From MIDI to Motion: Learning to Play the Piano at Scale with Bi-Manual Dexterous Robot Hands

Le Chen*, Yi Zhao*, Jan Schneider, Quankai Gao, Juho Kannala, Bernhard Schölkopf,
Joni Pajarinen, and Dieter Büchler

***Abstract*—It has been a long-standing research goal to endow robot hands with human-level dexterity. Bimanual robot piano playing constitutes a task that combines challenges from dynamic tasks, such as generating fast and precise motions, with slower but contact-rich manipulation problems. Although reinforcement learning-based approaches have shown promising results in single-task performance, these methods struggle in a multi-song setting. Our work aims to close this gap and, thereby, enable imitation learning approaches for robot piano playing at scale. To this end, we introduce the *Robot Piano 1 Million* (RP1M) dataset, containing bimanual robot piano playing motion data of more than one million trajectories. We formulate finger placements as an optimal transport problem, thus enabling automatic annotation of vast amounts of unlabeled songs. With RP1M, we train a multi-song piano playing policy with imitation learning approaches at scale, leveraging flow matching as the policy representation. Experiments show that our method obtains promising results.**

I. Introduction

Empowering robots with human-level dexterity is notoriously challenging. Robot piano playing combines various aspects of dynamic and manipulation tasks: the agent is required to coordinate multiple fingers to precisely press keys, which is a high-dimensional and rich control task. RoboPianist [14] is a simulated piano-playing environment that features two Shadow robot hands. Sheet music is represented by Musical Instrument Digital Interface (MIDI) transcription. Each time step in the MIDI file specifies which piano keys to press. RoboPianist uses human-annotated fingering information, telling which finger is supposed to press a particular piano key at each time step, to form a dense reward function for training RL agents. Since asking human pianists to annotate the fingering for each musical note is very expensive, Pianomime [8] replaces the human-annotated fingering with extracted hand motions from collected human piano playing videos on YouTube. To obtain high quality human hand motion data, FürElise [13] builds a data capture setup with five GoPro cameras placed around the piano to provide multi-view recordings of elite pianists' performances, reconstructs the motions with vision-based methods, and refines the reconstructed motions with inverse kinematics. However, relying on human annotations or demonstrations to train a piano-playing agent with RL restricts it to reproducing only human-labeled or demonstrated music pieces, yet many songs lack annotated fingerings or performance videos. Besides, those annotations may be infeasible for robots with morphologies different from human hands, such as different numbers of fingers or distinct hand dimensions. In addition, although RL-based approaches have shown promising results in single-task performance, they struggle in the multi-song setting [14]. The advent of scalable imitation learning (IL) techniques [2] enables representing complex and multi-modal distributions. So far, creating large datasets for robot piano playing is problematic due to the time-consuming fingering annotations. We propose an automatic fingering method that formulates the fingering problem as an optimal transportation problem [15]. It enables robots with different hand morphologies to play the piano given only MIDI files without any human demonstrations or annotations. The automatic fingering also allows learning piano playing with different embodiments, such as robots with four fingers only. We then collect *Robot Piano 1 Million dataset* (RP1M), which comprises the motion data of high-quality bi-manual robot piano play, by training RL agents for each of the 2k songs and rolling out each policy 500 times with different random seeds. With RP1M, we train a multi-song piano robotic playing policy for bi-manual dexterous robot hands using imitation learning at scale.

II. Method

Task setup. The simulated piano-playing environment is built upon RoboPianist [14]. The piano playing environment features a full-size keyboard with 88 keys driven by linear springs, two Shadow robot hands [9], and a pseudo sustain pedal. Sheet music is represented by Musical Instrument Digital Interface (MIDI) transcription. Each time step in the MIDI file specifies which piano keys to press (active keys). The goal of a piano-playing agent is to press active keys and avoid inactive keys under *space* and *time* constraints. The observation includes the state of robot hands, fingertip positions, piano sustain state, piano key states, and a goal vector. The action space consists of the robot hands' joint positions, forearms' positions, and a sustain pedal. We evaluate the performance of the trained agent with an average F1 score calculated by $F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$. For piano playing, recall and precision measure the agent's performance on pressing the active keys and avoiding inactive keys respectively [14].

Piano Playing with RL. We use RL to train specialist agents per song to control the bimanual dexterous robot hands to play the piano, without any human fingering labels or demonstrations. We frame the piano playing task as a finite MDP. At time step t, the agent $\pi_\theta(a_t|s_t)$, parameterized by θ, receives state s_t and takes action a_t to interact with the environment and receives new state s_{t+1} and reward r_t. The agent's goal is to maximize the expected cumulative rewards over an episode of length H, defined as $\mathcal{J} = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{H} \gamma^t r_t(s_t, a_t)\right]$, where γ is a discount factor ranging from 0 to 1.

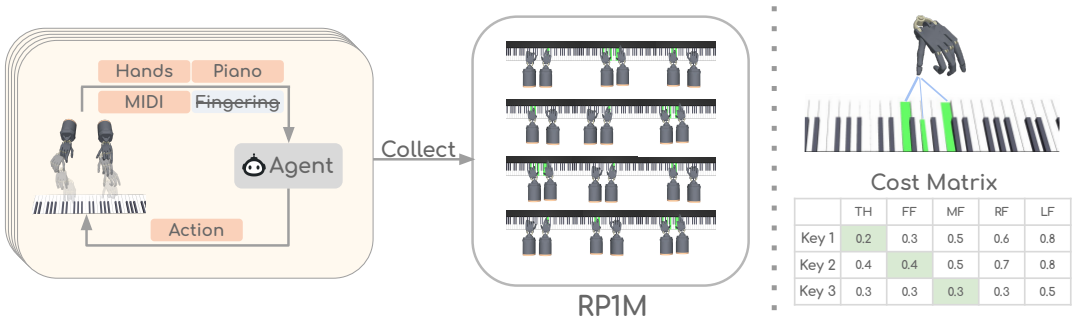

Fig. 1. Overview of RP1M. (**Left**) RP1M is a large-scale motion dataset for piano playing with bi-manual dexterous robot hands. The dataset includes $\sim$*1M* expert trajectories collected by $\sim$*2k* RL specialist agents. (**Right**) To collect a diverse motion dataset of playing sheet music, we lift the requirement of human-annotated fingering by formulating the finger placement as an optimal transport problem such that the robot hands play piano in an energy-efficient way.

Fingering Generation with Optimal Transport. Fingering is the assignment of fingers to notes, mapping which finger is supposed to press a particular piano key at each time step. Although fingering is highly personalized, generally speaking, it helps pianists to press keys timely and efficiently. Motivated by this, apart from maximizing the key pressing rewards, we also aim to minimize the moving distances of fingers. Specifically, at time step t, for the i-th key k^i to press, we use the j-th finger f^j to press this key such that the overall moving cost is minimized. We define the minimized cumulative moving distance as $d_t^{\text{OT}} \in \mathbb{R}^+$:

$$
\begin{aligned}
d_t^{\text{OT}} &= \min_{w_t} \sum_{(i,j)\in K_t \times F} w_t(k^i, f^j) \cdot c_t(k^i, f^j), \\
\text{s.t.,} \quad i) \ & \sum_{j\in F} w_t(k^i, f^j) = 1, \quad \text{for} \quad i \in K_t, \\
ii) \ & \sum_{i\in K_t} w_t(k^i, f^j) \le 1, \quad \text{for} \quad j \in F, \\
iii) \ & w_t(k^i, f^j) \in \{0, 1\}, \quad \text{for} \quad (i,j) \in K_t \times F.
\end{aligned} \tag{1}
$$

K_t represents the set of keys to press at time step t and F represents the fingers of robot hands. $c_t(k^i, f^j)$ represents the cost of moving finger f^j to piano key k^i at time step t calculated by their Euclidean distance. $w_t(k^i, f^j)$ is a boolean weight. It enforces that each key in K_t will be pressed by only *one* finger in F, and each finger presses *at most* one key. The constrained optimization problem in Eq. (1) is an optimal transport problem. Intuitively, it tries to find the best "transport" strategy such that the overall cost of moving (a subset of) fingers F to keys K_t is minimized. We solve this optimization problem with a modified Jonker-Volgenant algorithm [3] and use the optimal combinations (i^*, j^*) as the fingering for the agent. The fingering is calculated on the fly based on the hands' state, so during the RL training, the fingering adjusts according to the robot hands' state. We define a reward r_t^{OT} based on d_t^{OT} to encourage the agent to move the fingers close to the keys K_t:

$$
r_t^{\text{OT}} = \begin{cases} \exp(c \cdot (d_t^{\text{OT}} - 0.01)^2) & \text{if} \quad d_t^{\text{OT}} \ge 0.01, \\ 1.0 & \text{if} \quad d_t^{\text{OT}} < 0.01. \end{cases} \tag{2}
$$

c is a constant scale value, and we use the same value as Tassa et al. [12]. The overall reward function is defined as:

$$
r_t = r_t^{\text{OT}} + r_t^{\text{Press}} + r_t^{\text{Sustain}} + \alpha_1 \cdot r_t^{\text{Collision}} + \alpha_2 \cdot r_t^{\text{Energy}} \tag{3}
$$

r^{Press} and r_t^{Sustain} represent the reward for correctly pressing the target keys and the sustain pedal. $r_t^{\text{Collision}}$ encourages the agent to avoid collision between forearms and r_t^{Energy} prefers energy-saving behaviors.

Large-Scale Motion Dataset Collection. Removing the requirement of human fingering labels allows the agent to play any sheet music available on the Internet (under copyright license). We collect and release a large-scale motion dataset for piano playing, called *Robot Piano 1 Million* (RP1M) dataset. Our dataset includes $\sim$1M expert trajectories covering $\sim$2k musical pieces. For each musical piece, we train an individual RL agent with our method for 8 million environment steps and collect 500 expert trajectories with the trained agent. We chunk each sheet music every 550 time steps, corresponding to 27.5 seconds, so that each run has the same episode length. The sheet music used for training is from the PIG dataset [7] and a subset (1788 pieces) of the GiantMIDI-Piano dataset [5]. In Fig. 2, we show the statistics of our collected motion dataset. The top plot shows the histogram of the pressed keys. We found that keys close to the center are more frequently pressed than keys at the corner. Also, white keys, taking 65.7%, are more likely to be pressed than black keys. In the bottom left plot, we show the distribution of the number of active keys over all time steps. It roughly follows a Gaussian distribution, and 90.70% musical pieces in our dataset include 1000-4000 active keys. We also include the distribution of F1 scores of trained agents used for collecting data. We found most agents (79.00%) achieve F1 scores larger than 0.75, and 99.89% of the agents' F1 scores are larger than 0.5. The distribution of F1 scores reflects the quality of the collected dataset. We empirically found agents with F1 score ≥ 0.75 are capable of playing sheet music reasonably well with only minor errors. Agents with $\le$ 0.5 F1 scores usually have notable errors due to the difficulty of songs or the mechanical limitations.

Training a Multi-song Policy with IL. With RP1M, we can train a multi-song piano playing policy with imitation learning approaches at scale. Considering the difficulty of piano playing and the multimodality in the data distribution, we need a policy representation that is expressive enough to capture the complexity in the dataset. We formulate the robot behavioral cloning policy as a generative process using flow matching [6]. It constructs a flow vector that continuously transforms a source probability distribution into a target distribution. Instead of

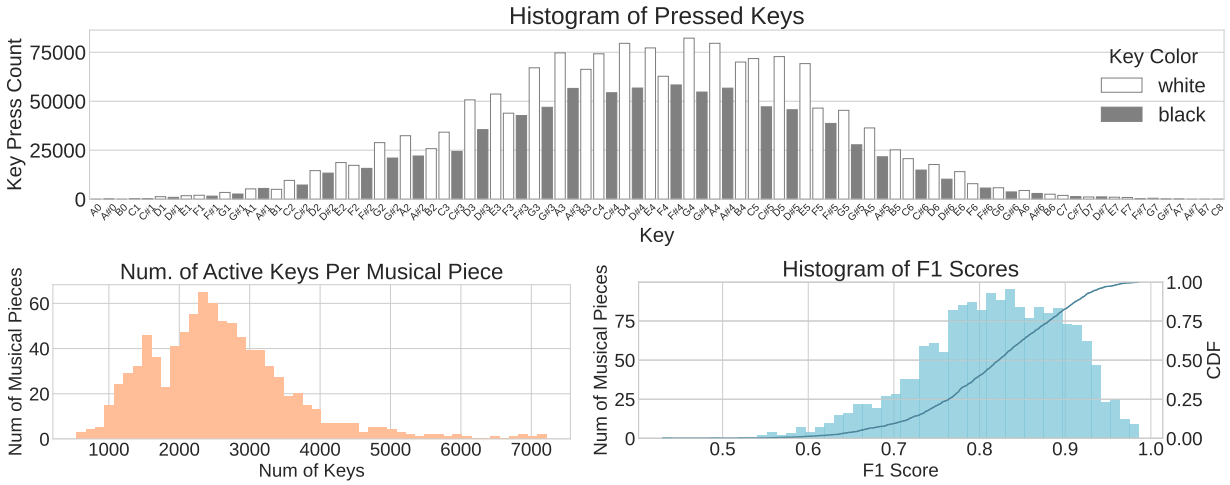

Fig. 2. Statistics of our RP1M dataset. (**Top**) Histogram of pressed keys in our RP1M dataset. (**Bottom Left**) Distribution of the number of active keys over all time steps. (**Bottom Right**) Distribution of F1 scores in our dataset.

relying on stochastic differential equations by introducing noise like DDPM [4], flow matching uses an ordinary differential equation to deterministically shape the data distribution. Similar to Diffusion Policy [2], we model the flow estimation conditioned on input observations and output the robot hand actions. We compare it with a simple MLP-based BC, and the Diffusion Policy with DDIM [11].

III. Experiments

A. Single-song Policy Learning

We compare our method to baselines with human fingering (*RoboPianist-RL*) and without human fingering (*No Fingering*). RoboPianist-RL includes human fingering in its inputs, and the fingering information is also used in the reward function to force the agent to follow this fingering. Our method, marked as *OT*, removes the fingering from the observation space and uses OT-based finger placement to guide the agent to discover its own fingering. The first two columns of Fig. 3 show that our method matches *RoboPianist-RL*'s performance on two different songs. Our method outperforms the baseline without human fingering by a large margin, showing that the proposed OT-based finger placement boosts agent learning. The proposed method works well even on challenging songs. We test our method on *Flight of the Bumblebee* and achieve a 0.79 F1 score after 3M training steps.

Analysis of the Learned Fingering. We compare the fingering discovered by the agent itself and the human annotations. In Fig. 4, we visualize the sample trajectory of playing *French Suite No.5 Sarabande* and the corresponding fingering. We found that although the agent achieves strong performance for this song (the second plot in Fig. 3), our agent discovers different fingering compared to humans. For example, for the right hand, humans mainly use the middle and ring fingers, while our agent uses the thumb and first finger. Furthermore, in some cases, human annotations are not suitable for the robot hand due to different morphologies. For example, in the second time step of Fig. 4, the human uses the first finger and ring finger. However, due to the mechanical limitation of the robot hand, it can not press keys that far apart with these two fingers, thus mimicking human fingering will miss one key. Instead, our agent discovered to use the thumb and little finger, which satisfies the hardware limitation and accurately presses the target keys.

Cross Emboidments. Labs usually have different robot platforms, thus having a method that works for different embodiments is highly desirable. We test our method on a different embodiment. To simplify the experiment, we disable the little finger of the Shadow robot hand and obtain a four-finger robot hand, which has a similar morphology to Allegro [1] and LEAP Hand [10]. We evaluate the modified robot hand on the song French Suite No.5 Sarabande (first 550 time steps), where our method achieves a 0.95 F1 score, similar to the 0.96 achieved with the original robot hands. In the bottom row of Fig. 4, we visualize the learned fingering with four-finger hands. The agent discovers different fingering compared to humans and the original hands but still accurately presses active keys, meaning our method is compatible with different embodiments.

B. Multi-song Policy Learning

The objective here is to train a single multi-task policy capable of playing various music pieces on the piano.

Design Choices of Input Observation. We try different options for input observation. We trained several MLP-based policies on 12 songs with different designs of observation and evaluated their in-distribution performance (F1 scores on songs included in the training data). As shown in Fig. 5, when we only include the goal, piano state, and hand joints of the current step in the observation, the agent performs the worst. After we add fingertip positions in the observation, we obtain an improvement on the average F1 score. We additionally add 3 steps of future goals in the observation and the performance improves again, showing that it is reasonable to give the agent some lookahead information. We then increase the number of steps of future goals to 10 and obtain the best performance.

Comparing Different Policy Representations. We first

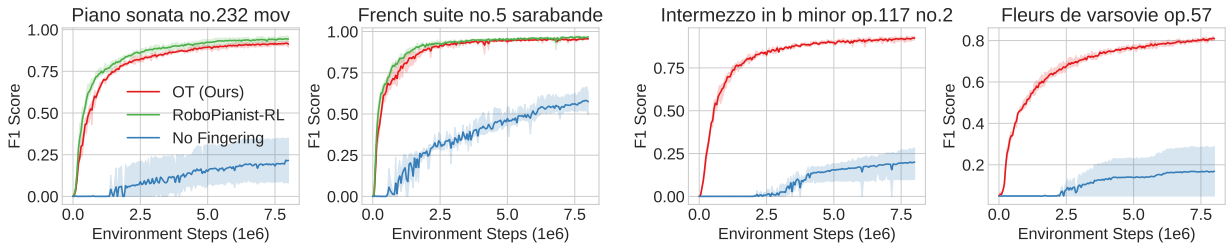

Fig. 3. Comparison of the RL performance with our OT fingering, human-annotated fingering, and no fingering. Our method matches the performance of RoboPianist-RL, which is trained with human fingering. We also outperforms the baseline without any fingering information by a large margin. The plots show the mean over 3 random seeds and the shaded areas represent the 95% confidence interval.

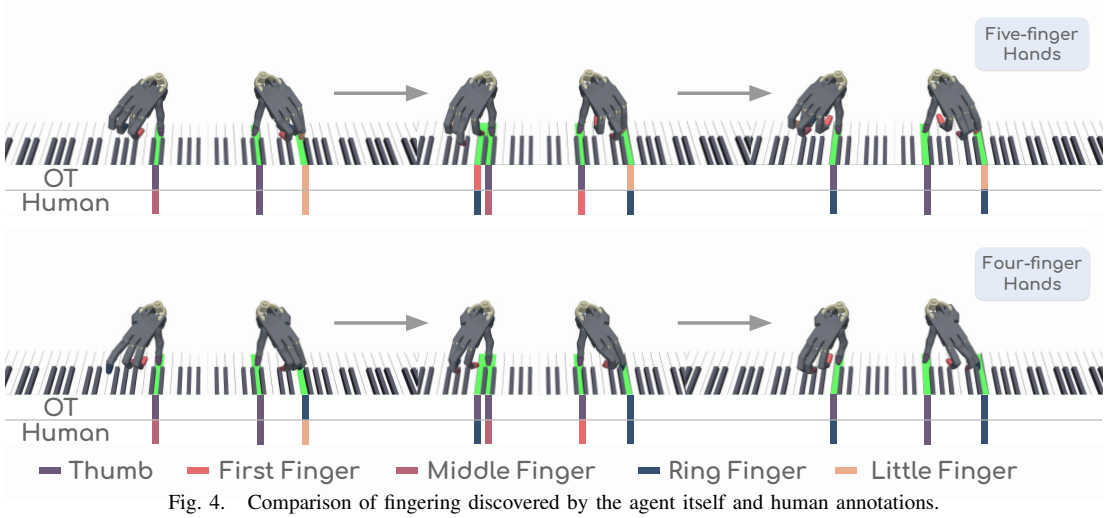

Fig. 4. Comparison of fingering discovered by the agent itself and human annotations.

TABLE I
COMPARING DIFFERENT POLICY REPRESENTATIONS ON 150 SONGS.

	In-Distribution	Out-of-Distribution
MLP	0.3377	0.2372
DDIM	0.6649	0.2934
FM	**0.6839**	**0.3221**

take 12 songs from the RP1M dataset to train the policies and evaluate their in-distribution performance. As shown in Fig. 6, flow matching policy works the best compared with Diffusion Policy with DDIM. The MLP-based method performs the worst since it is not expressive enough. We then evaluate them with a larger scale of data. We take 150 songs from the RP1M dataset, randomly sample 20% trajectories from each song, and use them to train the multi-song policy with imitation learning. We evaluate its in-distribution performance (on 20 songs) and its generalization ability (F1 scores on 5 out-of-distribution songs). From Table I we can see that flow matching policy works the best on both in-distribution and out-of-distribution songs. The performance of Diffusion policy with DDIM is close to flow matching policy, while the MLP-based method is much worse.

IV. CONCLUSION

We propose a novel automatic fingering annotation approach based on optimal transport, with which we can train specialist agents with RL on any music piece given only the MIDI file. It allows us to scale up the number of expert policies and collect a large-scale motion dataset named RP1M for piano playing with bimanual dexterous robot hands. With RP1M, we train

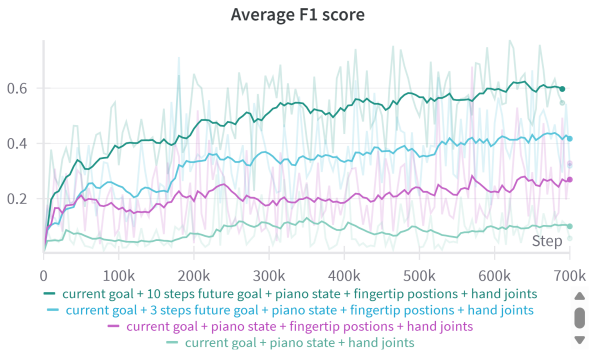

Fig. 5. Different design choices of observation.

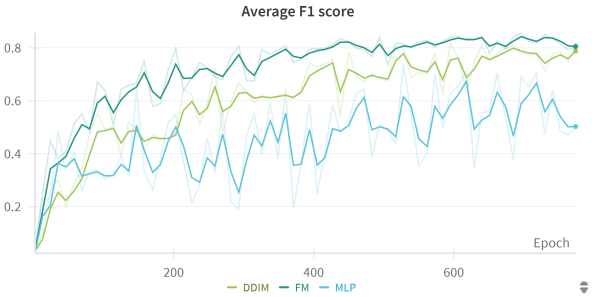

Fig. 6. Comparing different policy representations on 12 songs.

a multi-song piano playing policy with imitation learning at scale, leveraging flow matching as the policy representation. Experiments show that our method obtains promising results on both in-distribution and out-of-distribution evaluations.

REFERENCES

[1] Allegro. https://www.wonikrobotics.com/research-robot-hand.

[2] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[3] David F Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[5] Qiuqiang Kong, Bochen Li, Jitong Chen, and Yuxuan Wang. GiantMIDI-Piano: A large-scale midi dataset for classical piano music. *arXiv preprint arXiv:2010.07061*, 2020.

[6] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[7] Eita Nakamura, Yasuyuki Saito, and Kazuyoshi Yoshii. Statistical learning and estimation of piano fingering. *Information Sciences*, 517:68–85, 2020.

[8] Cheng Qian, Julen Urain, Kevin Zakka, and Jan Peters. Pianomime: Learning a generalist, dexterous piano player from internet demonstrations. 2024.

[9] ShadowRobot. ShadowRobot Dexterous Hand. https://www.shadowrobot.com/products/dexterous-hand/, 2005.

[10] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

[11] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[12] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[13] Ruocheng Wang, Pei Xu, Haochen Shi, Elizabeth Schumann, and C Karen Liu. Fürelise: Capturing and physically synthesizing hand motion of piano performance. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[14] Kevin Zakka, Philipp Wu, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, et al. RoboPianist: Dexterous piano playing with deep reinforcement learning. In *7th Annual Conference on Robot Learning*, 2023.

[15] Yi Zhao, Le Chen, Jan Schneider, Quankai Gao, Juho Kannala, Bernhard Schölkopf, Joni Pajarinen, and Dieter Büchler. Rp1m: A large-scale motion dataset for piano playing with bi-manual dexterous robot hands. 2024.