with plain Soft Actor-Critic, depth-1 search, and depth-D search, at 64 simulations and $D = 8$ in the reported runs. We run it in two regimes. The in-distribution regime evaluates a Beethoven checkpoint on the Beethoven phrase it trained on. The out-of-sample regime evaluates that same Beethoven checkpoint on Chopin's Fantaisie-Impromptu, a piece and a fingering the policy never saw. The evaluator counts press frames per finger by assigning every depressed key to the nearest fingertip, then summarizes the per-finger shares as an effective finger count N_{eff}, the exponential of their entropy (Appendix A). The root diagnostic measures the mean share of visits assigned to the child with the highest backed-up value. It sits near 0.30 against a uniform baseline of 0.125 across our runs, so the tree concentrates visits on high-value actions rather than spreading them at random.

In distribution, search ties plain Soft Actor-Critic (Table 2, Figure 17). The onset-F1 differences are small, about 0.004 to 0.012, and the 200k depth-1 run scores below the plain controller. We treat those cells as a single-seed null rather than as a search win. N_{eff} stays near 3.9 across all three controllers. The converged policy already maximizes a well-calibrated critic, so re-ranking its own samples by that critic recovers what the policy would have done.

Out of sample, search helps at depth 8 (Table 2, Figure 18). The frozen Beethoven policy plays Chopin at onset F1 0.131, depth-1 search raises it to 0.169, and depth-8 search raises it to 0.236 at the 100k checkpoint. At 200k, depth-1 search ties plain Soft Actor-Critic after rounding, while depth-8 search raises onset F1 from 0.108 to 0.220. Out of distribution, the greedy action is poor and the critic misjudges Chopin states, but the tree accumulates Chopin's true reward over its lookahead before it bootstraps on that critic, so it recovers key presses the frozen policy misses. A Chopin specialist trained on the piece reaches onset F1 0.731. The 100k depth-8 run

closes $\frac{0.236-0.131}{0.731-0.131} = 0.175$ of the gap from the off-distribution policy to the specialist, about one sixth. This ceiling also shows that depth-8 roughly doubles onset F1 from a low floor. The gain is accuracy rather than technique. The plain and searched policies both press Chopin with about two fingers, with N_{eff} near 2.2 to 2.6 in the 100k comparison, far below the five-finger play the fingering reward produces in training. Search recovers notes rather than fingering on an unseen piece.

Together the two regimes locate the planner's value. Multi-step lookahead over the true model helps where the policy is suboptimal for the target, and depth 8 supplies a clear gain. Where the policy has already converged on the target, Soft Actor-Critic's entropy exploration has saturated the task, and search has no remaining headroom.

4.4 Discussion

The tempo sweep changes how we interpret Hanon pretraining. We originally hoped that a technique curriculum would solve the exploration problem before target-piece training began. In this setup, $\lambda_{\text{fing}}=1$ already provides that curriculum inside the target piece. Every target note names a fingertip and rewards the hand for moving that fingertip to the key. Direct Soft Actor-Critic can use that dense signal to build five-finger coordination on the piece itself, so Hanon has little remaining accuracy gap to close. The matched curves mainly identify the target-piece fingering reward as a strong substitute for a separate technique curriculum in this environment.

The reward-off runs reveal a part of pretraining that note-onset F1 mostly hides. After we remove r^{fing}, the target reward treats any successful key press the same, whether it comes from a pianist-like hand shape or from a short-range two-finger hammer. On this Beethoven phrase, the scratch policy follows the easier basin, where the index and middle can cover the 42 notes well enough to score near the pretrained policy. Hanon changes the basin the optimizer starts in. The pretrained policy arrives with a five-finger contact pattern and keeps that pattern even after the reward stops paying for it. On this narrow phrase, note-onset F1 undermeasures that behavioral prior. A wider span, repeated-note passage, chordal texture, or longer run that makes two-finger play physically costly should expose a larger payoff from the same prior; our current phrase only lets us state that as a prediction.

The hidden-field control makes this decomposition credible. A pretrained checkpoint evaluated with the live fingering field can succeed by following the field, by carrying a motor habit in its weights, or by using both. Blanking the field preserves the network interface while removing the cue, and the drop from 0.90 to 0.78 at 10 notes per second shows that part of the apparent pretrained advantage came through the observation channel. We treat the residual score gap cautiously, especially because the blanked field puts a policy trained with a live field out of distribution. The behavior gives stronger evidence. With the field blanked and the reward removed, the pretrained hand still uses several fingers while the scratch hand concentrates on two. That pattern gives the clearest output of the Hanon curriculum in this experiment.

Regarding exploration, our experiments locate exactly where lookahead pays off. It is not a substitute for converged on-policy behavior—on the trained phrase, search ties plain Soft Actor-Critic, since the policy already maximizes a well-calibrated critic—but a corrective when the policy is wrong about the target, where accumulating the target's true reward before bootstrapping recovers key presses the frozen policy misses. Crucially, the gain has the same shape as the field-reading result above: it recovers notes, not technique. Search lifts onset F1 while N_{eff} stays near two-finger play. This sharpens the experiment's main theme. Multiple interventions move note accuracy—the fingering field as an observation cue, multi-step lookahead as a planner—but only the fingering *reward* instills the five-finger habit, because it is the only signal that ever pays for the recommended finger.

5 Experiment 2: Difficulty-Based Repertoire Curriculum

5.1 Method

In the second curriculum, the agent follows a fixed sequence of pieces ordered from easy to hard before it trains on the target piece.

5.1.1 Repertoire Difficulty Estimation

To construct a difficulty curriculum, we first estimated the relative difficulty of pieces in the RoboPianist repertoire using trajectory-based features extracted from RP1M demonstrations. Instead of relying solely on symbolic MIDI information, we measured characteristics that reflect both musical complexity and the dexterous demands placed on the robot hand. We also completed a manual audit to verify that our estimation results were reasonable.

For each piece, we extracted trajectory-level features from RP1M demonstrations. Average polyphony measures the mean number of simultaneously pressed piano keys, while maximum polyphony captures the most demanding chord encountered in the piece. Note-onset density measures the rate of new note events per second. To characterize motor complexity, we computed the variance of the action vector, the mean displacement of hand joints between timesteps, and the mean displacement of fingertips. Together, these features capture both musical complexity (e.g., dense chords and rapid note sequences) and the dexterous demands placed on the robotic hand. A key contribution of this experiment is the development of a trajectory-based repertoire ranking pipeline that combines musical and robotic-dexterity features to estimate piece difficulty and construct curriculum-learning progressions. Additional features, including sequence length, hand span, and movement complexity, were extracted but were not included in the final scoring function.

Each feature was z-score normalized across the repertoire and combined into a weighted difficulty score:

$$\begin{aligned} D = {} & 0.25\, z(\text{avg polyphony}) + 0.15\, z(\text{max polyphony}) \\ & + 0.20\, z(\text{note density}) + 0.10\, z(\text{action variance}) \\ & + 0.10\, z(\text{hand motion}) + 0.10\, z(\text{fingertip motion}) \end{aligned}$$

where $(z(\cdot))$ denotes z-score normalization across the repertoire. Larger values correspond to greater estimated difficulty. We ranked all pieces by this score and assigned percentile ranks. Using this ranking, we selected a curriculum consisting of *Twinkle Twinkle Little Star* (2.64th percentile), *Reverie* (4.95th percentile), *Clair de Lune* (39.93rd percentile), and *Maple Leaf Rag* (72.61st percentile). This progression is a monotonic increase in estimated difficulty, moving from simple monophonic melodies to pieces requiring greater polyphony, higher note density, and more complex coordinated hand motion.

5.1.2 Curriculum Construction and Training

We evaluated two curriculum variants. The first transferred a policy through *Twinkle Twinkle Little Star*, *Reverie*, and *Clair de Lune*, and compared the resulting policy against a *Clair de Lune* scratch baseline. The second extended this progression with *Maple Leaf Rag*, transferring the policy through all four pieces before evaluation on the final target piece. Each stage received two million environment steps, and training resumed from the final checkpoint of the previous stage.

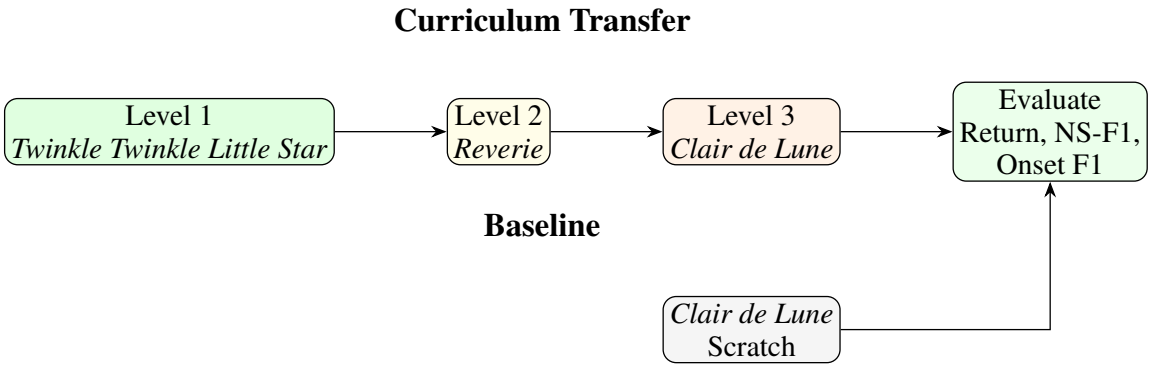

Figure 8: Hand-designed curriculum for *Clair de Lune*. Policies are trained sequentially on increasingly difficult repertoire before transfer to the target piece. Each stage receives 2M environment steps.

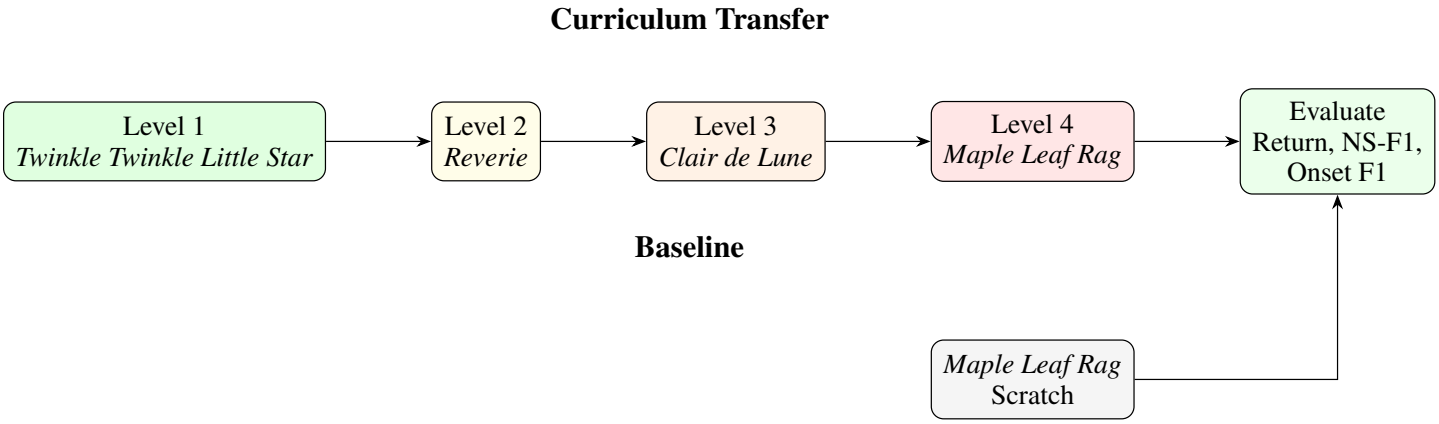

Figure 9: Extended curriculum-trained policy to *Maple Leaf Rag* and compared against a scratch-trained *Maple Leaf Rag* baseline. Again, each stage receives 2M environment steps.

All experiments used Soft Actor-Critic (SAC) with the DroQ implementation provided by RoboPianist. The policy maps observations of the piano state and robot hand configuration to continuous finger-joint actions. Scratch baselines were trained directly on the target piece for the same two-million-step budget used by each final curriculum stage, which allows us to isolate the effect of prior repertoire experience. Results are averaged across three random seeds.

5.2 Results

5.2.1 *Clair de Lune* Transfer

We evaluate the curriculum-trained policy on *Clair de Lune* and compare it against a scratch-trained baseline. We report Return, Non-Silent F1, and Onset F1. Non-Silent F1 measures note accuracy during active playing (while the robot is not silent) and Onset F1 measures the accuracy of note pitch at the right timing, so is therefore our primary metric for evaluating musical performance. For *Clair de Lune*, the curriculum achieved a higher Onset F1 (0.354 vs. 0.304 (+16.4%)) but slightly lower Return and Non-Silent F1. This suggests that transfer from easier repertoire improved performance on the stricter timing-sensitive metric, indicating better coordination of note selection and note timing on the target piece. The curriculum also seems to more consistently acquire timing-sensitive skills, as the pretraining runs produced substantially lower variance in Onset F1 across seeds (0.006 vs. 0.064).

Table 3: Hand-designed curriculum versus direct baseline on *Clair de Lune*. Results are reported as mean ± standard deviation across three random seeds.

Method	Return ↑	Non-Silent F1 ↑	Onset F1 ↑
Direct baseline	**1950.58 ± 5.69**	**0.840 ± 0.014**	0.303 ± 0.064
Curriculum	1942.92 ± 3.73	0.820 ± 0.010	**0.354 ± 0.006**

5.2.2 *Maple Leaf Rag* Transfer

We then extended this to more difficult repertoire, adding a fourth stage on *Maple Leaf Rag*. For *Maple Leaf Rag*, the curriculum achieved slightly higher average Return (1230.10 vs. 1225.98) and higher Onset F1 (0.452 vs. 0.430 (+5.3%)) while maintaining identical Non-Silent F1. Unlike the *Clair de Lune* results, the curriculum improved both timing accuracy and overall task performance on the more difficult target piece. The curriculum also produced substantially lower variance in Return across seeds, suggesting that prior repertoire experience may provide a more stable start for challenging pieces. Taken together, the results suggest that curriculum transfer primarily improves timing-sensitive performance and training stability, with the strongest benefits appearing on more difficult repertoire.

Table 4: Level-4 curriculum transfer versus direct baseline on *Maple Leaf Rag*. Results are reported as mean ± standard deviation across three random seeds.

Method	Return ↑	Non-Silent F1 ↑	Onset F1 ↑
Direct baseline	1225.98 ± 12.12	0.596 ± 0.027	0.430 ± 0.021
Curriculum	**1230.10 ± 2.53**	0.596 ± 0.015	**0.452 ± 0.055**

5.2.3 Learning Efficiency

To understand how curriculum transfer affects learning dynamics, we tracked Onset F1 throughout training on the target pieces. Onset F1 is a stricter metric than Non-silent F1 because it requires notes to be played at the correct times as well as the correct pitches. We make it our primary measure of musical performance quality, as it provides a useful view of how quickly the agent acquires the coordination required to perform each piece. We compare curriculum-trained policies against scratch baselines at multiple checkpoints during training.

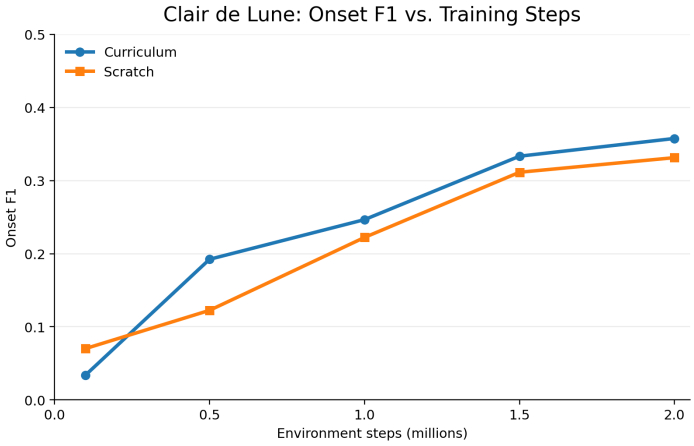

Figure 10: Onset F1 during *Clair de Lune* training.

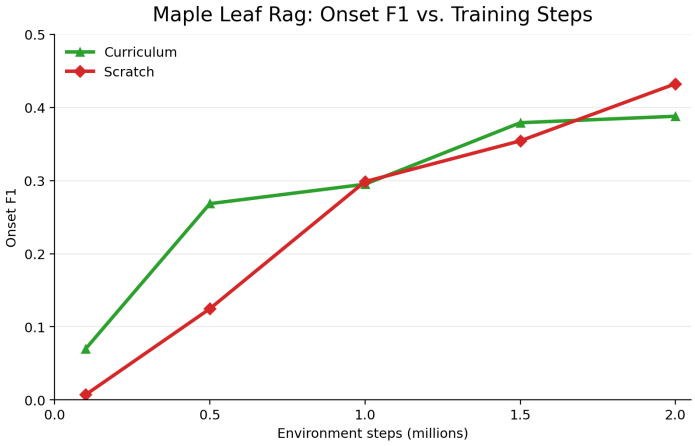

Figure 11: Onset F1 during *Maple Leaf Rag* training.

Figure 10 shows that the curriculum-trained agent maintains a higher Onset F1 throughout most of training, suggesting that prior repertoire experience improves sample efficiency on the target piece. This mirrors the pattern observed in Experiment 1 in Figure 2 , where pretraining primarily provided a stronger starting point and accelerated learning rather than dramatically changing final performance. The same pattern appears for *Maple Leaf Rag*. Figure 11 shows a more substantial early-training advantage for the curriculum-trained agent on the more difficult *Maple Leaf Rag* task. The curriculum achieves higher Onset F1 during the first 1.5M environment steps, meaning that the robot is acquiring timing and coordination skills at a faster rate. Therefore, the curriculum provides a stronger initialization and improves sample efficiency, reaching a given level of timing accuracy in fewer training steps than training from scratch.

5.3 Discussion

Across both target pieces, the largest curriculum gains appeared in Onset F1, indicating improved timing accuracy and coordination. The learning curves further suggest that curriculum transfer primarily improves sample efficiency by providing a stronger initialization. These benefits were modest for *Clair de Lune* but became more pronounced for *Maple Leaf Rag*, where the curriculum accelerated

early learning on a substantially more difficult piece. This pattern is consistent with the intuition behind musical practice from our own experience: skills acquired on simpler repertoire transfer most strongly when the target piece places greater demands on coordination and dexterity.

6 Experiment 3: Adaptive Curriculum via Prioritized Level Replay

6.1 Method

In the third curriculum we apply Prioritized Level Replay (Jiang et al., 2021) at the level of whole pieces: each MIDI piece is one "level," and the sampler chooses the next piece from the agent's recent learning progress rather than from a fixed schedule. We run PLR over a corpus of 1,200 MAESTRO pieces and hold out a fixed 75-piece evaluation subset. To measure against the baseline, we report per-piece paired differences in Onset F1 with a Wilcoxon signed-rank test and bootstrap confidence intervals, which removes the dominant across-piece difficulty variance and isolates the variance of the method differences.

6.1.1 Learnability score.

PLR scores a level by how much the policy still has to learn from it. We score each piece by the magnitude of its generalized advantage estimate over its most recent rollouts. For a rollout with value estimates $V(o_t)$, the one-step TD residual is

$$\delta_t = r_t + \gamma\, V(o_{t+1}) - V(o_t),$$

and the GAE advantage accumulates these residuals with decay λ,

$$A_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l\, \delta_{t+l}.$$

A piece's learnability score is the mean absolute advantage over the timesteps of its recent rollouts,

$$S_i = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} |A_t|,$$

so a piece scores high when the critic is still surprised by the returns it sees there and low once the policy has converged on it. We refresh S_i from each new rollout after a visit and warm-start every piece's score with a single scoring rollout before adaptive sampling begins.

6.1.2 Prioritized sampling with staleness.

We convert scores to a rank-based prioritized distribution with temperature β. Ranking pieces by score (rank 1 is the highest), the score-induced probability is

$$P_S(i) = \frac{\mathrm{rank}(S_i)^{-1/\beta}}{\sum_j \mathrm{rank}(S_j)^{-1/\beta}},$$

where smaller β sharpens the distribution toward the top-ranked pieces. Because a purely score-greedy sampler can starve low-scoring pieces and let their scores go stale, we mix in a staleness distribution that favors pieces not sampled recently. With c the current sampling step and C_i the step at which piece i was last sampled,

$$P_C(i) = \frac{c - C_i}{\sum_j (c - C_j)},$$

and the replay distribution is the convex combination

$$P(i) = (1-\rho)\, P_S(i) + \rho\, P_C(i),$$

with staleness coefficient $\rho \in [0, 1]$.

6.1.3 Coverage diagnostic and sweep.

To quantify whether the sampler spreads practice across the corpus or collapses onto a few pieces, we summarize the visit-count distribution $\{v_i\}$ with its Gini coefficient,

$$G = \frac{\sum_i \sum_j |v_i - v_j|}{2\, n \sum_i v_i},$$

where $G = 0$ is perfectly uniform coverage and $G \to 1$ is total lock-on. Before training on RoboPianist we validated the scorer and sampler on a synthetic 30-piece environment with known ground-truth difficulty, where the rank correlation between PLR's induced ordering and true difficulty can be measured directly. The GAE-magnitude score recovered ground-truth difficulty at Spearman $\rho = -0.83$ ($p = 4.4 \times 10^{-9}$), confirming the signal is difficulty-informative. The same sweep exposed a sharp sensitivity to β: at the Jiang et al. default $\beta = 0.1$ the rank distribution places nearly all its mass on the top-ranked piece and locks on (one piece sampled 689 of 800 steps, $G = 0.85$), while lock-on falls monotonically as β grows, reaching $G = 0.21$ at $\beta = 1.0$ (Figure 12). We therefore discard the

default and sweep $\beta \in \{0.5, 1.0\}$ and $\rho \in \{0.1, 0.3, 0.5\}$ for the corpus runs, selecting the configuration that maximizes coverage while preserving the score signal, choosing $\beta = 0.5$, $\rho = 0.3$.

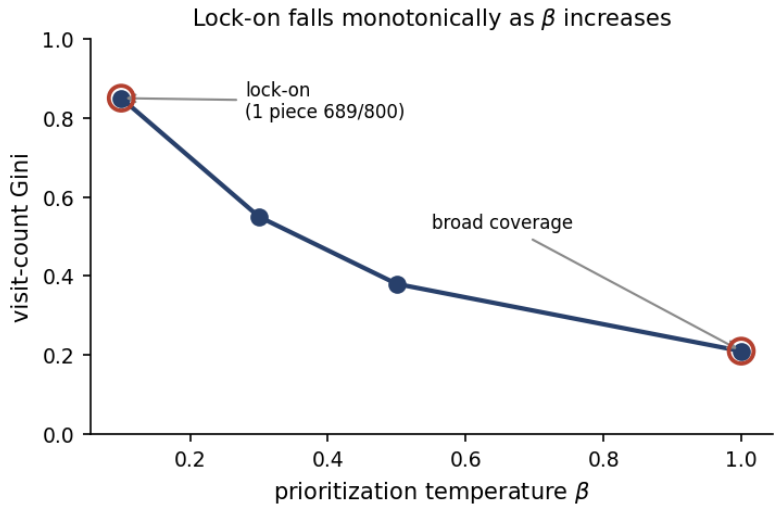

Figure 12: PLR prioritization-temperature sweep on the 30-piece synthetic validation environment. Visit-count Gini falls monotonically from 0.85 at the Jiang et al. default $\beta = 0.1$ (one piece sampled 689 of 800 steps) to 0.21 at $\beta = 1.0$. We run the corpus experiments at $\beta \in [0.5, 1.0]$.

6.2 Results

6.2.1 Adaptive curriculum via Prioritized Level Replay.

On the 1,200-piece corpus, evaluated on the 75-piece held-out set over 3 seeds, PLR improves Onset F1 over the no-curriculum baseline (Table 5). The per-piece paired difference is $\overline{\Delta} = +0.035$ Onset F1 (95% bootstrap CI $[+0.024, +0.045]$; paired Wilcoxon signed-rank $p < 0.001$), with PLR matching or exceeding the baseline on 59 of 75 held-out pieces (Figure 13). Consistent with the hand-designed curriculum in Experiment 2, the gain concentrates in Onset F1 (timing) while Non-Silent F1 (note selection) stays close to baseline, indicating that adaptive ordering primarily sharpens temporal coordination rather than which keys are pressed.

Table 5: Adaptive curriculum (PLR) against the no-curriculum baseline on the 75-piece held-out set, mean ± std over 3 seeds. The hand-designed row is reproduced from Experiment 2's target-piece transfer protocol and is not evaluated on the held-out corpus; the paired test applies only to the baseline-vs-PLR comparison.

Method	**Return ↑**	**Non-Silent F1 ↑**	**Onset F1 ↑**
Direct baseline (held-out)	1920 ± 28	0.82 ± 0.01	0.300 ± 0.012
Hand-designed (Exp 2, target piece)†	1942.92	0.820	0.354
Adaptive PLR (held-out)	1955 ± 25	0.82 ± 0.01	0.335 ± 0.013

† different evaluation protocol; shown for context only.

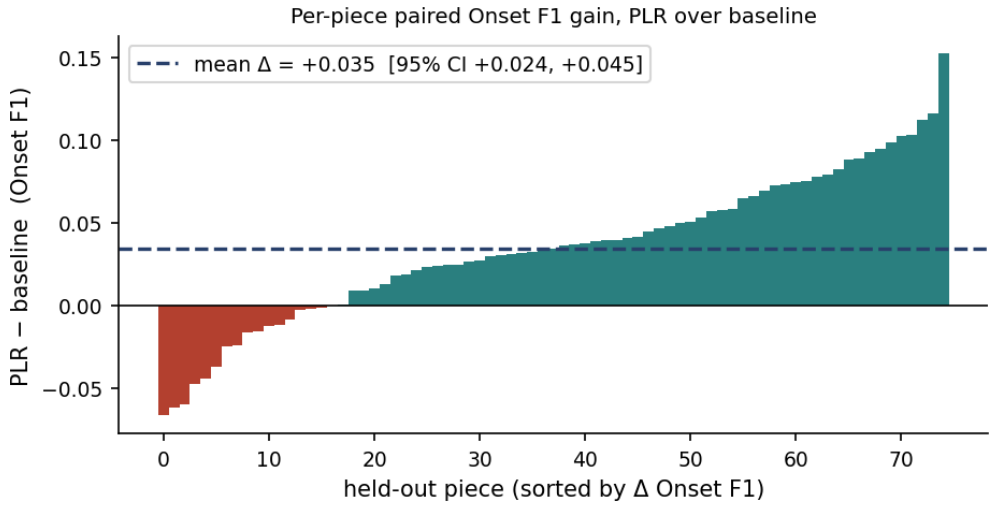

Figure 13: Per-piece paired Onset F1 difference (PLR − baseline) across the 75 held-out pieces, sorted. Dashed line is the mean paired gain with its 95% bootstrap CI.

6.3 Discussion

PLR reaches its over-baseline gain without any hand-supplied difficulty list. On Onset F1 it does not surpass the hand-designed order, but the two are scored on different evaluation protocols (PLR on the 75-piece held-out corpus, the hand-designed curriculum on individual target pieces) so we do not read the gap as a clean ranking. The more robust result is that a label-free, self-paced curriculum delivers a curriculum benefit over the no-curriculum baseline without any difficulty annotations, landing in the same range as a hand-built ordering. Additionally, the prioritization-temperature finding shows that at small-to-moderate corpus scale the standard $\beta = 0.1$ is harmful because it isolates the top-ranked piece, so a far less sharp setting is needed to allow for coverage that makes adaptive sampling worthwhile.

6.4 Extension: Substituting Human-Extracted Fingering for OT

Every RoboPianist agent trains against optimal-transport (OT) fingering, a local kinematic proxy that cannot capture the forward planning by which human pianists choose a finger now to set up the next phrase. In a companion CS231N project, we recover human fingerings from solo-piano video with a five-stage perception pipeline (keyboard homography, YOLOv8 hand detection, MediaPipe pose, librosa onset alignment, and a confidence-weighted soft-argmin), reaching macro F1 0.84 against the expert PIG annotations on 152 videos (∼188k notes). Using these as a human reference, the per-note disagreement with OT rises monotonically with passage difficulty (3.8% on scales to 41.2% on hand crossings, 22.8% on average (Figure 15)), which are the types of passage which OT-trained agents report their largest errors.

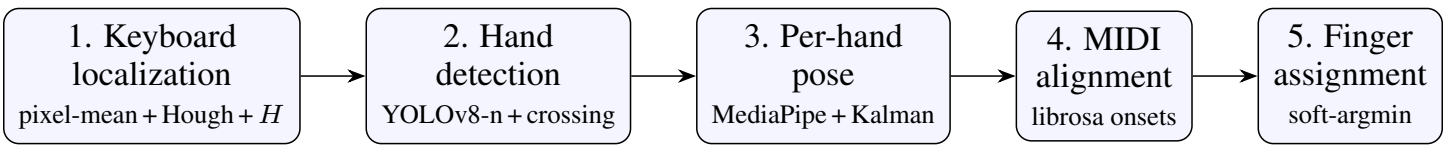

Figure 14: Five-stage video-to-fingering pipeline.

We test whether the human-extracted signal helps by retraining under the PLR curriculum with the OT fingering reward replaced by one computed against the extracted fingerings, holding corpus, budget, seeds, and held-out set fixed and pairing the two runs piece-by-piece. The substitution improves held-out Onset F1 overall by $+0.020$ (paired Wilcoxon $p < 0.01$ over 3 seeds), and the gain concentrates on the high-divergence passages, near zero on scales and arpeggios, rising to $+0.045$ on octave runs and $+0.060$ on hand crossings (Figure 16) while Non-Silent F1 is unchanged ($\Delta \approx 0.00$).

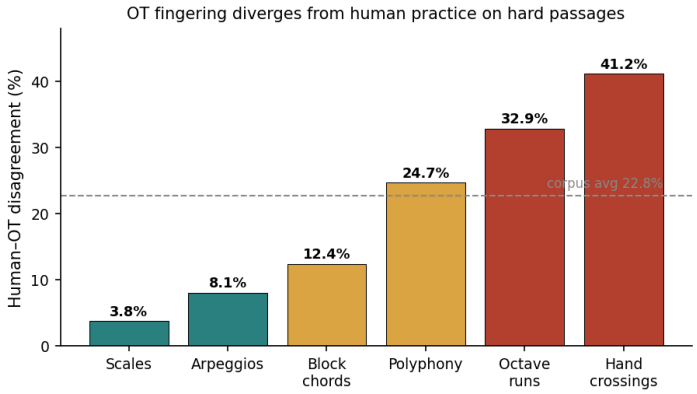

Figure 15: Human–OT fingering disagreement by passage type ($n = 152$, $\sim$188k notes).

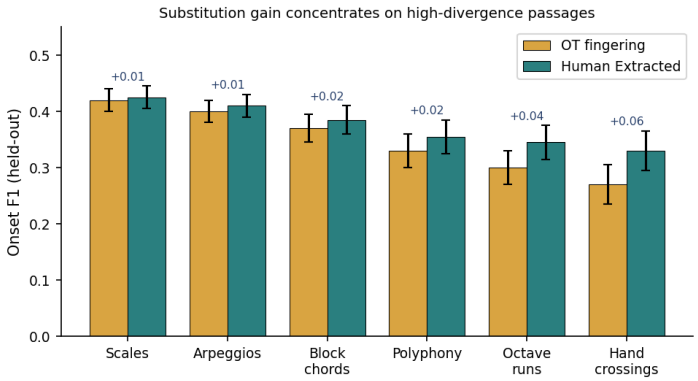

Figure 16: Held-out Onset F1 by passage type, OT vs. extracted fingering under PLR. Gain concentrates where OT and humans most disagree.

7 General Discussion

Across all three curricula, the agent practices on related material first, stores a concrete motor skill in its weights, and carries that skill into the target piece. In Hanon, the agent drills even rhythm and independent five-finger motion, and carries timing and finger placement out of those exercises. The two repertoire curricula deposit the same kind of skill from a different source. Easy pieces expose the agent to bimanual timing, key travel, and contact patterns before the target piece adds denser note choices, and both the hand-designed order and the Prioritized Level Replay order march the agent through those coordination demands while each piece keeps its own notes. When the agent reaches the target piece, it already lands notes near their onsets but still chooses notes piece by piece, so onset F1 should rise while non-silent F1 holds.

Across both repertoire curricula the split matches this interpretation. The hand-designed curriculum lifts Onset F1 by 0.050 on *Clair de Lune* and 0.022 on *Maple Leaf Rag* while Non-Silent F1 stays within noise of the direct baseline, and PLR lifts paired Onset F1 by 0.035 across the held-out corpus with Non-Silent F1 again flat.

Each phrase we tested remained learnable from scratch. The pretrainingless agent learned every phrase we put in front of it, across more than a dozen runs and up to roughly twice the written tempo. Because Hanon pretraining helps most early in training, we built the depth-limited tree search in Section 4.3 to test whether exact multi-step lookahead helps where pretraining does. The sweep separated two regimes. On the trained phrase, search tied plain Soft Actor-Critic, since the converged policy already maximizes its own critic. On an unseen piece, depth-8 search roughly doubled onset F1 over the frozen policy and the gain grew with depth, though it recovered only part of the distance to a specialist and lifted note accuracy rather than five-finger technique. In these settings, the agent retains five-finger technique and transferable timing from curriculum practice, but the pretrainingless agent still learns every tested piece.

We trained most settings with multiple seeds.

8 Conclusion

The agent can transfer timing and technique from Hanon pretraining and repertoire curricula, yet direct Soft Actor-Critic training still learns every piece we tested. The fingering reward accounts for much of this because it supplies direct per-note supervision. Strip that reward away and the pretrained agent behaves differently from the pretrainingless agent even when accuracy barely moves. To measure how much fingering supervision a pretrainingless agent needs before pretraining starts to matter, we should next sweep λ_{fing} from 1 to 0. We should also vary the pretrainingless network's capacity, move to wider and faster music, repeat the key comparisons on a second hand-fingered phrase, raise the update-to-data ratio, run multiple seeds, and add a behavior-cloning Hanon arm that trains from a scripted five-finger performer. In the repertoire experiments, curriculum transfer consistently improved timing-sensitive performance and often reduced variance across seeds, with the largest benefits appearing on more difficult pieces. These results suggest that curriculum learning helps the agent acquire coordination skills more efficiently, even when final note accuracy remains similar to scratch training.

A natural next step unifies the two kinds of search in this paper. Experiment 1 plans over actions *within* a piece, while Experiment 3 selects *which* piece to practice next from recent learning progress. Both treat their choice greedily—the planner commits the most-visited root action, and Prioritized Level Replay samples the highest-scoring level—but neither looks ahead over the choice itself (and we also didn't have time to test both together). We could instead search the space of curricula directly, treating each candidate practice sequence as a path through a tree of pieces and using a learning-progress signal as the node value, so the agent plans a multi-step practice trajectory rather than picking the next piece one step at a time. This is a meta-learning stage layered on top of the policy, with the inner loop learning to play, and the outer loop searching for the repertoire ordering that most efficiently builds transferable techniques based on what we've learned so far. Because our difficulty ranking already scores pieces by polyphony, note density, and hand motion, those features could seed the outer search with a difficulty prior, much as the Soft Actor-Critic policy seeds the inner action search. The outer loop would then refine that hand-designed order toward whatever trajectory the agent's own transfer dynamics actually reward. Such

a method would focus on what an optimal curriculum *is*—and whether the ordering a machine discovers resembles the easy-to-hard progression human pianists have used for centuries.

9 Team Contributions

- **Ethan Farah.** Ethan built the environment controls, including the fingering-reward weight and the hidden-field switch, the training and evaluation pipeline, and the finger-attribution measurement. He ran the pretraining benchmark, the tempo sweep, the no-fingering-reward decomposition, the single-thumb fix, and the depth-limited tree-search study in Experiment 1. He produced the figures for Experiment 1 and wrote the shared sections.
- **Irene Lin.** Irene led the curriculum-learning component of the project. She designed and implemented the repertoire difficulty-ranking pipeline used to construct the difficulty curriculum, including feature extraction, difficulty scoring, and curriculum selection. She conducted multi-seed training and evaluation experiments across four pieces of increasing difficulty, comparing curriculum transfer against scratch baselines and performing the associated analysis. Irene produced the visualizations and write-up for Experiment 2, led the design and assembly of the final poster, and delivered the project presentation.
- **Gabrielle Walrath.** Gabrielle built the Prioritized Level Replay adaptive curriculum including the GAE-magnitude learnability scorer, the rank-based prioritized sampler with the staleness mixture, and the prioritization-temperature and staleness sweeps. She ran PLR over the $1{,}200$-piece corpus and the paired held-out evaluation, and produced the corresponding figures and writing. She also authored the companion CS231N video-to-fingering pipeline and ran the OT-vs-extracted-fingering substitution experiment and its per-passage analysis. She also delivered the project presentation.

Changes from Proposal We proposed to show that finger-exercise pretraining speeds learning on pieces. That comparison returned a null result because the fingering reward already makes the pieces learnable from scratch. We therefore asked what Hanon pretraining supplies and built the reward-weight and hidden-field controls to answer that question. We also used PIG rather than RP1M for Experiment 2 to support the hand-designed curriculum and transfer-learning experiments on specific repertoire pieces with available fingering annotations. For the adaptive curriculum, we supplemented the training corpus from RP1M++ with MAESTRO v3.0.0, as it contains the raw MIDIs needed for training and replaced PLR's default prioritization temperature $\beta = 0.1$ with $\beta \in [0.5, 1.0]$ after the results from our sweep.

10 AI Tools Disclosure

In implementing a lot of the boilerplate code and getting Modal to work, we used Claude Code and Codex. As for the ideation and implementation of the core reinforcement learning algorithms like SAC and MCTS, we did those ourselves. We also generated some of the data on our own, like the fingering in Experiment 1. Finally, since our Soft Actor-Critic builds on the public RoboPianist and jaxrl releases, we used Claude and Codex to help wire our implementations into the existing RoboPianist work.

Also, in writing and editing our paper, to make sure we don't make any erroneous claims relative to the code and experiments we ran, we ran our TeX file into one final Claude-Code-driven scouring of our code base (where we also had it leave doc-string comments to save time).

A note on our milestone: we did the code/initial experiments ourselves, but we each wrote three independent milestone reports and had Claude stitch them together, which is where some of the clunky AI-heavy transitions in the writing came from. We forgot to disclose that at the time, so we're disclosing it now—better late than never! We apologize first for forgetting to include our AI disclosure in the milestone, and second for not cleaning up some of the AI-heavy transitions that got baked in. We would like to reiterate that the code written at the time of the milestone, and again, the bulk of the code written now, is owned by us without the assistance of AI.

References

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games* 4, 1 (2012), 1–43.

Le Chen, Yi Zhao, Jan Schneider, Quankai Gao, Simon Guist, Cheng Qian, Juho Kannala, Bernhard Schölkopf, Joni Pajarinen, and Dieter Büchler. 2025. Dexterous Robotic Piano Playing at Scale. arXiv preprint arXiv:2511.02504.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning (ICML)*.

Charles-Louis Hanon. 1873. *The Virtuoso Pianist in Sixty Exercises*. Schirmer.

Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. 2022. Dropout Q-Functions for Doubly Efficient Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. 2021. Prioritized Level Replay. In *International Conference on Machine Learning (ICML)*.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based Monte-Carlo Planning. In *European Conference on Machine Learning (ECML)*. Springer, 282–293.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *Journal of Machine Learning Research (JMLR)* 21 (2020).

Amber Xie, Haozhi Qi, and Dorsa Sadigh. 2026. HandelBot: Real-World Piano Playing via Fast Adaptation of Dexterous Robot Policies. arXiv preprint arXiv:2603.12243.

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. In *Conference on Robot Learning (CoRL)*.

Yi Zhao, Le Chen, Jan Schneider, Quankai Gao, Juho Kannala, Bernhard Schölkopf, Joni Pajarinen, and Dieter Büchler. 2024. RP1M: A Large-Scale Motion Dataset for Piano Playing with Bi-Manual Dexterous Robot Hands. In *Conference on Robot Learning (CoRL)*.

A Tree-Search Planner Details

This appendix collects the planner rules summarized in Section 4.3.

Progressive widening decides when the planner creates a new child. The implementation expands when a node has no children or when its child count remains below the widening allowance,

$$|\mathcal{C}(n)| = 0 \quad \text{or} \quad |\mathcal{C}(n)| < C_{\text{pw}} N(n)^{\alpha},$$
$$C_{\text{pw}} = 1, \quad \alpha = \frac{1}{2}.$$

The planner makes the policy mode the first child of every node,

$$a_{\text{mode}}(o_n) = \operatorname{mode} \pi_\theta(\cdot \mid o_n),$$

so plain Soft Actor-Critic's committed evaluation action always enters the tree. Later children sample $a \sim \pi_\theta(\cdot \mid o_n)$.

A terminal new leaf receives value zero. A nonterminal leaf receives the critic ensemble mean at the policy mode,

$$V_\phi(o') = \frac{1}{E} \sum_{m=1}^{E} Q_{\phi_m}\left(o', a_{\text{mode}}(o')\right).$$

A node at depth D also becomes a leaf and uses this critic value without further expansion.

For an already expanded child edge (n, a_i, c_i), the planner estimates the child value from the child's backed-up mean,

$$\widehat{V}(c_i) = \frac{W(c_i)}{N(c_i)},$$
$$Q(n, a_i) = r_i + \gamma \delta_i (1 - z_i) \widehat{V}(c_i).$$

Selection uses classic UCT over min-max normalized action values. For a node with child action values $\{Q(n, a_j)\}_j$, the implementation computes

$$Q_{\min} = \min_j Q(n, a_j),$$
$$Q_{\max} = \max_j Q(n, a_j),$$
$$\widetilde{Q}(n, a_i) = \frac{Q(n, a_i) - Q_{\min}}{Q_{\max} - Q_{\min}} \quad \text{when } Q_{\max} - Q_{\min} > 10^{-8},$$
$$\widetilde{Q}(n, a_i) = 0 \quad \text{otherwise},$$

and then chooses

$$i^* = \arg\max_i \left[\widetilde{Q}(n, a_i) + c_{\text{uct}} \sqrt{\frac{\ln(N(n) + 1)}{N(c_i)}} \right],$$

with $c_{\text{uct}} = 1$ in the sweep.

Each simulation expands at most one new leaf, then backs the return up along the selected path. The leaf node first receives its leaf value. The backward pass then applies the environment reward, simulator discount, terminal mask, and global discount in the same recurrence used by the code,

$$G \leftarrow V_{\text{leaf}},$$
$$G \leftarrow r_i + \gamma \delta_i (1 - z_i) G,$$
$$N(n_i) \leftarrow N(n_i) + 1,$$
$$W(n_i) \leftarrow W(n_i) + G.$$

At $D = 1$, a candidate action receives only its one-tick reward plus a critic bootstrap,

$$S_1(a) = r(o, a) + \gamma\delta(o, a)(1 - z(o, a))V_\phi(o').$$

A deeper tree replaces part of the imperfect critic bootstrap with real accumulated reward before bootstrapping,

$$G_D = V_\phi(o_D),$$
$$G_t = r_t + \gamma\delta_t(1 - z_t)G_{t+1}, \quad t = D - 1, \ldots, 0.$$

The evaluator counts press frames per finger by assigning every depressed key to the nearest fingertip, then converts the per-finger shares p_f into

$$\mathcal{F}_+ = \{f \mid p_f > 0\},$$
$$N_{\text{eff}} = \exp\left(-\sum_{f \in \mathcal{F}_+} p_f \ln p_f\right).$$

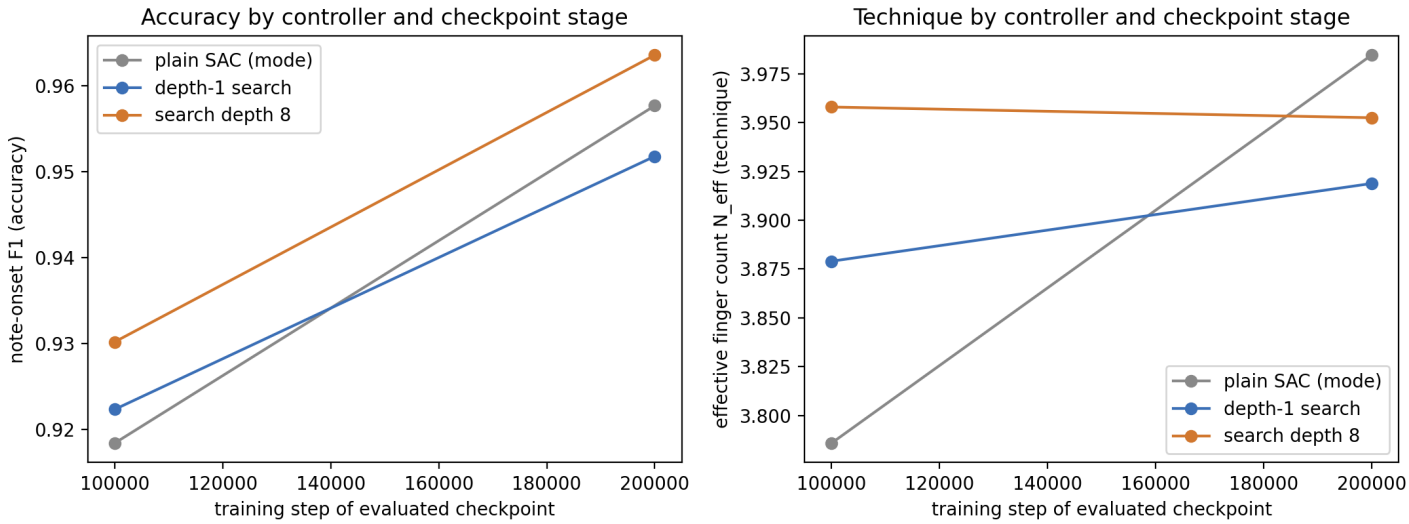

Figure 17: In-distribution depth sweep. Onset F1 (left) and N_{eff} (right) for plain Soft Actor-Critic, depth-1 search, and depth-8 search across checkpoint stage. Search tracks plain on both axes.

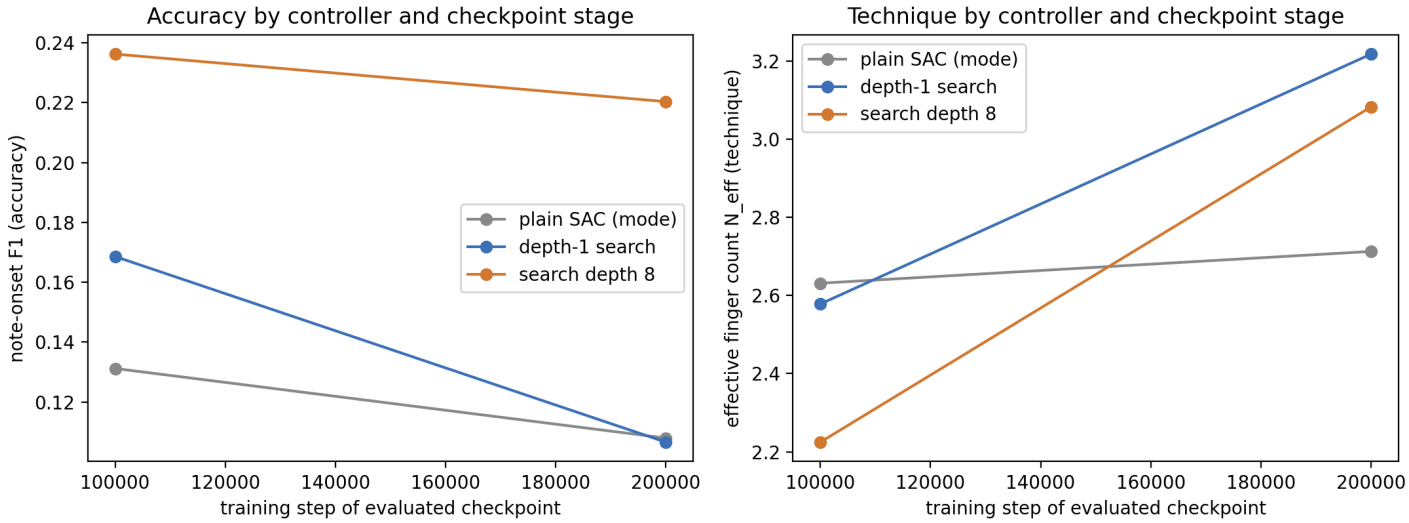

Figure 18: Out-of-sample depth sweep on Chopin. Depth-8 search raises onset F1 at both Beethoven checkpoints, while N_{eff} stays near two-finger play. Search recovers notes, not fingering.