

# FürElise: Capturing and Physically Synthesizing Hand Motions of Piano Performance

RUOCHENG WANG<sup>†</sup>, Stanford University, USA

PEI XU<sup>†</sup>, Stanford University, USA

HAOCHEN SHI, Stanford University, USA

ELIZABETH SCHUMANN, Stanford University, USA

C. KAREN LIU, Stanford University, USA

Piano playing requires agile, precise, and coordinated hand control that stretches the limits of dexterity. Hand motion models with the sophistication to accurately recreate piano playing have a wide range of applications in character animation, embodied AI, biomechanics, and VR/AR. In this paper, we construct a first-of-its-kind large-scale dataset that contains approximately 10 hours of 3D hand motion and audio from 15 elite-level pianists playing 153 pieces of classical music. To capture natural performances, we designed a markerless setup in which motions are reconstructed from multi-view videos using state-of-the-art pose estimation models. The motion data is further refined via inverse kinematics using the high-resolution MIDI key-pressing data obtained from sensors in a specialized Yamaha Disklavier piano. Leveraging the collected dataset, we developed a pipeline that can synthesize physically-plausible hand motions for musical scores outside of the dataset. Our approach employs a combination of imitation learning and reinforcement learning to obtain policies for physics-based bimanual control involving the interaction between hands and piano keys. To solve the sampling efficiency problem with the large motion dataset, we use a diffusion model to generate natural reference motions, which provide high-level trajectory and fingering (finger order and placement) information. However, the generated reference motion alone does not provide sufficient accuracy for piano performance modeling. We then further augmented the data by using musical similarity to retrieve similar motions from the captured dataset to boost the precision of the RL policy. With the proposed method, our model generates natural, dexterous motions that generalize to music from outside the training dataset.

**CCS Concepts:** • Computing methodologies → Animation; Physical simulation; Reinforcement learning.

**Additional Key Words and Phrases:** Character animation, hand animation, physics-based control, dexterous control, motion capture dataset

## ACM Reference Format:

Ruocheng Wang<sup>†</sup>, Pei Xu<sup>†</sup>, Haochen Shi, Elizabeth Schumann, and C. Karen Liu. 2024. FürElise: Capturing and Physically Synthesizing Hand Motions of Piano Performance. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24)*, December 3–6, 2024, Tokyo, Japan. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3680528.3687703>

---

<sup>†</sup> These two authors contributed equally to this work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan*

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-1131-2/24/12...\$15.00  
<https://doi.org/10.1145/3680528.3687703>

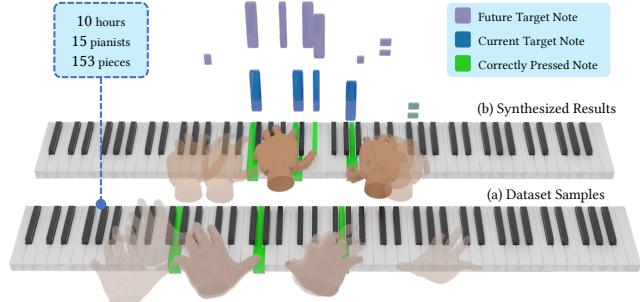


Fig. 1. Our paper (a) collects the first large-scale 3D hand motion dataset of piano playing, accompanied by synchronized audio and key pressing events; (b) proposes a method that can control a physically simulated hand to play novel pieces ‘unheard’ from the training set.

## 1 INTRODUCTION

Physically synthesizing human motion has a wide range of applications in character animation, embodied AI, AR/VR, robotics, and biomechanics. Researchers have made great strides in simulating functional and realistic human movements which enable digital agents to physically navigate and interact with environments while maintaining balance. As the application domain expands, the next frontier in human motion synthesis is to create digital agents that not only achieve motion tasks, but also exhibit elite-level athletic techniques and musical precision, comparable to the peak performance of human athletes and musicians. In this work, we take the first step toward synthesis of human peak performance through the lens of the movement of elite pianists.

Piano playing is a demanding motor skill that requires impeccable precision in finger control to press the correct keys at the correct time, agile coordination to press multiple keys simultaneously, and remarkable dexterity to fluidly play long sequences while anticipating upcoming notes. Previous works on simulating piano playing motions either rely on human-annotated fingering information (which finger to press which key) [Zakka et al. 2023] or are limited to scenarios involving easier compositions [Xu et al. 2022; Zhu et al. 2013]. We believe that a better model requires a deeper understanding of how humans play the piano. However, there is a significant shortfall in large-scale datasets that adequately capture the diversity and complexity of piano performances.

To address this gap, we design and build a comprehensive, non-intrusive data capture pipeline to record the 3D hand motions of pianists during their natural performances. This pipeline employs a markerless setup, where multi-view videos are processed using

state-of-the-art pose estimation model [Pavlakos et al. 2024] to reconstruct 3D motions. These reconstructions are further refined through inverse kinematics, utilizing music information obtained from sensors embedded in a specialized piano [Yamaha 2024]. Using this pipeline, we have collected the first large-scale dataset of piano motions, FürElise, capturing approximately 10 hours of 3D hand motions from 15 professional or conservatory pianists performing 153 pieces of classical music across various genres. This dataset encompasses a broad spectrum of piano skills demonstrated by elite pianists, and includes synchronized audio, providing a valuable resource for character animation and dexterous control. It also enables various music-related applications such as keyboard ergonomics, music pedagogy, and pianist injury prevention.

Leveraging FürElise, we take a step towards synthesizing physically simulated motions of piano playing for novel pieces of music “unheard” from the dataset. Specifically, given a piece of sheet music, our approach first uses a diffusion model trained on the collected dataset to generate an initial reference motion that provides high-level trajectory guidance and fingering information. However, this initial reference motion often includes numerous incorrect or missing keys, making it unsuitable for training an RL policy that would ensure musically correct physical interactions between hand fingers and the piano keys. We propose to enhance this process by combining a music-based motion retrieval method with the diffusion model to create an ensemble of reference motions, thus balancing the visual performance and physical plausibility for accurate key press.

Our experiments show that, given a piece of music unseen in the training dataset, our method can synthesize natural piano motions. The policy can handle chords, fast wrist motions, and other complex piano skills, playing melodious pieces given only the sheet music. Ablations have shown that the diffusion model, music-based retrieval and reinforcement learning all contribute to the performance of the final model.

In summary, this paper makes two major contributions toward physics-based synthesis of elite-level piano performance, as illustrated in Figure 1:

- We present the first large-scale dataset of 3D hand motions in piano performance with synchronized audio.
- We develop a model that combines diffusion models, motion retrieval, and reinforcement learning to synthesize natural dexterous motions playing a diverse set of piano music pieces. Our model was evaluated through extensive experiments and ablations.

## 2 RELATED WORK

### 2.1 Music2Motion

The problem of generating motions following music has been extensively studied in recent years. Alexanderson et al. [2023]; Li et al. [2021]; Tseng et al. [2023] tackle the problems of generating whole-body dancing motions from input music using diffusion models. Another line of research trains neural networks to generate upper-body motions of musicians from the audio of various instruments [Chen et al. 2023a; Kao and Su 2020; Li et al. 2018; Liu et al. 2020; Shlizerman et al. 2018]. These works typically utilize pose estimation models to estimate 3D joint locations only from

monocular videos, resulting in poor motion quality due to depth ambiguity. Moreover, these works focus on learning to generate visually plausible kinematics motions, overlooking their physical plausibility. In contrast, our work collects a large-scale high-quality dataset of piano performance motion. We propose a pipeline to train control policies that can play the novel piano pieces in a physically simulated environment.

Apart from data-driven approaches, some early works design heuristics to animate hands for music performance. Zhu et al. [2013] generates piano playing motion by using iterative optimizations to solve for hand trajectories that hit target keys and satisfy predefined constraints. ElKoura and Singh [2003] considers generating left-hand motions for playing the guitar by retrieving and blending motions from a motion capture dataset. In these works, a key challenge is to determine the fingering information, which specifies which finger should press each note. Previous works rely on heuristic cost functions or additional annotations to decide fingering, which can only handle simple or manually pre-processed pieces. Our work uses a generative model trained on a large-scale dataset to provide fingering information automatically for reinforcement learning policies to learn playing unseen pieces.

### 2.2 Physics-Based Dexterous Control

Studying the control strategy for physically simulated dexterous hands has wide applications in computer graphics, robotics, and biomechanics. Traditional approaches usually rely on trajectory optimization and/or human-designed heuristic rules to perform control[Chen et al. 2023b; Liu 2008, 2009; Mordatch et al. 2012; Wang et al. 2013; Ye and Liu 2012]. Most recent works on physics-based dexterous control only focus on single-hand scenarios and do not have high precision requirements[Andrychowicz et al. 2020; Liu 2009; Xie et al. 2023; Yang et al. 2022; Zhang et al. 2021; Zhao et al. 2013]. In this study, we focus on piano playing, a task that requires simultaneous bimanual control with exceptional temporal and spatial precision.

Piano playing is a common but intricate physical activity in daily life. Introducing physics can help generate physically feasible motions for piano playing. Algorithms are proposed to train policies to play piano in simulations using anthropomorphic robot hands [Xu et al. 2022; Zakka et al. 2023] via reinforcement learning. Due to the complexity of the task, Xu et al. [2022] only considers one hand playing on a simplified piano. Zakka et al. [2023] leverages human annotated fingering information (which finger should press which key) to facilitate policy training. Our proposed pipeline, once trained on our large-scale dataset, can play unseen pieces without any additional annotation.

Our approach follows previous work leveraging reinforcement learning to synthesize motions under the framework of imitation learning [Merel et al. 2017; Peng et al. 2022, 2021; Xu and Karamouzas 2021; Xu et al. 2023]. Though impressive results are achieved in generating realistic motions by imitation learning, it is still a challenging problem to perform learning efficiently from a very large set of reference motions. To better utilize our collected large set of piano-playing motions, we address the problem by developing a hybrid approach to generate and retrieve motions for the policy to synthesize.

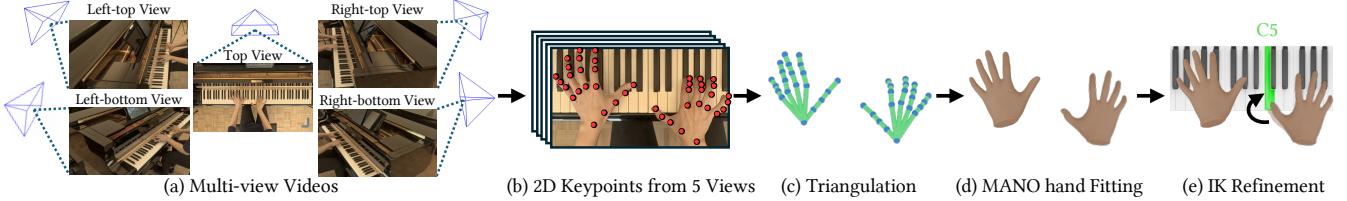


Fig. 2. Overview of our pipeline to reconstruct motion data from multi-view videos. We (a) shoot 4K videos from 5 different views at 59.94 FPS using RGB camera; (b) detect 2D keypoints of the hands from each view; (c) triangulate the 2D keypoints into 3D hand skeletons with calibrated camera intrinsics and extrinsics; (d) fit the skeleton onto MANO hand meshes [Romero et al. 2017]; and (e) run IK with ground-truth MIDI as end effector goals to refine the finger placements for correct key pressing.



Fig. 3. Data capture setup. Five GoPro cameras are placed around the piano to provide multi-view recordings of elite pianists' performances.

### 2.3 Hand Motion Datasets

Various hand motion datasets are collected in different scenarios such as grasping [Chao et al. 2021; Taheri et al. 2020], object manipulation [Fan et al. 2023; Wang et al. 2024], two-hand interactions [Moon et al. 2020]. However, few datasets capture the hand motions of piano performance, which are more complex and dynamics. Some works [Grauman et al. 2024; Simon et al. 2017] provide piano playing hand motions reconstructed using pose estimation models, but the pieces played are limited and there is no audio information recorded, which constrains their applications for tasks like MIDI-conditioned motion generation and data refinement. Wu et al. [2023] uses OptiTrack to reconstruct 3D hand motions of piano playing along with audio recorded in the format of Musical Instrument Digital Interface (MIDI). However, they only collected 11 pieces of music with limited variations. Our dataset contains 10 hours of 3D motions from 15 elite pianists playing 153 different classical compositions, which cover a wide range of piano skills. All motions are provided along with MIDI audio accurately recorded by the piano's built-in recorder.

## 3 DATASET

To study hand motions during piano playing, we collect a large-scale dataset, FürElise, with approximately 10 hours of 3D hand motions paired with synchronized audio. In this section, we will

first elaborate on the data capture and processing pipeline and provide an analysis of the dataset.

### 3.1 Data Capture

We aim to collect a large-scale dataset of piano playing motion performed by professional and conservatory-level pianists with minimal intrusion.

*Device Setup.* We record the data in a typical piano studio familiar to the performers, as shown in Figure 3. To minimize the influence of capture device, we design a markerless setup using multiview RGB cameras. Five calibrated GoPro cameras are placed around a grand piano to record synchronized videos and audio with 59.94 FPS. All the videos have a resolution of  $3840 \times 2160$ . The grand piano is a Yamaha Disklavier DS7X ENPRO, which has a built-in recorder to record the key and pedal pressing events during the performance with high precision in MIDI format, from which the original audio with high fidelity can be reproduced.

*Vision-based Motion Reconstruction.* Figure 2 summarizes the motion reconstruction process. We first use the state-of-the-art pose estimation model HaMeR [Pavlakos et al. 2024] to predict the hand pose  $K_{2D} \in \mathbb{R}^{N \times 5 \times 2 \times 21 \times 2}$ , which are the 2D locations of 21 joints on each hand from all 5 camera views for a sequence of  $N$  frames. While HaMeR can generate 3D meshes of MANO hands [Romero et al. 2017] in the camera space, we found that the predicted depths are not usable due to severe inaccuracy. As such, we only leverage the projected 2D keypoints from HaMeR and compute 3D locations of each joint  $K_{3D} \in \mathbb{R}^{N \times 2 \times 21 \times 3}$  via triangulation. RANSAC is used to filter out occluded keypoints, while a Butterworth filter is applied to every joint to enhance temporal smoothness, since HaMeR only considers one frame at a time. Next, we fit MANO hand parameters  $\Theta = \{\theta, \beta, t\}$  to obtain 3D hand meshes for every frame, where  $\theta \in \mathbb{R}^{N \times 2 \times 16 \times 3}$ ,  $\beta \in \mathbb{R}^{2 \times 45}$ ,  $t \in \mathbb{R}^{N \times 2 \times 3}$  are the joint rotations, shape parameters and global translations of the two hands. The shape parameters are computed with extra hand calibration videos. Other parameters are optimized by minimizing the mean-squared error between the triangulated joint locations and MANO hand joint locations.

*MIDI-based Motion Refinement.* Vision-based motion reconstruction achieves reasonable results, but visible artifacts such as incorrect key-pressing or missing keys are quite common in the reconstructed motion. To improve the quality, we utilize the key-press

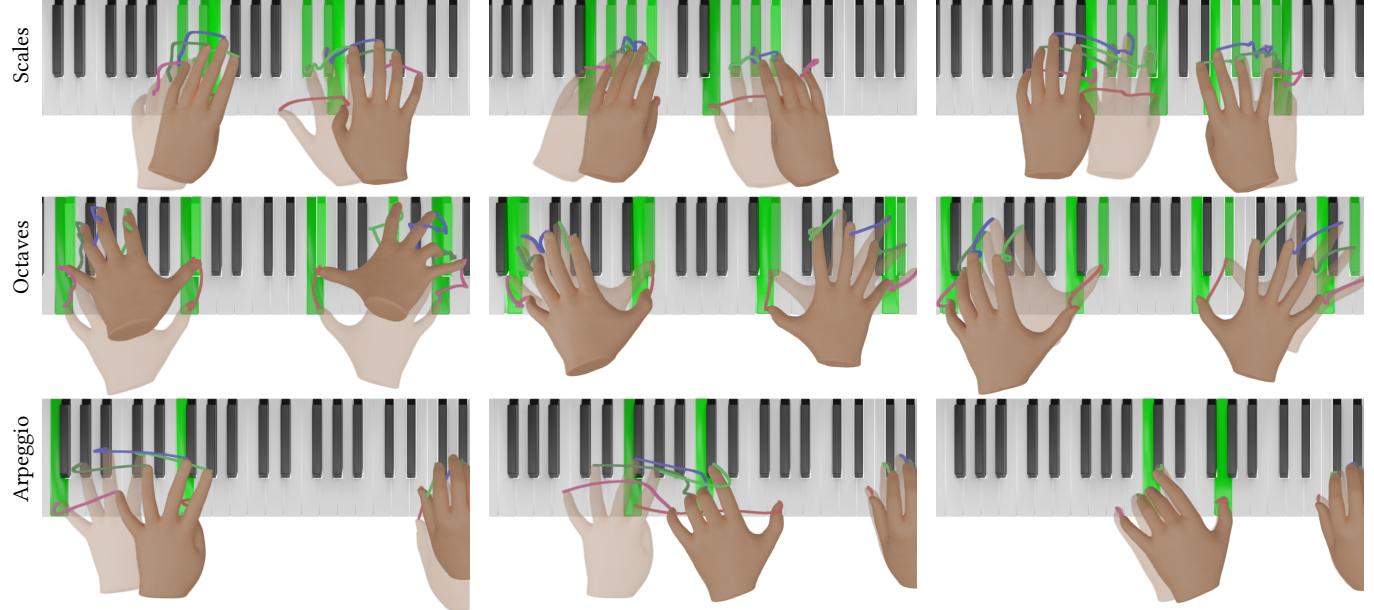


Fig. 4. Examples of some piano skills in our dataset, including scales, octaves, and arpeggio. The trajectory of each fingertip is visualized. The green keys show the pressed keys through the trajectory.

information stored in the accompanying MIDI file. For each note played during the session, the MIDI file records the precise moments each key is fully pressed down and released. By assuming that the fingertip remains in contact with the key throughout the duration of the note, we can infer the positions of fingertips based on the states of the keys. Therefore, we apply inverse kinematics to ensure two key properties of the reconstructed motion: a) when a key is being pressed according to the MIDI file, at least one fingertip must be on the top surface of that key and have a depth below a preset threshold to trigger sound; b) when a key is not pressed according to the MIDI file, no fingertip should press the key deep enough to trigger the note; To prevent large modifications by IK, we only optimize the local joint rotations and the wrist orientation of each hand. We also limit the maximum change of fingertips to 1cm. A smoothness term is added to prevent abrupt changes between frames. Further details can be found in the appendix.

### 3.2 Dataset Analysis

*Data statistics.* We collect and reconstruct a total of 10 hours of 3D hand motions paired with synchronized MIDI. 8 male and 7 female elite pianists contribute a total of 153 classical compositions in various genres.

*Quality Evaluation.* Following Zakka et al. [2023], we use precision, recall and F1 to quantitatively evaluate the quality of our reconstructed motions according to the recorded MIDI:

$$\begin{aligned} \text{Precision}_i &= \frac{TP_i}{FP_i + TP_i} & \text{Recall} &= \frac{TP_i}{TP_i + FN_i} \\ F1_i &= \frac{2\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}, \end{aligned} \quad (1)$$

where  $TP_i$  computes the number of keys that are correctly pressed,  $FP_i$  computes the number of keys that are wrongly pressed, and  $FN_i$  computes the number of keys that the motion failed to press. We do this for every frame  $i$  and average over all the frames in the dataset. To extract the pressed keys from reconstructed motions, similar to the IK procedure mentioned earlier, any fingertip horizontally over a key and below a preset threshold is treated as pressing the key. Using this evaluation protocol, we got a precision of 88.55, a recall of 92.53, and an F1 of 86.49 on the whole dataset. We also visualize our reconstructed motion and include the audio of the extracted MIDI in the supplementary video.

*Qualitative Examples.* To demonstrate the diversity of motions in our dataset, we show examples of various primitive piano playing skills [Neuhaus 2008] in Figure 4.

## 4 PLAY PIANO WITH PHYSICALLY SIMULATED HANDS

Leveraging the collected dataset, we aim to train a policy that controls two physically simulated hands in concert to play a given piece of music. Thus, the input to our method is a musical score represented as a list of notes  $\{O_i = (t_i^{\text{start}}, t_i^{\text{end}}, p_i) \mid i \in \{1, 2, \dots, n\}\}$ , where  $t_i^{\text{start}}, t_i^{\text{end}} \in \mathbb{R}$  is the start and end time of the note, and  $p_i \in \{1, \dots, 88\}$  is the pitch, which can also be mapped to one of the 88 piano keys. Our method finally outputs a policy that controls two hands interacting with a piano keyboard physically. A digital sound is generated by matching the pitch of the keys being pressed by the physically simulated hands.

We propose a method that combines data-driven and physics-based approaches to achieve the goal (Figure 5). A diffusion motion model [Ho et al. 2020] is trained on the FürElise dataset to generate kinematics motions for the given piece. Despite the strong abilities

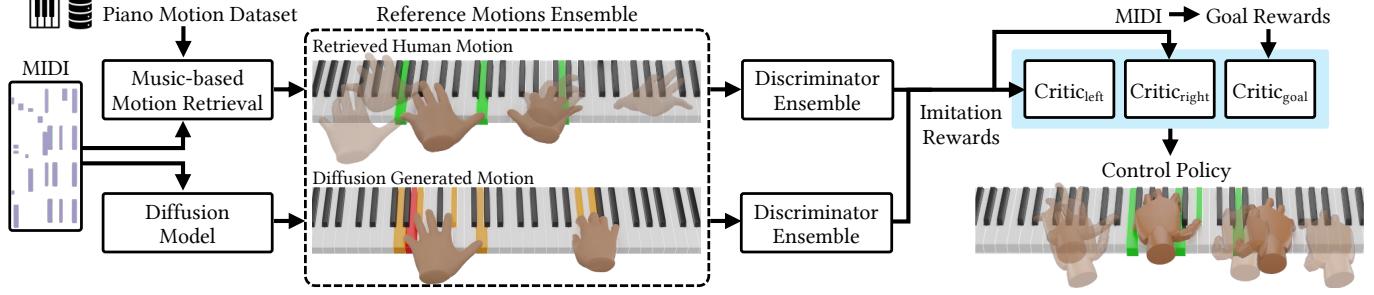


Fig. 5. Overview of our method to physically simulate piano performance from a given sheet music. We use MIDI to retrieve motion data from the collected motion dataset and as input to a diffusion model for generating piano performance motions. These two sets of motions are combined into a reference motion ensemble. Utilizing the reference motions, we then employ two discriminator ensembles and three critics, which consider imitation and goal rewards, respectively, to train a control policy via reinforcement learning.

of diffusion models to generate visually plausible motions when trained on large datasets, they often produce physically implausible artifacts, such as penetration, floating, and inconsistent interactions with objects [Liu and Yi 2024; Yuan et al. 2023]. We observe similar issues in our settings where the generated motions exhibit seemingly plausible wrist trajectories and hand poses, but frequently exhibit incorrect contact with the keyboard, such as pressing the wrong keys or failing to press the right keys. Directly applying reinforcement learning to imitate these flawed motions would lead to unsuccessful policies. As such, we propose a music-based motion retrieval method to utilize the high-quality motions in the FürElise dataset. Combining the diffusion-generated and retrieved motions together, we form an ensemble of natural or precise reference trajectories to train an RL policy that minimizes the goal-based reward and imitates the reference motions [Xu et al. 2023].

#### 4.1 Diffusion Model

The goal of this module is to generate a kinematic hand trajectory given a piece of sheet music. We leverage a diffusion model, which is proven to be very effective in modeling distributions of human motions [Alexanderson et al. 2023; Li et al. 2023; Tevet et al. 2023; Tseng et al. 2023] to perform kinematic motion generation.

*Overview.* The core of the diffusion model [Ho et al. 2020] trains a denoiser network on dataset examples corrupted by different levels of Gaussian noises with the objective function reconstructing the original clean examples. The loss function for a conditional diffusion model is as follows:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}, t} [\|\mathbf{x} - \hat{\mathbf{x}}_{\theta}(\mathbf{x}_t, t, \mathbf{c})\|^2], \quad (2)$$

where  $\mathbf{x}$  are the clean examples,  $\mathbf{x}_t$  is the corrupted examples on noise level  $t$ ,  $\mathbf{c}$  is the condition vector. After training, conditional samples can be drawn by running the denoiser network iteratively on a trajectory of Gaussian noises.

*Motion Representation.* Since the task requires high precision for the location of fingertips, we represent the dual-hand motion as a trajectory of  $2 \times 21$  joint locations  $\mathbf{K} \in \mathbb{R}^{M \times 2 \times 21}$  defined in MANO hands [Romero et al. 2017], similar to [Liu and Yi 2024].  $M$  is the number of frames considered for the diffusion model. Here we use

$M = 120$  which corresponds to a window of 120 frames. To ensure consistent bone lengths during generation, we fit MANO hand models to the generated trajectory with fixed shape parameters to achieve the final predicted joint locations.

*Condition Representation.* To compute the condition vector  $\mathbf{c}_T$ , we first quantize input sheet music  $\{O_i = (t_i^{\text{start}}, t_i^{\text{end}}, p_i) \mid i \in \{1, 2, \dots, n\}\}$  into a binary matrix  $C \in \{0, 1\}^{N \times 88}$ , where  $N$  is the total number of frames in the input music and  $n$  is the total number of notes. Then, we divide each non-zero entry in the matrix by the duration of the corresponding key being pressed:

$$C_{i,p} = \frac{1}{t_{ip} - t_{ip}^{\text{start}} + 1}. \quad (3)$$

In this way, the key information, as well as the duration information, are encoded into the condition vector  $\mathbf{c}_T \in \mathbb{R}^{88}$ .

*Model Architecture.* We leverage a transformer-based architecture proposed in EDGE [Tseng et al. 2023] to train our model. In accordance with our dataset, motions and music are quantized into 59.94FPS. The diffusion model, therefore, generates 2 seconds by outputting the results of 120 frames at a time.

*Long-form Generation.* Although we train the diffusion model on a window of 2 seconds, we can generate arbitrary long sequences from conditions by denoising a batch of sequences while enforcing the adjacent sequences in the batch share an overlapping path, following [Tseng et al. 2023].

#### 4.2 Music-Based Motion Retrieval

To complement the diffusion-generated motions, we retrieve additional reference motions from the whole dataset for reinforcement learning policy to perform imitation learning. To do so, first, we quantize all notes in the dataset  $\{O_i = (t_i^{\text{start}}, t_i^{\text{end}}, p_i) \mid i \in \{1, 2, \dots, n\}\}$  into a binary matrix  $M \in \{0, 1\}^{N \times 88}$  that align with the frames of hand motions. We perform the same quantization for the input piece to obtain binary a matrix  $M' \in \{0, 1\}^{N' \times 88}$ . Next, we compute a sliding window of length 30 and stride 1 individually over  $M$  and  $M'$  to obtain  $W \in \{0, 1\}^{N_w \times 30 \times 88}$ ,  $W' \in \{0, 1\}^{N'_w \times 30 \times 88}$ .  $N_w$  and  $N'_w$  are the numbers of windows for the dataset and the input piece. We then compute matching from windows in the target piece

to those of the dataset by minimizing their L2 distances:

$$\mathbf{c}_j = \arg \min_{i \in \{1, 2, \dots, N_w\}} \|\mathbf{W}_i - \mathbf{W}'_j\|_2 \quad \forall j \in \{1, 2, \dots, N'_w\} \quad (4)$$

This produces  $N'_w$  windows of musical pieces from the dataset. We then retrieve the corresponding hand motions, merge the overlapping windows, and generate a list of reference motions. These motions are combined with the diffusion-generated motions to train the policy more effectively.

#### 4.3 Policy Training for Physics-based Control

We set up our simulation environment using IsaacGym [Makoviychuk et al. 2021]. While the simulation runs at 240 FPS, the control runs at 60 FPS which is consistent with our diffusion model. Our physics-based hand models are modified from [Kumar and Todorov 2015] with geometry optimized according to the mocap subjects. Each hand has 17 links with 27 degrees of freedom (DoFs) driven by PD servos, where the wrist has 6 DoFs, the MCP joints have 2 DoFs except that the thumb MCP has 3, and all the PIP and DIP joints have 1 DoF. This leads to an action space of  $\mathbf{a}_t \in \mathbb{R}^{2 \times 27}$  for two hands. Similar to our diffusion model, we take the key-based binary vector as the goal representation for key pressing. To balance the goal vector size and the observation horizon, we utilize a compressed representation by merging the same key-pressing goal in consecutive frames into one. We take the future five merged goals as the goal state with an additional timer variable that indicates the time (in terms of the number of simulation frames) left for the associated key-pressing goal. Thus the final goal state vector is of shape  $\mathbf{g}_t \in \mathbb{R}^{5 \times (88+1)}$ . To perform control, we take a 2-frame historical observation composed of the position, orientation, and linear and angular velocities of all the links of two hands. This results in a pose state vector  $\mathbf{s}_t \in \mathbb{R}^{2 \times 2 \times 208}$  for two hands.

Due to the limited performance of the motion generated by the diffusion model, we do not directly perform motion tracking during the control policy training. Rather, we take the generated and retrieved motions as the reference simultaneously, and perform imitation learning using reinforcement learning with a GAN-like architecture [Xu and Karamouzas 2021] for motion synthesis. Following the previous literature [Xu et al. 2023], to utilize the reference motions more effectively, we decouple the motions of two hands and employ two discriminators at the same time for motion imitation of the left and right hand respectively. By doing so, the pose of one hand does not rely on that of the other hand anymore. We, thereby, facilitate the single-hand motion imitation by performing learning independently rather than using a dual-hand state space. The imitation-related reward is computed by

$$r_t^{\text{imit}, h}(\mathbf{s}_t^h, \mathbf{s}_{t+1}^h) = \frac{1}{N} \sum_{n=1}^N \text{CLIP}\left(D_n^h(\mathbf{s}_t^h, \mathbf{s}_{t+1}^h), -1, 1\right), \quad (5)$$

where  $h \in \{L, R\}$  indicates the imitation of the left and right hand respectively,  $\mathbf{s}_t^h$  is the pose state of the single hand  $h$ , and  $D_n^h$  is the discriminator trained using hinge loss [Lim and Ye 2017].

To encourage expected key-pressing behaviors, besides imitation, we also employ a goal-based reward function to evaluate the policy's key-pressing performance at each time step  $t$ . The reward definition is different depending on the pressing condition of each key.

We assume that a key  $k$  is pressed to generate sound if the pressed distance  $p_k$  is greater than 90% of that key's maximal travel distance  $d_k$ , which is defined using the allowed rotation range of that key. For each target key  $k$  that needs to be pressed, we have the reward term to encourage the correct key-pressing behavior:

$$r_{t,k}^+ = \begin{cases} 1 & \text{if } p_k/d_k > 0.9 \\ \exp(||\mathbf{p}_i - \mathbf{p}_k|| + 0.01p_k/d_k) & \text{otherwise,} \end{cases} \quad (6)$$

where  $\mathbf{p}_i$  is the global position of the target fingertip  $i$ , and  $\mathbf{p}_k$  is the target position of the key. To determine the target fingertip, We extract fingering information based on the nearest finger to that key in the diffusion-generated motion. The target position of the key is obtained using the surface center of a key horizontally and the 85% position along the key's length axis vertically.

For each non-target key  $\kappa$ ,  $r_{t,\kappa}^-$  measures the errors of key pressing and is employed to penalize incorrect key-pressing behaviors:

$$r_{t,\kappa}^- = \begin{cases} p_\kappa/0.9d_\kappa & \text{if key } \kappa \text{ is touched and } p_\kappa/d_\kappa > 0.1 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

To emulate a physical piano generating clear sound, we perform penalization even if the key is assumed not to trigger any sound virtually (i.e.  $p_\kappa/d_\kappa < 0.9$ ) but ignore trivial touch (i.e.  $p_\kappa/d_\kappa < 0.1$ ). However, in difficult scenarios, key touching cannot be completely avoided. To prevent the policy from achieving a lower error of  $r_{t,k}^-$  by not touching any key, an additional reward term is introduced to encourage correct key pressing behaviors even if some non-target keys are touched.

The overall goal-driven reward is defined as

$$r_t = \prod_k r_{t,k}^+ - 0.15 \sum_\kappa r_{t,\kappa}^- + 0.5r_{\text{correct}} - 0.05r_{\text{energy}}, \quad (8)$$

where  $r_{\text{correct}} = 1$  if all target keys are pressed correctly or 0 otherwise, and  $r_{\text{energy}}$  is a term measuring the energy consumption based on the average linear velocity of fingers and wrists between two frames:

$$r_{\text{energy}} = \exp\left(-0.75 \sum_{h \in \{L, R\}} \left(||\mathbf{v}_w^h|| + 0.1 \sum_i ||\mathbf{v}_i^h||\right)^2\right), \quad (9)$$

where  $\mathbf{v}_w^h$  is the velocity of one hand's wrist in the global space,  $\mathbf{v}_i^h$  is the average velocity of each fingertip in the local system defined by its corresponding wrist joint.

The policy is trained using a multi-objective framework [Xu et al. 2023] to optimize

$$\max \mathbb{E}_t \left[ \sum_i w_i \bar{A}_{t,i} \pi(\mathbf{a}_t | \mathbf{g}_t, \mathbf{s}_t) \right], \quad (10)$$

where  $\bar{A}_{t,i}$  is the standardized advantage that is estimated according to the achieved reward of each objective  $i$ , and  $w_i$  is an associated weight. In our case, we have three objectives (two imitation objectives of left and right hand respectively, and one goal-driven objective). To encourage the policy to perform expected key-pressing behaviors, the associated weights are 0.9 for the goal-driven objective and 0.05 for each imitation objective. Please refer to the supplementary materials for the hyperparameters.

## 5 EXPERIMENTAL RESULTS

We evaluate our method quantitatively on 14 pieces of music using the F1 score. We also conduct numerous ablation studies to analyze the impact of each component in our algorithm. Our dataset and method are best evaluated in the supplementary video with the audio turned on.

### 5.1 Setup

**Data.** We use 14 sheets of music to test our proposed pipeline. Although most recorded compositions in our dataset are classical, we include a wider range of genres including popular music, and jazz unseen during training. Because the chosen music pieces are very long with repetition, we select a clip of music from each piece and use it to train our model. The lengths of the clips are in the range from 14.4 to 28.94 seconds and 20.72 seconds on average. We do not modify the speed of the original music.

**Metrics.** Similar to our data quality evaluation, we record the key-pressing states of model predictions and compare them with the input sheet music. Precision, recall, and F1 scores are computed for each frame and averaged over the whole piece. For diffusion-generated motions, we use the same heuristics used in data quality evaluation to extract the pressed keys: when a fingertip is below a preset depth and horizontally over a key, we treat the key as pressed. For physics-based policy, we directly query the key-pressing states from the physical simulator.

**Implementation Details.** For diffusion models, we train with a window of 120 frames (2 seconds). The training takes around 1 day on 2 NVIDIA A5000 GPUs. We train a single diffusion model for all the testing compositions. Policy trained with reinforcement learning takes around 1–3 days depending on the difficulty of studied music pieces on a single A5000 GPU and consumes about  $2 \times 10^8$  to  $4 \times 10^8$  training samples.

### 5.2 Diffusion Generated Motions

We first show qualitative results in Figure 7. The diffusion models can generate natural and plausible kinematic trajectories on unseen pieces if viewed from a top-down perspective. However, the model cannot press keys accurately. The generated motions frequently float above the keys without pressing them or press the wrong keys, as shown in Figure 8. These observations resonate with other works using diffusion models on whole-body motion and hand-object interactions [Liu and Yi 2024; Yuan et al. 2023]. The observations are also supported by the quantitative results in Figure 6. More visualizations are shown in the supplementary video.

### 5.3 Full Pipeline

Quantitative results of our full pipeline are summarized in Figure 6: the policy outperforms the diffusion model by a large margin. As shown in Figure 7, the policy can handle large wrist motions (Fig 7f), chords (pressing multiple keys at the same time, Fig 7abc), double notes (pressing different pairs of notes sequentially, Fig 7d), as well as arpeggios (pressing individual notes of a chord in sequence, Fig 7e). Despite the average F1 scores being as high as more than 0.8 for all the tested songs, the policy still could perform unexpected key

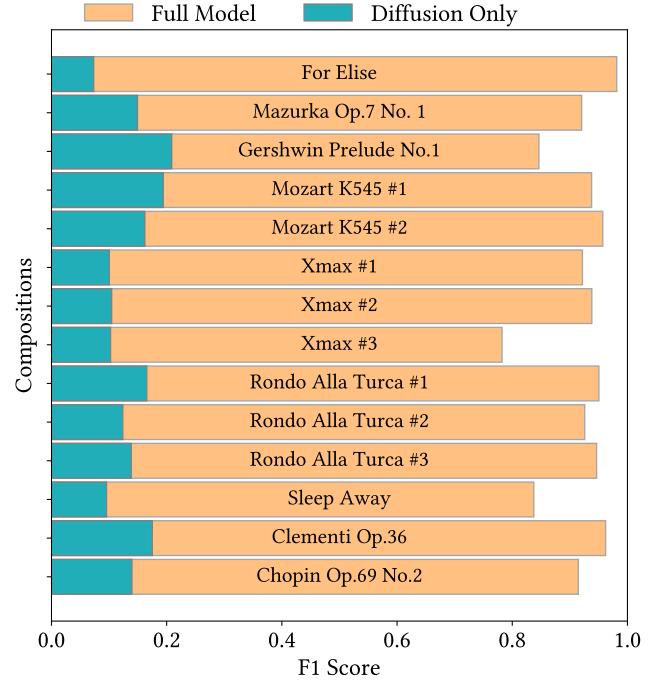


Fig. 6. F1 scores of the diffusion model and our policy on the 14 test pieces. RL policies have a significant improvement over diffusion-generated motions across all 14 pieces.

pressing, though lasting for a very short duration. This sometimes leads to a negative impact on humans' auditory perception more than what F1 scores can reflect.

### 5.4 Ablations

To understand the effect of using an ensemble of motions generated by the diffusion model and those retrieved from the dataset as the reference for the control policy to learn, we design the following ablation studies tested on four music pieces:

- *RL+Retr*. The policy is trained with only the reference motion retrieved from the dataset.
- *RL+Diff*. The policy is trained with only the reference motion generated by the diffusion model.
- *RL Only*. The policy is trained only using the goal-driven reward without motion imitation.
- *RL+Whole*. The policy is trained only using the whole motion dataset as the reference for imitation without motions generated by the diffusion model.

**Results.** The performance of each model is listed in Table 1. The training curve is shown in Figure 9. The full model outperforms the ablative models by a large margin in all the tested cases. We show qualitative comparisons visually of the studied ablative models in Figure 10. As we can see, the *RL* only case performs the worst and behaves in a manner not human-like, which highlights the necessity of using motion imitation to ensure the motion naturalness and to help better key-pressing task execution. When the policies are trained

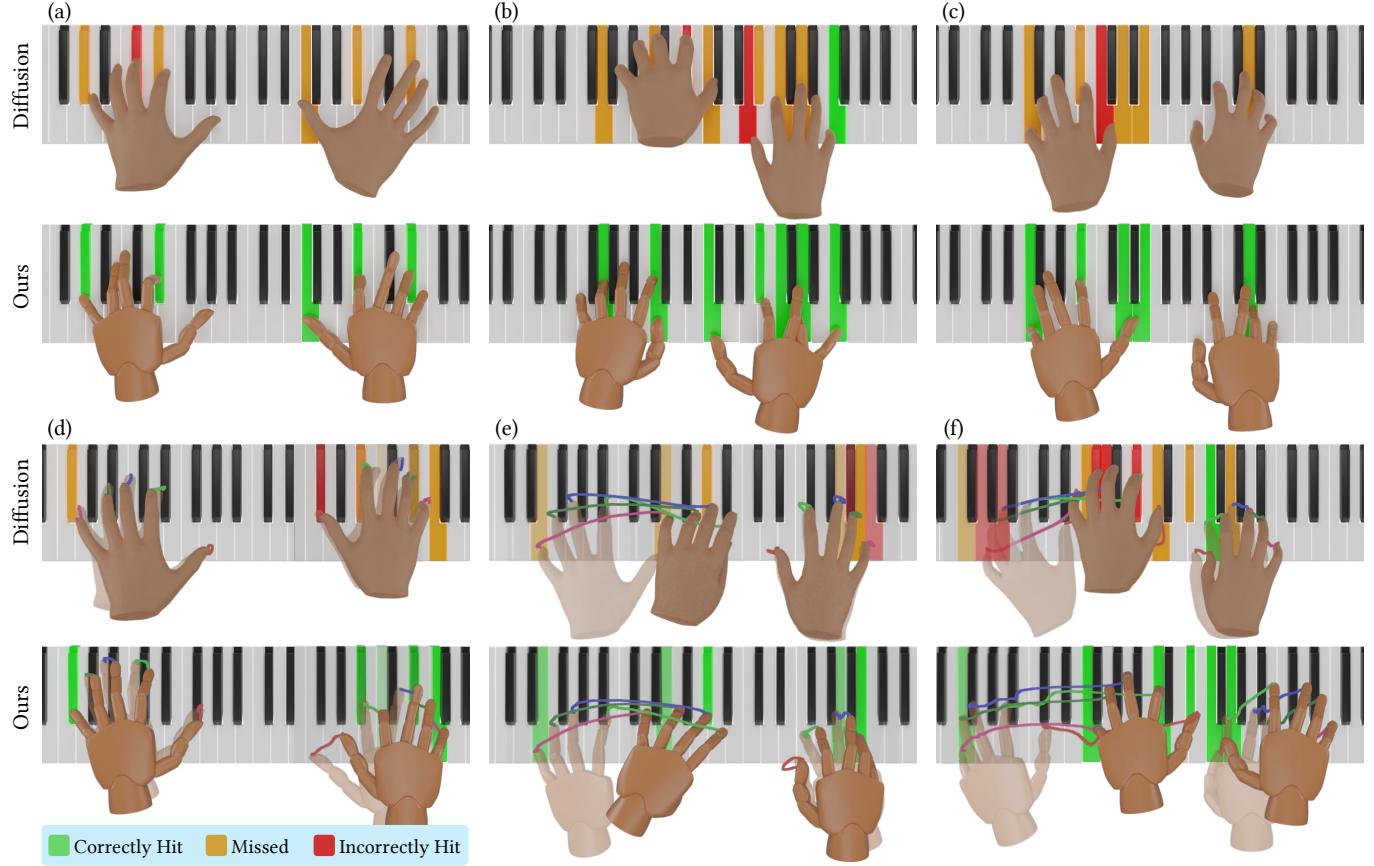


Fig. 7. Results of our model accompanied with generated diffusion motions. The trajectory of each fingertip is visualized. The simulated hands can correctly press all the target keys while trying to follow the diffusion-generated motion. The policy can handle chords (abc), double notes (d), and large wrist movements (ef) naturally and accurately.

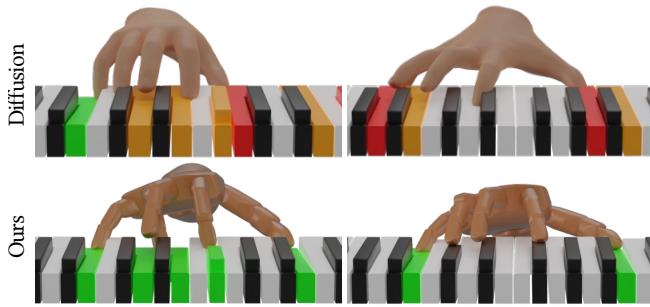


Fig. 8. Comparison between motions generated by the diffusion model and our full model with reinforcement learning. Our full model fixes the imprecise key-pressing issue of the diffusion model. In the left demo, by imitating retrieved motions, the control policy learns to use the ring finger to press two keys at the same. This pose is not provided by the diffusion-generated motions.

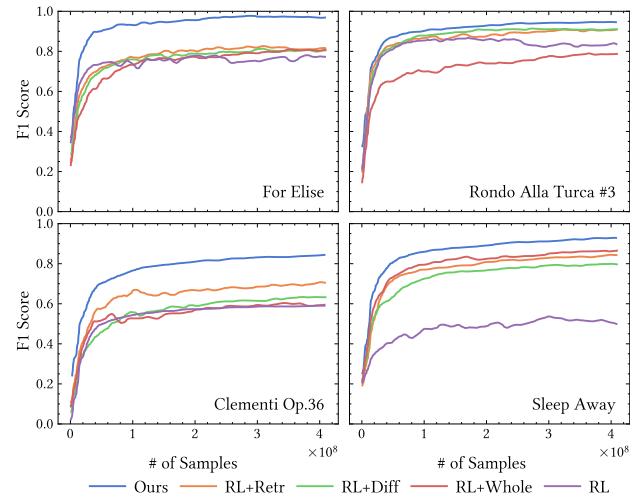


Fig. 9. Learning performance of our full model and the ablative models. In all the tested cases, our model shows better performance compared to the ablative models.

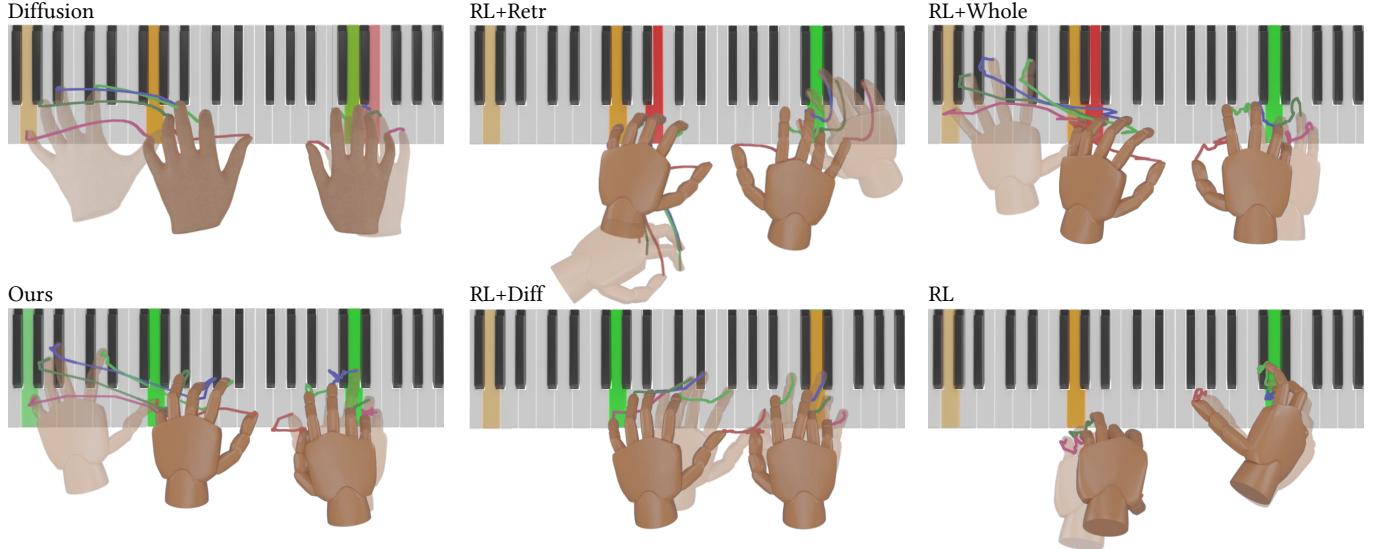


Fig. 10. Comparisons between our full model and the ablative models. Without diffusion guidance, the ablative models *RL*, *RL+Retr*, and *RL+Whole* have excessive or unnatural movements because no motions corresponding to the given piece are provided during training. Without retrieved motions from the dataset (*RL+Diff*), the model tends to overfit the imprecise motions generated by the diffusion models, resulting in lower accuracy.

Table 1. We study 4 ablations of our method on 4 different pieces. We show that the F1 score of our method is significantly higher than all the variants.

	Ours	RL+Retr	RL+Diff	RL+Whole	RL
For Elise	<b>98.15</b>	78.17	85.77	80.37	73.20
Rondo Alla Turca #3	<b>94.65</b>	81.94	78.73	88.51	34.36
Clementi Op.36	<b>96.21</b>	95.00	94.17	79.10	75.28
Sleep Away	<b>83.75</b>	53.33	64.13	49.38	49.97

without diffusion-generated motions (*RL+Retr* and *RL+Whole*), they yield unnatural hand poses due to the lack of fingering information. The policies also tend to have redundant motions during playing in this case because during training they could try to imitate some unrelated motions that may not strictly apply to the input music piece. When the model is only trained with diffusion-generated motions (*RL+Diff*), the policy tends to overfit the erroneous finger placements existing in the diffusion-generated motions and thus has lower accuracy of key pressing. Those results demonstrate that the diffusion model and motion retrieval are complimentary and both of them are crucial to the final performance of our pipeline. Additionally, in supplementary materials, we qualitatively compare the motions generated by our control policies to those in our dataset when facing the same target notes.

## 6 CONCLUSION

We present a first-of-its-kind large-scale dataset of 3D hand motion and audio of piano performance. Our dataset, FürElise, contains 8 hours of performance from 11 elite-level pianists playing 98 pieces of classical music. Leveraging FürElise, we propose a physics-based method to synthesize accurate piano playing motion for music outside the training dataset. We evaluate our method through extensive experiments and ablations.

Our work takes the first step toward motion synthesis of human peak performance using data collected from musicians for unseen songs. However, there is still a significant gap between the skill level our model achieves and that of human pianists. Several limitations in our current work might contribute to this gap. First, our method does not consider sound amplitude, a critical element in music performance. Consequently, our current model generates music with constant amplitude. However, the key-pressing velocity, which determines amplitude, is recorded in our dataset and can be utilized for future work. Second, we let the model determine fingering, resulting in policies that may struggle with some basic skills such as finger crossover. Future work could incorporate high-level, common fingering rules to facilitate policy learning. Moreover, we leverage F1 scores to evaluate performance averaged over each frame, which may not align well with humans' auditory perception, as humans could be sensitive to some transient errors that contribute little to F1 scores such as breaking a chord or inconsistent tempo. Developing a better audio evaluation metric that meets humans' perceptions would be a great direction for future work. Finally, while the simulated hand models have a reasonably accurate kinematic structure, they can exert unnaturally large joint torques or generate infeasible acceleration. A promising future direction is to consider a realistic hand musculoskeletal model that generates motion through muscle activation, providing a computational tool for biomechanics studies and injury prevention.

## ACKNOWLEDGMENTS

We thank Yifeng Jiang and Jiaman Li for providing detailed feedback on the paper. This work was supported in part by the Wu-Tsai Human Performance Alliance, Stanford Institute for Human-Centered Artificial Intelligence and Roblox. We thank the 15 pianist volunteers

for their essential contributions to this study. To protect their privacy, they remain unnamed, but their participation was invaluable to our research.

## REFERENCES

- Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. 2023. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–20.
- OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research* 39, 1 (2020), 3–20.
- Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. 2021. DexYCB: A benchmark for capturing hand grasping of objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Jiali Chen, Changjie Fan, Zhimeng Zhang, Gongzheng Li, Zeng Zhao, Zhigang Deng, and Yu Ding. 2023a. A Music-Driven Deep Generative Adversarial Model for Guzheng Playing Animation. *IEEE Transactions on Visualization and Computer Graphics* 29, 2 (2023), 1400–1414.
- Sirui Chen, Albert Wu, and C Karen Liu. 2023b. Synthesizing Dexterous Nonprehensile Pregrasp for Ungraspable Objects. In *ACM SIGGRAPH 2023 Conference Proceedings*, 1–10.
- George ElKoura and Karan Singh. 2003. Handrix: Animating the Human Hand. (2003), 110–119.
- Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. 2023. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 12943–12954.
- Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Booté, Eugene Byrne, Zachary Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, María Escobar, Cristhian Forrigua, Abrham Kahsay Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Dutt Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh K. Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiao Hu, Yifei Huang, Yiming Huang, Wenslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mingjing Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatuminu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbeláez, Gedas Bertasius, David J. Crandall, Dima Damen, Jakob Julian Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard A. Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. 2024. Ego-Exo4D: Understanding Skilled Human Activity from First- and Third-Person Perspectives. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 6840–6851.
- Hsuan-Kai Kao and Li Su. 2020. Temporally guided music-to-body-movement generation. In *Proceedings of the 28th ACM International Conference on Multimedia*. 147–155.
- Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, and Yuxuan Wang. 2020. High-Resolution Piano Transcription With Pedals by Regressing Onset and Offset Times. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2020).
- Vikash Kumar and Emanuel Todorov. 2015. Mujoco haptix: A virtual reality system for hand manipulation. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 657–663.
- Bochen Li, Akira Maezawa, and Zhiyao Duan. 2018. Skeleton Plays Piano: Online Generation of Pianist Body Movements from MIDI Performance.. In *ISMIR*. 218–224.
- Jiaman Li, Jiajun Wu, and C Karen Liu. 2023. Object motion guided human motion synthesis. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–11.
- Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. 2021. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 13401–13412.
- Jae Hyun Lim and Jong Chul Ye. 2017. Geometric gan. *arXiv preprint arXiv:1705.02894* (2017).
- C Karen Liu. 2008. Synthesis of interactive hand manipulation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 163–171.
- C Karen Liu. 2009. Dextrous manipulation from a grasping pose. In *ACM SIGGRAPH 2009 papers*. 1–6.
- Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming* 45, 1 (1989), 503–528.
- Jun-Wei Liu, Hung-Yi Lin, Yu-Fen Huang, Hsuan-Kai Kao, and Li Su. 2020. Body Movement Generation for Expressive Violin Performance Applying Neural Networks. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), 3787–3791.
- Xueyi Liu and Li Yi. 2024. GeneOH Diffusion: Towards Generalizable Hand-Object Interaction Denoising via Denoising Diffusion. In *International Conference on Learning Representations (ICLR)*.
- Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. 2021. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. *arXiv:2108.10471 [cs.RO]*
- Josh Merel, Yuval Tassa, Dhruva TB, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess. 2017. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201* (2017).
- Gyeongsik Moon, Shouo-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. 2020. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 548–564.
- Igor Mordatch, Zoran Popović, and Emanuel Todorov. 2012. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*. 137–144.
- Heinrich Neuhaus. 2008. *The art of piano playing*. Kahn and Averill.
- Georgios Pavlakos, Dandan Shan, Ilijia Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. 2024. Reconstructing Hands in 3D with Transformers.
- Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Trans. Graph.* 41, 4, Article 94 (2022).
- Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. *ACM Trans. Graph.* 40, 4, Article 144 (2021).
- Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied hands: modeling and capturing hands and bodies together. *ACM Trans. Graph.* 36, 6, Article 245 (nov 2017), 17 pages. <https://doi.org/10.1145/3130800.3130883>
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv:1707.06347 [cs.LG]*
- Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. 2018. Audio to body dynamics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 7574–7583.
- Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand keypoint detection in single images using multiview bootstrapping. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. 2020. GRAB: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 581–600.
- Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bernman. 2023. Human Motion Diffusion Model. In *International Conference on Learning Representations (ICLR)*. <https://openreview.net/forum?id=SJ1k5yOzjwu>
- Jonathan Tseng, Rodrigo Castellon, and C. Karen Liu. 2023. EDGE: Editable Dance Generation From Music. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. 2024. DexCap: Scalable and Portable Mocap Data Collection System for Dexterous Manipulation. *arXiv preprint arXiv:2403.07788* (2024).
- Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. 2013. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–14.
- Erwin Wu, Hayato Nishioka, Shinichi Furuya, and Hideki Koike. 2023. Marker-removal Networks to Collect Precise 3D Hand Data for RGB-based Estimation and its Application in Piano. In *Winter Conference on Applications of Computer Vision (WACV)*. 2977–2986.
- Zhaoming Xie, Jonathan Tseng, Sebastian Starke, Michiel van de Panne, and C Karen Liu. 2023. Hierarchical planning and control for box loco-manipulation. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6, 3 (2023), 1–18.
- Huazhe Xu, Yuping Luo, Shaowiong Wang, Trevor Darrell, and Roberto Calandra. 2022. Towards learning to play piano with dexterous hands and touch. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10410–10416.
- Pei Xu and Ioannis Karamouzas. 2021. A GAN-Like Approach for Physics-Based Imitation Learning and Interactive Character Control. *Proc. of the ACM on Computer Graphics and Interactive Techniques* 4, 3 (2021).
- Pei Xu, Xiumin Shang, Victor Zordan, and Ioannis Karamouzas. 2023. Composite Motion Learning with Task Control. *ACM Transactions on Graphics* 42, 4 (2023).

- Yamaha. 2024. Yamaha Disklavier Pianos. [https://usa.yamaha.com/products/musical\\_instruments/pianos/disklavier/index.html](https://usa.yamaha.com/products/musical_instruments/pianos/disklavier/index.html).
- Zeshi Yang, Kangkang Yin, and Libin Liu. 2022. Learning to use chopsticks in diverse gripping styles. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–17.
- Yuting Ye and C Karen Liu. 2012. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics (ToG)* 31, 4 (2012), 1–10.
- Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. 2023. Physdiff: Physics-guided human motion diffusion model. In *International Conference on Computer Vision (ICCV)*. 16010–16021.
- Kevin Zakka, Philipp Wu, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, et al. 2023. Robopianist: Dexterous piano playing with deep reinforcement learning. In *Conference on Robot Learning (CoRL)*.
- H Zhang, Y Ye, T Shiratori, and T Komura. 2021. ManipNet: neural manipulation synthesis with a hand-object spatial representation. *ACM Transactions on Graphics (TOG)* 32, 6 (2021), 1–12.
- Yuanfeng Zhu, Ajay Sundar Ramakrishnan, Bernd Hamann, and Michael Neff. 2013. A system for automatic animation of piano performances. *Computer Animation and Virtual Worlds* 24, 5 (2013), 445–457.

## A DATA CAPTURE DETAILS

We refine our reconstructed motions using the MIDI recorded during the motion capture to obtain audio-synchronized motions. All the MIDI files were recorded by the piano's built-in recorder with very high accuracy.

### A.1 MIDI synchronization

Since the MIDI and each video are recorded separately by the piano and cameras, we perform a synchronization procedure to align them temporally. We first use Kong et al. [2020] to transcribe the audio of the video to MIDI format. Then, we iterate a list of candidate offsets and find the offset where the audio and the MIDI have the maximum number of notes matched. Two notes  $(t_0^{\text{start}}, t_0^{\text{end}}, p_0)$  and  $(t_1^{\text{start}}, t_1^{\text{end}}, p_1)$  are treated to be matched if they have the same pitch and  $\|t_0^{\text{start}} - t_1^{\text{start}}\| \leq 0.016$ . We then manually fine-tune the offset by aligning the pressing motions of fingers and the start time of the corresponding note.

### A.2 MIDI-based Inverse Kinematics.

To improve the quality of the reconstructed motions, we perform inverse kinematics (IK) based on the key-pressing information provided by the MIDI files. We first compute the pressed keys of the motions by the following heuristics: when any fingertip is horizontally over a key and its depth is below a preset threshold, we treat that key as being pressed. We then compare the extracted pressed keys with the key-pressing information from the recorded MIDI. Frames, where the extracted pressed keys are different from the ground-truth MIDI, are considered inaccurate and the corresponding hand poses will be fixed by IK. We consider two possible cases of inaccurately reconstructed hand poses: (1) a muted key is wrongly pressed by any finger and (2) an activated key is omitted by all fingers for pressing.

*Wrongly pressed keys.* For wrongly pressed keys, when multiple fingertips are pressing it, we select the fingertip with the lowest depth as the IK subject. The IK target is set such that the culprit's fingertip will move out of the key at a minimum distance.

*Omitted keys.* For keys that all fingertips fail to press, we first find the fingertip closest to the key by projecting all fingertips onto the surface of the key and assume the one with minimum distance to the projected location as the one performing pressing as well as the IK subject. We then set the IK target to the projected point.

We invalidate IK targets that need to move the fingertips for more than 1cm, and set up IK by minimizing the following loss function for every frame:

$$\mathcal{L}(\Theta^t)_{\text{ik}} = \frac{1}{\sum_{i=1}^{10} I_i^t} \sum_{i=1}^{10} I_i^t \|p_i^t - \hat{p}_i(\Theta^t)\|^2, \quad (1)$$

where:

- $t$  is the index of the frame;
- $I_i^t$  is the mask for the  $i$ -th tip, with  $I_i^t = 1$  if the tip is to be included in the IK and  $I_i^t = 0$  otherwise;
- $p_i^t$  is the target position of the  $i$ -th tip;

Table S1. Hyperparameters for Model Training

Parameter	Value
<i>Diffusion Model Training</i>	
learning rate	0.0004
batch size	512
training epochs	100
<i>Reinforcement Learning</i>	
policy network learning rate	$5 \times 10^{-6}$
critic network learning rate	$1 \times 10^{-4}$
discriminator learning rate	$1 \times 10^{-5}$
reward discount factor ( $\gamma$ )	0.95
GAE discount factor ( $\lambda$ )	0.95
surrogate clip range ( $\epsilon$ )	0.2
gradient penalty coefficient ( $\lambda^{GP}$ )	10
number of PPO workers (simulation instances)	512
PPO replay buffer size	4096
PPO batch size	256
PPO optimization epochs	5
discriminator replay buffer size	8192
discriminator batch size	512

- $\hat{p}_i(\Theta^t)$  is the position of the  $i$ -th tip given the hand parameters  $\Theta$ ; and
- $\Theta^t$  represents the hand parameters (pose and shape and translation) in the MANO model.

Since IK is performed only on frames with wrong key-pressing results, we further add a smoothing term to ensure temporal consistency:

$$\mathcal{L}_{\text{smooth}}(\Theta) = \frac{1}{N-1} \sum_{t>1}^N (\|\Theta^{t-1} - \Theta^t\|_2^2), \quad (2)$$

The final loss is computed by:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(\Theta^t)_{\text{ik}} + \lambda \mathcal{L}_{\text{smooth}}, \quad (3)$$

where  $N$  is the total number of frames in the dataset and  $\lambda = 0.0005$ . During optimization, we only optimize the local pose parameter and freeze other parameters, using L-BFGS [Liu and Nocedal 1989] optimizer iteratively for 100 epochs.

## B HYPERPARAMETERS

The hyperparameters used for diffusion model training and reinforcement learning are listed in Table S1. We employ PPO [Schulman et al. 2017] as our backbone reinforcement learning algorithm.

## C ADDITIONAL RESULTS

Here, we include a qualitative comparison between the motions generated by our control policies and those in our dataset when facing the same target notes in Figure S1. There are often multiple ways to perform the same target notes. Our pipeline enables the policy to either imitate motions generated by the diffusion model or from the captured dataset, resulting in diverse piano-playing patterns. The synthesized motions can be distinct from human pianists

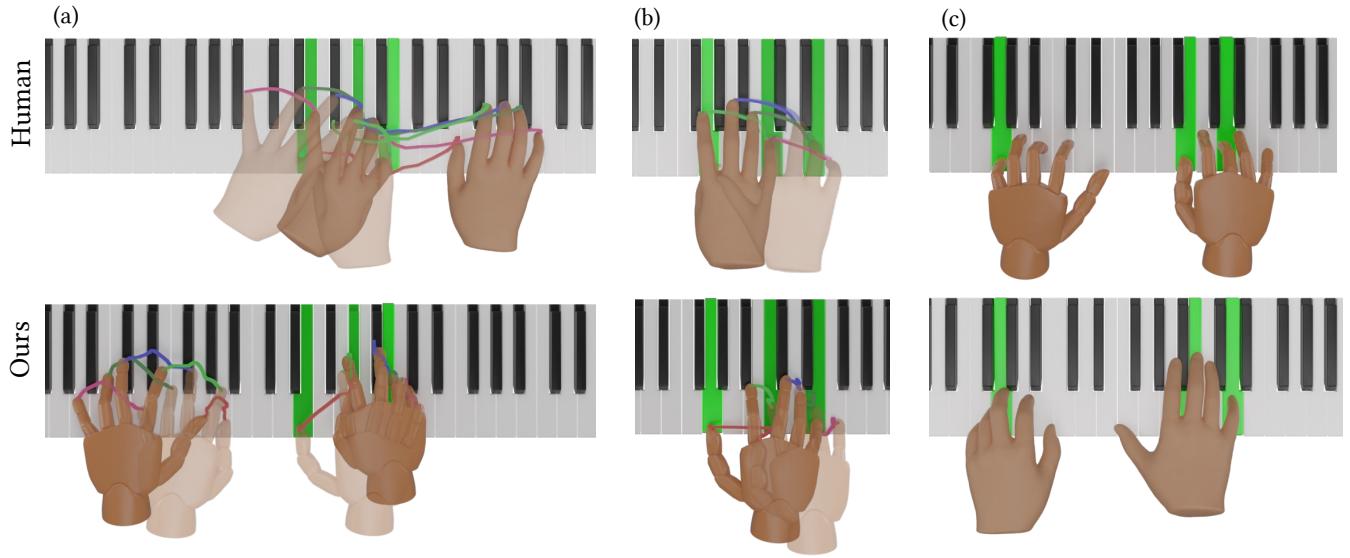


Fig. S1. Comparison between the synthesized motions and motions in our dataset when facing the same target notes. Our control policy could take diverse key-pressing poses by leveraging motions synthesized by the diffusion models (a), or imitating poses existing in our dataset with the similar (b) or different (c) fingering strategy.

as shown in Figure S1a. Figure S1b shows an example where the synthesized motion largely resembles human motion in terms of fingering and hand poses. Finally, we show an example where the control policy yields results with similar hand poses but different fingering compared with the human pianist.

#### D REPERTOIRE

*List of compositions in the dataset.* We include the list of all the compositions in our dataset in Table S2.

Table S2. List of compositions in FürElise.

Piece	Composer
Excerpt of Nocturne for the Left Hand, Op. 9, No. 2	Alexander Scriabin
In meines Vaters Garten	Alma Mahler
Laue Sommernacht	Alma Mahler
Bei dir ist es traut	Alma Mahler
Die stille Stadt	Alma Mahler
Concerto No. 5 in F Major, Op. 103, "Egyptian": I. Allegro animato	Camille Saint-Saëns
Concerto No. 5 in F Major, Op. 103, "Egyptian": III. Molto Allegro	Camille Saint-Saëns
Concerto No. 5 in F Major, Op. 103, "Egyptian": II. Andante	Camille Saint-Saëns
Clair de Lune from Suite Bergamasque	Claude Debussy
Trois Chanson de Bilitis	Claude Debussy
Images, Book I: III. Mouvement	Claude Debussy
Arabesque No. 1 in E Major	Claude Debussy
Images, Book II: I. Cloches à travers les feuilles	Debussy
Images, Book II: III. Poissons d'or	Debussy
Images, Book II: II. Et la lune descend sur le temple qui fut	Debussy
Lyric Pieces, Op. 71: No. 2 Sommerabend	Edvard Grieg
Lyric Pieces, Op. 43: No. 1 Schmetterling	Edvard Grieg
Lyric Pieces, Op. 62: No. 6 Heimwärts	Edvard Grieg
Lyric Pieces, Op. 62: No. 4 Bächlein	Edvard Grieg
Lyric Pieces, Op. 54: No. 3 Zug der Zwerge	Edvard Grieg
Lyric Pieces, Op. 38: No. 1 Berceuse	Edvard Grieg
Piano Quintet No. 1 in C Minor, Op. 1: II. Scherzo (Allegro vivace)	Ernő Dohnányi
Piano Quintet No. 1 in C Minor, Op. 1: III. Adagio, quasi andante	Ernő Dohnányi
Piano Quintet No. 1 in C Minor, Op. 1 IV. Finale: Allegro moderato	Ernő Dohnányi
Piano Quintet No. 1 in C Minor, Op. 1: I. Allegro	Ernő Dohnányi
Excerpt of Prelude No. 7	Federico Mompou
Prelude No. 8 "On a Drop of Water"	Federico Mompou
Clouds	Florence Price
Impromptu in G-Flat Major, Op. 90, No. 3, D. 899	Franz Schubert
Mazurka in A Minor, Op. 59, No. 1	Frédéric Chopin
Prelude in E Minor, Op. 28, No. 4, "Largo"	Frédéric Chopin
Prelude in D Major, Op. 28, No. 5, "Allegro molto"	Frédéric Chopin
Prelude in B Minor, Op. 28, No. 6, "Lento assai"	Frédéric Chopin
Prelude in A Major, Op. 28, No. 7, "Andantino"	Frédéric Chopin
Prelude in F-Sharp Minor, Op. 28, No. 8, "Molto agitato"	Frédéric Chopin
Prelude in E Major, Op. 28, No. 9, "Largo"	Frédéric Chopin
Prelude in C-Sharp Minor, Op. 28, No. 10, "Allegro molto"	Frédéric Chopin
Étude in A-Flat Major, Op. 25, No. 1	Frédéric Chopin
Étude in F Major, Op. 10, No. 8	Frédéric Chopin
Prelude in G Major, Op. 28, No. 3, "Vivace"	Frédéric Chopin
Ballade No. 4 in F Minor, Op. 52	Frédéric Chopin
Prelude in B Major, Op. 28, No. 11, "Vivace"	Frédéric Chopin
Prelude in G-Sharp Minor, Op. 28, No. 12, "Presto"	Frédéric Chopin
Concerto No. 1 in E Minor, Op. 11: II. Romance - Larghetto	Frédéric Chopin
Concerto No. 1 in E Minor, Op. 11: I. Allegro maestoso	Frédéric Chopin
Grande Polonaise Brillante	Frédéric Chopin
Andante Spianato	Frédéric Chopin
Concerto No. 1 in E Minor, Op. 11: III. Rondo - Vivace	Frédéric Chopin
Étude in G-Sharp Minor, Op. 25, No. 6	Frédéric Chopin
Prelude in A Minor, Op. 28, No. 2, "Lento"	Frédéric Chopin
Étude in E Major, Op. 10, No. 3	Frédéric Chopin
Piano Sonata No. 2 in B-Flat Minor, Op. 35: I. Grave – Doppio movimento	Frédéric Chopin

Ballade No. 3 in A-Flat Major, Op. 47	Frédéric Chopin
Piano Sonata No. 2 in B-Flat Minor, Op. 35: IV. Finale: Presto	Frédéric Chopin
Piano Sonata No. 2 in B-Flat Minor, Op. 35: III. Marche funèbre: Lento	Frédéric Chopin
Waltz in A-Flat Major, Op. 34, No. 1	Frédéric Chopin
Prelude in D-Flat Major, Op. 28, No. 15, "Raindrop"	Frédéric Chopin
Waltz in D-Flat Major, Op. 64, No. 1, "Minute Waltz"	Frédéric Chopin
Piano Sonata No. 2 in B-Flat Minor, Op. 35: II. Scherzo	Frédéric Chopin
Prelude in C Major, Op. 28, No. 1, "Agitato"	Frédéric Chopin
Sonata No. 3 in B Minor, Op. 58: IV. Finale: Presto, non tanto	Frédéric Chopin
Nocturne in D-flat major, Op. 27, No. 2	Frédéric Chopin
Fantaisie-Impromptu in C-Sharp Minor, Op. 66	Frédéric Chopin
Air "The Harmonious Blacksmith"	George Frideric Handel
Suite No. 5 in E Major: III. Courante	George Frideric Handel
Suite No. 5 in E Major: II. Allemande	George Frideric Handel
Suite No. 5 in E Major: I. Prelude	George Frideric Handel
Rhapsody in Blue	George Gershwin
Preludes, Nos. 1, 2, 3	George Gershwin
Ich atmet einen Lindenduft	Gustav Mahler
Twenty-six Etudes (2007) Part II: No. 10 Andantino Cantabile	H. Leslie Adams
Concerto in D Minor, BWV 1052: I. Allegro	J.S. Bach
Prelude in E Major, BWV 854	J.S. Bach
Goldberg Variations, BWV 988: I. Aria	J.S. Bach
English Suite No. 3 in g minor, BWV 808: I. Prelude	J.S. Bach
English Suite No. 3 in g minor, BWV 808: II. Allemande	J.S. Bach
English Suite No. 3 in g minor, BWV 808: III. Courante	J.S. Bach
English Suite No. 3 in g minor, BWV 808: IV. Sarabande	J.S. Bach
English Suite No. 3 in g minor, BWV 808: V. Gavotte I	J.S. Bach
English Suite No. 3 in g minor, BWV 808: VI. Gavotte II (ou la musette)	J.S. Bach
English Suite No. 3 in g minor, BWV 808: VII. Gigue	J.S. Bach
Prelude in C Minor, BWV 847	J.S. Bach
Italian Concerto, BWV 971: I. Allegro	J.S. Bach
Fugue in D Minor, BWV 875	J.S. Bach
Prelude in D Minor, BWV 875	J.S. Bach
Concerto in D Minor, BWV 1052: III. Allegro	J.S. Bach
English Suite no 6 in D minor, BWV 811, Gavotte	J.S. Bach
Preludes and Fugue in F Minor, BWV 881	J.S. Bach
Concerto in D Minor, BWV 974: II. Adagio	J.S. Bach
Prelude in C Sharp Major, BWV 872	J.S. Bach
Summer Hue	Jennifer Higdon
Intermezzo in A Major, Op. 118, No. 2: Andante teneramente	Johannes Brahms
Intermezzo in A Minor, Op. 116, No. 2: Andante	Johannes Brahms
Theme and Var 1-6 from Johannes Brahms Variations	Johannes Brahms
Intermezzo in A Minor, Op. 116, No. 2: Andante	Johannes Brahms
Andantino Cantabile	Leslie Adams
Trois morceaux pour piano: D'un vieux jardin	Lili Boulanger
Trois morceaux pour piano: D'un jardin clair	Lili Boulanger
Trois morceaux pour piano: Cortège	Lili Boulanger
Sonata in C Major, Op. 2, No. 3: I. Allegro con brio	Ludwig van Beethoven
Piano Trio No. 7 in B-Flat Major, Op. 11, "Gassenhauer": I. Allegro con brio	Ludwig van Beethoven
Piano Sonata in B-flat Major, Op. 22: I. Allegro con brio	Ludwig van Beethoven
Piano Sonata No. 32 in C Minor, Op. 111: I. Maestoso – Allegro con brio ed appassionato	Ludwig van Beethoven
Concerto No. 5 in E-Flat Major, Op. 73: I. Allegro (excerpt)	Ludwig van Beethoven
Piano Sonata No. 32 in C Minor, Op. 111: II. Arietta: Adagio molto semplice e cantabile	Ludwig van Beethoven
Troubled Waters	Margaret Bonds
Miroirs: III. Une Barque sur l'Océan	Maurice Ravel
Not Everyone Thinks That I'm Beautiful	Michael Tilson Thomas

Grace	Michael Tilson Thomas
All Blues	Miles Davis
Pictures at an Exhibition, Mvt. 1: Promenade	Modest Mussorgsky
Pictures at an Exhibition, Mvt. 10: The Great Gate of Kiev	Modest Mussorgsky
What is this thing called love	Others
Bewitched Bothered and Bewildered	Others
Slow Boat to China	Others
Scales	Others
Czerny No. 1-3 from the School of Velocity	Others
Hanon No. 21 & 22 from The Virtuoso Pianist Pt II	Others
Scales, Arpeggios and Chords	Others
Scales	Others
Scales in 2nds, other exercises	Others
Fantasie in C major, Op. 17: I. Durchaus phantastisch und leidenschaftlich vorzutragen	Robert Schumann
Fantasie in C major, Op. 17: III. Langsam getragen. Durchweg leise zu halten	Robert Schumann
Piano Sonata No. 1 in F-sharp minor, Op. 11: I. Introduzione. Un poco adagio – Allegro vivace	Robert Schumann
Piano Sonata No. 1 in F-sharp minor, Op. 11: II. Aria	Robert Schumann
Fantasie in C major, Op. 17: II. Mäßig. Durchaus energisch	Robert Schumann
Kinderszenen, Op. 15 No.10: Fast zu ernst	Robert Schumann
Kinderszenen, Op. 15 No.5: Glückes genug	Robert Schumann
Novellette No. 1 in F Major, Op. 21	Robert Schumann
Kinderszenen, Op. 15 No.1: Von fremden Ländern und Menschen	Robert Schumann
Kinderszenen, Op. 15 No.2: Kurose Geschichte	Robert Schumann
Kinderszenen, Op. 15 No.3: Hasche-Mann	Robert Schumann
Kinderszenen, Op. 15 No.4: Bittendes Kind	Robert Schumann
Kinderszenen, Op. 15 No.6: Wichtige Begebenheit	Robert Schumann
Kinderszenen, Op. 15 No.7: Träumerei	Robert Schumann
Kinderszenen, Op. 15 No.8: Am Kamin	Robert Schumann
Piano Sonata No. 1 in F-sharp minor, Op. 11: IV. Finale. Allegro un poco maestoso	Robert Schumann
Piano Sonata No. 1 in F-sharp minor, Op. 11: III. Scherzo e Intermezzo. AllegriSSimo	Robert Schumann
Kinderszenen, Op. 15 No.11: Fürchtenmachen	Robert Schumann
Kinderszenen, Op. 15 No.12: Kind im Einschlummern	Robert Schumann
Kinderszenen, Op. 15 No.13: Der Dichter spricht	Robert Schumann
Kinderszenen, Op. 15 No.9: Ritter vom Steckenpferd	Robert Schumann
Expanse of my Soul	Scott Ordway
Études-Tableaux, Op. 39: IV. Allegro assai in b minor	Sergei Rachmaninoff
Études-Tableaux, Op. 39: VI. Allegro in a minor	Sergei Rachmaninoff
Études-Tableaux, Op. 39: VII. Lento lugubre in c minor	Sergei Rachmaninoff
Barcarolle in G minor, Op. 10, No. 3	Sergei Rachmaninoff
Going up Yonder Improvisation	Stephen Prutsman
Black Pearl	Stephen Prutsman
Chopin Freddie	Stephen Prutsman
Sonata in B-Flat Major, K. 333: III. Allegretto grazioso	Wolfgang Amadeus Mozart
Ah, vous dirai-je, maman Variations, K. 265	Wolfgang Amadeus Mozart
Sonata in B-Flat Major, K. 333: I. Allegro	Wolfgang Amadeus Mozart