DexDrummer: In-Hand, Contact-Rich, and Long-Horizon Dexterous Robot Drumming

Anonymous

***Abstract*— Performing in-hand, contact-rich, and long-horizon dexterous manipulation remains an unsolved challenge in robotics. Prior hand dexterity works have considered each of these three challenges in isolation, yet do not combine these skills into a single, complex task. To further test the capabilities of dexterity, we propose drumming as a testbed for dexterous manipulation. Drumming naturally integrates all three challenges: it involves in-hand control for stabilizing and adjusting the drumstick with the fingers, contact-rich interaction through repeated striking of the drum surface, and long-horizon coordination when switching between drums and sustaining rhythmic play. We present DexDrummer, a hierarchical object-centric bimanual drumming policy trained in simulation with sim-to-real transfer. The framework reduces the exploration difficulty of pure reinforcement learning by combining trajectory planning with residual RL corrections for fast transitions between drums. A dexterous manipulation policy handles contact-rich dynamics, guided by rewards that explicitly model both finger–stick and stick–drum interactions. In simulation, we show our policy can play two styles of music: multi-drum, bimanual songs and challenging, technical exercises that require increased dexterity. Across simulated bimanual tasks, our dexterous, reactive policy outperforms a fixed grasp policy by 1.87x across easy songs and 1.22x across hard songs F1 scores. In real-world tasks, we show song performance across a multi-drum setup. DexDrummer is able to play our training song and its extended version with an F1 score of 1.0.**

I. INTRODUCTION

Dexterous hand manipulation is an attractive problem in robotics because it unlocks lots of real-world tasks. Existing works have tackled challenges such as in-hand object reorientation [1], [2], grasping [3], [4], [5], [6], [7], [8], [9], and tool-based manipulation [10], all of which require complex finger–object interactions. While these works provide useful insights on dexterity, these studies typically focus on short-horizon tasks or narrow aspects of dexterity in isolation.

In contrast, many real-world tasks such as assembly or cooking require dexterous skills that combine in-hand control, robustness to external perturbations, and long-horizon robustness. For example, assembling parts often involves reorienting a fastener in the hand while applying force to connect components, and cooking requires both holding utensils stably and stirring against resistance.

Motivated by the need for a compelling testbed, we propose drumming, a long-horizon, contact-rich dexterous manipulation task. Drumming inherently requires balancing in-hand control – maintaining and adjusting the grasp of the stick with fine finger control – and external contact – forcefully and repeatedly striking drums. To play long songs, this control becomes even more crucial: drumming requires a policy robust to these contacts for extended periods of time.

To address the challenges of in-hand control, external forces, and long-horizon robustness, we propose DexDrummer, a hierarchical dexterous drumming framework trained in simulation with sim-to-real transfer. As illustrated in Fig. 1, our system decomposes the task into a high-level policy and a low-level dexterous policy.

At the high level, we reduce the exploration difficulty of pure RL by introducing parameterized motion primitives that generate task-space drumstick trajectories from musical inputs. These trajectories are converted into arm motions via motion planning, producing nominal control commands for the robot arms. A residual RL policy then learns corrective adjustments on top of this planner to compensate for tracking errors during fast transitions between drums.

At the low level, we train a dexterous manipulation policy to handle the contact-rich dynamics of drumming. Our key insight is to structure learning using contact-targeted rewards, which address two types of interactions: in-hand contacts and external contacts. In-hand contacts correspond to finger–stick interactions that manipulate the drumstick through fingertip contact and a fulcrum grasp and stabilize through arm energy penalty. External contacts correspond to interactions between the stick and the drum surface. To learn robust striking behavior, we introduce trajectory-guided rewards and a contact curriculum that stabilizes learning of impacts.

In simulation, we show DexDrummer performance on two styles of music: bimanual, multi-drum songs, and a uni-drum technical exercise. We show sim-to-real transfer for real-world drumming, showcasing DexDrummer's capability to perform dexterous, contact-rich control of the drum stick across a long horizon.

In summary, our main contributions are threefold:

1) We introduce drumming as a challenging testbed for dexterous manipulation, unifying the challenges of in-hand manipulation, external contacts, and long-horizon robustness.
2) We propose DexDrummer, a hierarchical, two-stage policy for drumming. DexDrummer consists of a high-level residual RL policy and a low-level dexterous policy that optimized contact-targeted rewards, to address the challenges of in-hand and external contacts.
3) We showcase simulated drumming across six different musical genres at two difficulty levels, with performances ranging from 20 to 40 seconds, as well as a high-speed exercise piece designed to test finger dexterity. We further evaluate our method on a real-world drumming task using a cymbal and snare setup.

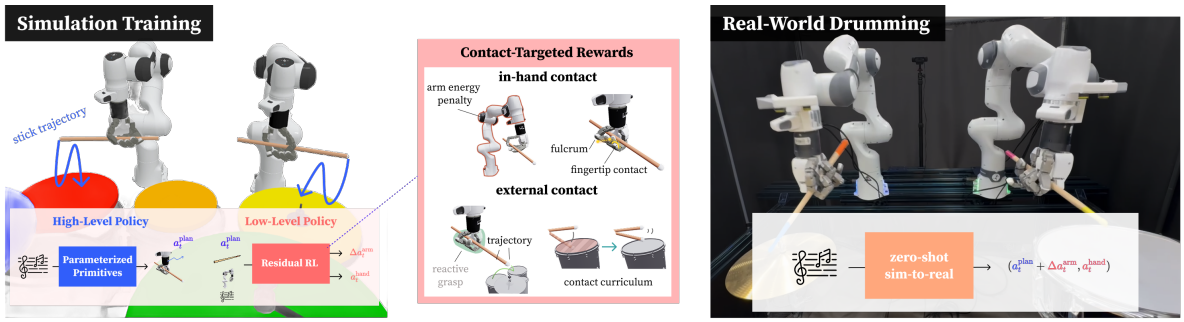

Fig. 1. **DexDrummer**: A hierarchical dexterous drumming framework trained in simulation and deployed in the real world. **Left: Training in simulation.** Our policy decomposes the task into a hierarchical controller. A *high-level policy* generates parameterized motion primitives that produce drumstick trajectories from musical inputs. A *low-level policy* then uses residual RL to learn corrective arm and hand actions for accurate trajectory tracking during fast transitions. **Middle: Contact-targeted rewards.** To handle contact-rich dynamics, we design rewards that target two types of interactions. *In-hand contacts* encourage stable finger–stick manipulation through fingertip contact rewards, a fulcrum reward between the thumb and index finger, and an arm energy penalty that promotes finger-dominant control. *External contacts* address stick–drum interactions with a trajectory guidance reward to encourage drum strikes and a contact curriculum that gradually introduces drum contact during training. **Right: Real-world drumming.** Policies trained in simulation transfer zero-shot to the real robot, enabling dexterous multi-drum playing over long horizons.

II. Setup

A. Problem Statement

We would like to learn a robot drumming policy via reinforcement learning. Under this framework, our objective is to learn a policy $\pi(a \mid s)$ that maximizes the expected discounted cumulative reward across trajectories. We assume an environment with observation o_t, r_t per timestep t, and discount factor γ. We detail the policy inputs in table II and reward functions in table I.

B. Drum Environment

We create a simulated drum environment in the ManiSkill framework [11] that consists of a bimanual robot setup and a full drum set (snare, tom, ride, hi-hat, and crash). In particular, this requires us to control and coordinate two arms and hands under a single policy, that can simultaneously play different drums. We design three types of tasks for evaluation, as shown in Fig. 5: (1) bimanual full–drum set songs in simulation, (2) single-drum tasks that emphasize dexterity in both simulation and the real world, and (3) bimanual two-drum songs in the real world.

III. System Overview

We present DexDrummer, a bimanual dexterous manipulation policy capable of playing drums in both simulation and the real world. An overview of the framework is shown in Figure 1. Our method adopts a hierarchical, object-centric architecture inspired by prior work [12], [13], consisting of a high-level policy (Sec. III-A) and a low-level dexterous policy (Sec. III-B). To address the complex contact dynamics inherent in drumming, we introduce contact-targeted rewards that handle both in-hand object interactions and external drumstick–surface contacts.

A. High-Level Policy

The first challenge in drumming is coordinating large arm movements to navigate between drums while maintaining controllable stick motion. We formulate this as an object-centric task, where the drumstick trajectory serves as the primary reference for planning. Given the target drum sequence, we first generate the desired stick trajectory in task space. This trajectory is then transformed into corresponding end-effector motions via relative pose offsets, enabling arm-level motion planning.

However, purely kinematic planning can fail when the required motions are highly dynamic or involve rapid transitions between drums. In such cases, tracking errors accumulate and the stick may deviate from the desired trajectory. To improve robustness, we introduce a residual reinforcement learning (RL) policy that performs small corrective adjustments on top of the nominal planner. This residual formulation significantly reduces the effective action space while enabling fine-grained trajectory correction for precise stick tracking.

B. Low-Level Dexterous Policy

The second challenge lies in achieving highly dexterous finger control for stable in-hand manipulation under dynamic impacts. We design the reward function with three components: (1) *In-Hand Contact Rewards*, (2) *External Contact Rewards*, and (3) *Task Rewards*. Further details on the design decisions are provided in Appendix V-B.

In-Hand Contact Rewards. We encourage stable finger–stick interaction via fingertip contact rewards [12] and a fulcrum reward that biases grasping toward the stick center following human drumming priors. We further include an energy regularization term to promote efficient, fine-grained finger control.

External Contact Rewards. We decompose external interaction into (1) contact initiation and (2) contact handling. To guide initiation, we apply a trajectory reward based on pre-computed stick motions. We also introduce a contact curriculum that first disables stick–drum contact to learn trajectory following, then reintroduces contact to enable stable hitting. This improves robustness under contact dynamics.

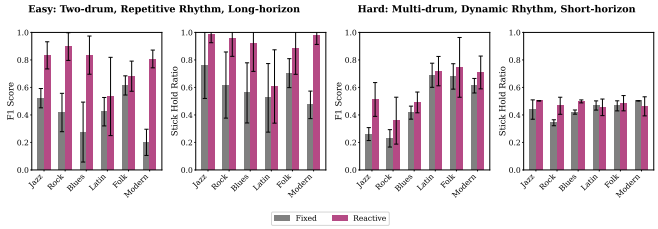

Fig. 2. **Results for Dexterous Song-Playing.** ***Left***: Reactive grasp outperforms fixed grasp by a large margin in long-horizon contacts. ***Right***: For more challenging songs requiring frequent drum-to-drum transitions, reactive grasp still improves performance, but with a smaller margin, primarily due to the reduced action space of fixed grasp.

Task Rewards. We use a sparse hit reward to evaluate whether the drum is struck at the desired time [14].

IV. EXPERIMENTS

We seek to answer the following questions:

1) Is dexterity *essential* for achieving robust long-horizon drumming? Can we learn such a dexterous policy?
2) How can we enable finger-driven control for precise drumming?
3) How do our design decisions affect dexterous control?
4) Can the learned behaviors transfer to real world?

A. Experimental Setup

1) Hardware Setup: We use two 7-DOF Franka Panda arms and two 20-DOF Tesollo DG-5F hands in both simulated bimanual and real-world unimanual tasks. Policy inference runs at 20 Hz with a PID joint position controller at 100 Hz.

We evaluate real-world drumming on a unimanual setup with a drum pad and cymbal, and construct a digital twin with randomized parameters for sim-to-real transfer. Stick proprioception is obtained via color segmentation and depth sensing using a RealSense camera, projected into the robot frame using calibrated intrinsics and extrinsics. Additional details are provided in Appendix V-C.

2) Evaluation Metrics: We evaluate bimanual performance using (1) F1 score for song execution and (2) stick-hold ratio. For dexterous control, we report (1) trajectory error and (2) energy consumption. See Appendix V-C for details.

3) Drum Songs: We use MIDI files to specify drum sequences following prior work [15], [14]. We construct **Easy** and **Hard** tracks across multiple genres to evaluate long-horizon stability and coordination complexity, respectively. We also include a Finger-Driven Control task to assess high-speed fine-grained dexterity. More details are provided in Appendix V-C.

B. Dexterity for Bimanual Song Playing

In our first set of experiments, we evaluate the importance of dexterity for long-horizon tool manipulation. For drumming, grasping the drum stick and repeatedly hitting a drum for extended periods of time likely necessitates reorientation and adjustment of the drum stick. To test the effectiveness of our dexterous policy, we compare *Fixed Grasp*, where the finger joints are frozen after reaching an initial grasp of the stick, to our method, *Reactive Grasp*, where the agent exerts dexterous control of the stick.

We evaluate the F1 Score and Hold Duration of the stick. The F1 Score represents how well the policy can play the song, showing that DexDrummer can learn a successful dexterous drumming policy. The Hold Duration evaluates how *Fixed Grasp* and *Reactive Grasp* are affected by slippage as the robot continues to move and hit the drum stick.

In fig. 2, we present two setups: an easier scenario with a repetitive loop but long horizon (*left*), and a more challenging scenario with multiple drums, dynamic rhythm, but shorter horizon (*right*). In the left case, the reactive grasp clearly outperforms the fixed grasp in both song performance and stick hold duration, highlighting the necessity of reactive, closed-loop dexterous control for long-horizon contact. In the right case, the reactive grasp still outperforms the fixed grasp, but with a much smaller margin. This is mainly because the fixed grasp only needs to learn arm motion within a lower-dimensional action space, which makes it easier to handle complex drum-to-drum transitions. The result shows a trade-off between dexterity and learning complexity, making it an interesting direction for balancing the two.

C. Finger-Driven Control

Next, we explore how to enable fine-grained, dexterous motions instead of relying on unnatural, whole-arm movements. Drumming can be attempted through arm movements or dexterous finger movements, but the precise finger control can often be overshadowed by initial arm exploration and movement. In this experiment, we evaluate how DexDrummer guides dexterity, and whether this leads to better performance of the song.

To assess dexterity, we design an exercise requiring rapid hits on a single drum, which is difficult for arm-driven motion and encourages finger control. We incorporate an arm energy penalty and contact curriculum (see table I) to promote finger-driven behavior. The arm penalty discourages arm movement, while the contact curriculum helps the policy handle external contact. The arm-driven baseline uses the same rewards, except for these two components.

For this task, we evaluate the F1 Score, Trajectory Error, and Energy Consumption across a range of speeds for the exercise, denoted by Beats Per Minute (BPM). A higher BPM implies quicker hits and less time in between hits, and is thus more challenging.

As shown in fig. 3, finger-driven control outperforms arm-driven control, particularly as the BPM increases. This highlights the necessity of fingers for fine-grained, high-speed motions, as arm-driven motion is unable to accurately replicate the quick stick movements. Moreover, finger-driven motion results in significantly lower energy consumption, which is advantageous for practical deployment. Qualitatively, in fig. 7, we find finger-driven motions to look more

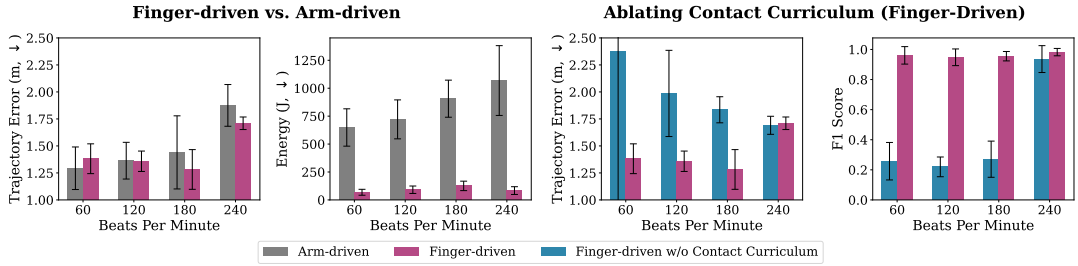

Fig. 3. **Results for Finger-Driven Control.** ***Left:*** As tempo (beats per minute) increases, trajectory error for finger-driven control decreases and the gap to arm-driven control widens, showing the superior dexterity of finger-driven control. Energy consumption is also substantially lower compared to arm-driven control. ***Right:*** Without contact-targeted rewards, finger-driven control struggles to manage contact interactions effectively.

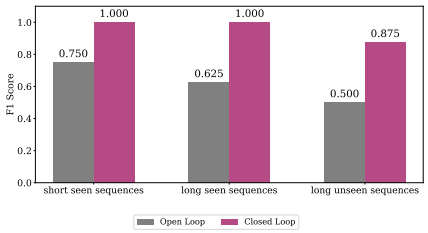

Fig. 4. **Results for Real-world Unimanual Drum Playing.** We denote the cymbal as c and the drum pad as d. The short seen sequence used for training is $ccdd$. The long seen sequence is $ccccdddd$, while the long unseen sequence is $ccddccdd$. Open-loop replay achieves reasonable performance but cannot adapt to real-world dynamics. Our policy is able to play unseen sequences (e.g., $d \rightarrow c$) by conditioning on the desired trajectory.

natural and human-like, whereas arm-driven motions are clunky and dangerous.

The fig. 3 (right) investigates what enables this dexterous finger movement. We ablate our finger-driven motion policy by removing the contact curriculum. Across different BPMs, removing the contact curriculum leads to much higher trajectory errors, implying that the curriculum is necessary to learn finger dexterity for following the stick trajectory. An exception occurs at a very high tempo (240 BPM), where the robot only needs to lift the stick slightly above the drum before hitting it again. Here, the effect of the contact curriculum is limited because the dexterous motion is small and easier to learn. This is further exemplified through the F1 scores, which shows a drastic gap when the contact curriculum is not enabled, besides for the fastest tempo (240 BPM). Qualitatively, in fig. 3, without the contact curriculum, the stick often rests on the drum head, preventing effective finger exploration. We hypothesize that because finger movements are relatively small, they can easily be canceled out by external contact.

D. Real-World Drumming

Open-Loop vs. Closed-Loop Drumming for Robustness. Finally, we show results on real-world drumming with a drum pad and cymbal. We evaluate the effectiveness of our sim-to-real policy across multiple songs in an open and closed-loop setting, similar to [16]. First, we evaluate F1 performance on the seen song our RL agent is trained on. Then, we evaluate on an extended version of the song, which should have hits and trajectories that are seen in the train song. Lastly, our most difficult song not only requires more hits than the train song, but it also includes unseen drum transitions. This is to evaluate the generalization of our policy to out-of-distribution songs.

In fig. 4, we find that our closed-loop policy consistently outperforms the open-loop policy. This shows that closed-loop control is essential for real-world drumming, as the robot is able to react to stick movements to better play the song. Our closed-loop policy is able to play both the train song and an extended version of the train song with an F1 score of 1.0, showing that sim-to-real can enable effective drum playing. Additionally, the closed-loop policy can play songs with unseen drum transitions due to high-level trajectory guidance that our policy is conditioned on.

In Fig. 8, we show a sample rollout from real-world drum playing. Notably, the dynamics of hitting a cymbal and a drum pad differ: the cymbal is not fixed and involves less contact, whereas the drum pad is fixed and introduces more substantial contact. When striking the cymbal, the grip remains relatively loose since the interaction does not cause significant instability. In contrast, after striking the drum pad, the grip becomes firmer to handle the stronger contact. This adaptation shows the effectiveness of the reactive grasp.

Dexterous Control on Exercise Songs. We also showcase the experiments from IV-C in the real world. As shown in the supplemental video, DexDrummer is able to execute highly dexterous actions to play at high speeds. Qualitatively, we evaluate the behavior under different speeds. At 120 BPM, the last three fingers are used to drag the stick downward for the upstroke. At 240 BPM, the last three fingers instead press the stick to generate a faster downstroke. This behavior resembles the technique used by human drummers.

V. CONCLUSION

We introduced DexDrummer, a testbed for dexterous manipulation that unifies the challenges of in-hand control, contact-rich interaction, and long-horizon tasks. To address these coupled challenges, we propose a hierarchical framework in which a high-level policy leverages parameterized motion primitives to reduce RL exploration, while a low-level policy handles both in-hand and external contacts under dexterous control. We demonstrated the effectiveness of our framework through bimanual song playing and fine-grained control on the real robot. In future work, we aim to extend these insights to broader problems that involve interaction with in-hand objects and the physical world.

REFERENCES

[1] OpenAI *et al.*, "Solving rubik's cube with a robot hand," 2019. [Online]. Available: https://arxiv.org/abs/1910.07113

[2] J. Wang *et al.*, "Lessons from learning to spin "pens"," in *CoRL*, 2024.

[3] P. Li *et al.*, "Gendexgrasp: Generalizable dexterous grasping," *arXiv preprint arXiv:2210.00722*, 2022.

[4] Y. Zhong *et al.*, "Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness," *arXiv preprint arXiv:2503.08257*, 2025.

[5] A. T. Miller *et al.*, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[6] R. Wang *et al.*, "Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," *arXiv preprint arXiv:2210.02697*, 2022.

[7] S. Brahmbhatt *et al.*, "Contactgrasp: Functional multi-finger grasp synthesis from contact," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2386–2393.

[8] Y. Xu *et al.*, "Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4737–4746.

[9] Y. Zhong *et al.*, "Dexgraspvla: A vision-language-action framework towards general dexterous grasping," 2025. [Online]. Available: https://arxiv.org/abs/2502.20900

[10] K. Shaw *et al.*, "Bimanual dexterity for complex tasks," in *8th Annual Conference on Robot Learning*, 2024.

[11] S. Tao *et al.*, "Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai," *Robotics: Science and Systems*, 2025.

[12] V. de Bakker *et al.*, "Scaffolding dexterous manipulation with vision-language models," 2025. [Online]. Available: https://arxiv.org/abs/2506.19212

[13] Y. Chen *et al.*, "Object-centric dexterous manipulation from human motion data," in *8th Annual Conference on Robot Learning*.

[14] A. A. Shahid *et al.*, "Robot drummer: Learning rhythmic skills for humanoid drumming," *arXiv preprint arXiv:2507.11498*, 2025.

[15] K. Zakka *et al.*, "Robopianist: Dexterous piano playing with deep reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2023.

[16] "Learning to play piano in the real world," 2025.

[17] A. Bhatt* *et al.*, "Surprisingly robust in-hand manipulation: An empirical study," in *Robotics: Science and Systems XVII*, ser. RSS2021. Robotics: Science and Systems Foundation, Jul. 2021. [Online]. Available: http://dx.doi.org/10.15607/RSS.2021.XVII.089

[18] Y. Bai *et al.*, "Dexterous manipulation using both palm and fingers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1560–1565.

[19] J. Yin *et al.*, "Learning in-hand translation using tactile skin with shear and normal force sensing," 2025. [Online]. Available: https://arxiv.org/abs/2407.07885

[20] Z.-H. Yin *et al.*, "Rotating without seeing: Towards in-hand dexterity through touch," 2023. [Online]. Available: https://arxiv.org/abs/2303.10880

[21] H. Qi *et al.*, "General in-hand object rotation with vision and touch," 2023. [Online]. Available: https://arxiv.org/abs/2309.09979

[22] T. G. W. Lum *et al.*, "Get a grip: Multi-finger grasp evaluation at scale enables robust sim-to-real transfer," 2024. [Online]. Available: https://arxiv.org/abs/2410.23701

[23] H.-S. Fang *et al.*, "Dexop: A device for robotic transfer of dexterous human manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2509.04441

[24] C. Wang *et al.*, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2403.07788

[25] L. Han *et al.*, "Dextrous manipulation by rolling and finger gaiting," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 1. IEEE, 1998, pp. 730–735.

[26] Y. Bai *et al.*, "Dexterous manipulation using both palm and fingers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1560–1565.

[27] I. Mordatch *et al.*, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 2012, pp. 137–144.

[28] A. Xie *et al.*, "Handelbot: Real-world piano playing via fast adaptation of dexterous robot policies," 2026. [Online]. Available: https://arxiv.org/abs/2603.12243

[29] M. Liu *et al.*, "Deep differentiable grasp planner for high-dof grippers," *arXiv preprint arXiv:2002.01530*, 2020.

[30] T. Liu *et al.*, "Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 470–477, 2021.

[31] N. C. Dafle *et al.*, "Extrinsic dexterity: In-hand manipulation with external forces," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1578–1585.

[32] H. Qi *et al.*, "In-hand object rotation via rapid motor adaptation," 2022. [Online]. Available: https://arxiv.org/abs/2210.04887

[33] M. Yang *et al.*, "Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch," 2024. [Online]. Available: https://arxiv.org/abs/2405.07391

[34] E. Corona *et al.*, "Ganhand: Predicting human grasp affordances in multi-object scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5031–5041.

[35] C. Rosales *et al.*, "Synthesizing grasp configurations with specified contact regions," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 431–443, 2011.

[36] S. P. Arunachalam *et al.*, "Holo-dex: Teaching dexterity with immersive mixed reality," 2022. [Online]. Available: https://arxiv.org/abs/2210.06463

[37] Y. Qin *et al.*, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," in *Robotics: Science and Systems*, 2023.

[38] T. Lin *et al.*, "Learning visuotactile skills with two multifingered hands," 2024. [Online]. Available: https://arxiv.org/abs/2404.16823

[39] T. Tao *et al.*, "Dexwild: Dexterous human interactions for in-the-wild robot policies," *Robotics: Science and Systems (RSS)*, 2025.

[40] M. Xu *et al.*, "Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2505.21864

[41] A. Agarwal *et al.*, "Dexterous functional grasping," in *7th Annual Conference on Robot Learning*, 2023.

[42] K. Shaw *et al.*, "Videodex: Learning dexterity from internet videos," *CoRL*, 2022.

[43] T. G. W. Lum *et al.*, "Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration," 2025. [Online]. Available: https://arxiv.org/abs/2504.12609

[44] T. Lin *et al.*, "Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids," *arXiv:2502.20396*, 2025.

[45] A. Kapur *et al.*, "A comparison of solenoid-based strategies for robotic drumming," 01 2007.

[46] S. Jordà, "Afasia: The ultimate homeric one-man-multimedia-band." 01 2002, pp. 132–137.

[47] Y. G. Kim *et al.*, "Drum stroke variation using variable stiffness actuators," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3892–3897.

[48] E. Berdahl *et al.*, "A physically intuitive haptic drumstick," in *ICMC*, 2007.

[49] S. M. Karbasi *et al.*, "Embodied intelligence for drumming: A reinforcement learning approach to drumming robots," in *Front Robot AI*, 2024.

[50] J. Tan *et al.*, "Sim-to-real: Learning agile locomotion for quadruped robots," 2018. [Online]. Available: https://arxiv.org/abs/1804.10332

[51] X. Song *et al.*, "Rapidly adaptable legged robots via evolutionary meta-learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3769–3776.

[52] Z. Fu *et al.*, "Minimizing energy consumption leads to the emergence of gaits in legged robots," in *Conference on Robot Learning (CoRL)*, 2021.

[53] C. Qian *et al.*, "Pianomime: Learning a generalist, dexterous piano player from internet demonstrations," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=t0LkF9JnVb

[54] Z. Mandi *et al.*, "Dexmachina: Functional retargeting for bimanual dexterous manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2505.24853

[55] J. Schulman *et al.*, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

Fig. 5. **Drumming Environments.** Our first simulation environment includes bimanual, multi-drum song-playing. Our second environment involves unimanual, uni-drum control for a high-speed technical exercise. Finally, in the real-world, we play songs with a drum pad and cymbal.

APPENDIX

A. Related Work

Dexterous Manipulation. Dexterous manipulation has been a long-standing challenge in robotics. Many prior works have focused on tasks falling under these broad categories: in-hand manipulation, grasping, and post-grasp tool use. First, in-hand manipulation typically consists of rotating or translating objects with multi-finger control [17], [18], [19], [20], [21]. Next, dexterous grasping is a useful skill for automation or household tasks [4], [3], [22]. Then, grasping an object, the robot may place the object (pick-and-place) or use it as a tool, such as for drilling [23], pouring [10], wiping [12], scissor cutting [24], and more. While these tasks are challenging and important for researching dexterity, we propose drumming, a task that requires both in-hand control, post-grasp tool-based contact, and long-horizon robustness for playing long songs.

To learn dexterous policies, many classical methods use planning with precise models [25], [26], [27], [28], and propose grasps based on collision detection or optimization-based methods [5], [29], [7], [30], [6], [31]. Other works use sim-to-real training with reinforcement learning [32], [2], [1], [33] for in-hand tasks, or use synthetically generated grasp datasets in simulation with learning-based methods can leverage [34], [35], [8], [4], [3], [22]. Imitation learning from human demonstrations has been used for many dexterous applications [36], [37], [10], [38], [9], [23]. However, to reduce the data burden on designing simulated environments or teleoperating dexterous data, many works retarget human data [39], [40], [24], which may also utilize reinforcement learning in simulation [41], [42], [43], [44], [13]. Following many prior works, we train an RL policy in sim, which reduces the need to hire expert drummers to generate demonstration data. We use simple sim-to-real techniques, such as domain randomization, to showcase real-world drumming.

Robot Drumming. Robot drumming is an exciting subset of research in robotic music-playing. Many prior works design custom hardware for drum-playing [45], [46], [47], [48]. Our project assumes a more general embodiment – robot arm and hand – which requires learned dexterity to hit the drums, instead of special hardware designs. Recent works explore learning-based approaches to drumming, but likewise use either custom embodiments [49], or directly fix the stick to the embodiment in simulation [14], without examples of real-world drumming. Like [14], we learn our policy in simulation via reinforcement learning, but instead of focusing on humanoid control, we assume a realistic stick-holding embodiment, which allows sim-to-real transfer.

B. Low-Level Dexterous Policy

We design the reward function with three components: (1) *In-Hand Contact Rewards* for finger—stick interaction (2) *External Contact Rewards* for stick—drum interaction and (3) *Task Rewards* for drum-playing.

In-Hand Contact Rewards. In-hand control for the drum stick is paramount to drum playing. To enable this, we incorporate rewards targeting in-hand contact. First, we encourage finger-stick contact through a fingertip contact reward, a general reward function used in prior work [12] for finger-object interactions. Next, following human drum priors, we introduce the fulcrum reward, which specifically encourages the thumb and index finger to grasp the "fulcrum," the center of the drum stick, following human priors in drumming. This further enhances finger-stick contact based on drumming priors. The fingertip contact reward encourages the fingers to successfully touch and grasp the object, which in this case, allows us to hold and manipulate the stick. Next, we propose an arm penalty constraint, which reduces excessive arm movements, making in-hand contact more natural. This prevents the agent from manipulating the stick with arm movements, incentivizing the agent to develop fine-grained finger control. Energy minimization has been widely used in locomotion [50], [51], [52] to induce diverse gaits, and here, we adapt it for in-hand contact with the stick.

External Contact Rewards. To address external contact, we divide the challenges into two stages: (1) initiating contact and (2) maintaining long-horizon contact. We address the first with a trajectory reward and contact curriculum, and we address the second with a reactive grasp reward.

First, we would like our policy to successfully initiate external contact between the stick and drum. To achieve this, we apply a trajectory reward, which explicitly guides the speed and motion, hence guiding precise control of the contact force. In our case, we pre-compute the desired trajectories of the drumstick tip and end by modeling drum hits with a sinusoidal wave and interpolating between drum positions across hits. This approach is inspired by prior work on high-level trajectory planning and low-level control [12], [53], [13], [43]. In this drumming environment, both arm actions and hand actions can help optimize this objective. For instance, when transitioning in between drums, both hand and arm control are important for following the stick trajectory. However, closer to the drum, this primarily becomes a dexterous challenge, as the fingers should be initiating the contact with the drum, compared to excessive, large arm movements.

While this reward guides the hitting motion of the stick, the reward by itself is often insufficient to learn how to properly hit the drum. Specifically, during learning, the stick often rests on the drumhead, which often blocks the exploratory finger motions necessary for controlling the stick. To address this issue, we introduce a contact curriculum.

TABLE I

REWARD FUNCTIONS AND CURRICULUM. n_{contacts} is the number of fingertip contacts, $\mathbf{p}_{\text{thumb}}, \mathbf{p}_{\text{index}}, \mathbf{p}_{\text{stick}}$ are positions, $g(\cdot)$ is a shaping function from [15] mapping distances to $[0, 1]$, and $\mathbb{1}$ denotes indicator functions for grasp.

Reward	Formula	Explanation	Weight
		In-Hand Contact	
Fingertip	$\exp\left(-1/(n_{\text{contacts}} + \varepsilon)\right)$	Make fingertips contact the stick	1.0
Fulcrum	$g\big((\|\mathbf{p}_{\text{thumb}} - \mathbf{p}_{\text{stick}}\|_2 + \|\mathbf{p}_{\text{index}} - \mathbf{p}_{\text{stick}}\|_2)/2\big)$	Position thumb and index finger to hold the fulcrum	1.0
Arm	$\|\tau_{\text{arm_joints}}\| + \|v_{\text{arm_joints}}\|$	Penalize arm movement to incentivize finger movement	0.03
		External Contact	
Trajectory	$\mathbb{1}_{\text{is_grasped}} \cdot g(\|\mathbf{p}_{\text{stick}} - \hat{\mathbf{p}}_{\text{stick}}\|_2)$	Guide the stick with a reference trajectory	2.0
Contact Curriculum	N/A	Disable stick-drum contact for first N steps	N/A
		Task	
Drum Hit	$\mathbb{1}_{\text{is_stick_hit_drum}} \cdot \mathbb{1}_{\text{is_hit_window}}$	Hit drum according to music	1.0

TABLE II

OBSERVATION SPACE. L denotes lookahead horizon.

Observation	Dimension
Arm Proprioception	$7 \times n_{\text{hand}}$
Hand Proprioception	$20 \times n_{\text{hand}}$
Stick Head Proprioception	$3 \times n_{\text{hand}}$
Stick Tail Proprioception	$3 \times n_{\text{hand}}$
Trajectory Plan: Stick Head	$3 \times n_{\text{hand}} \times L$
Trajectory Plan: Stick Tail	$3 \times n_{\text{hand}} \times L$
Stick is Grasped	$1 \times n_{\text{hand}}$
Previously Played Drum	7 (discrete)
Next Drum to Play	7 (discrete)
Time Before Next Drum Hit	1

Initially, contact between the stick and the drum is disabled, allowing the agent to practice trajectory following in free space while following the trajectory reward. Contact is later reintroduced, enabling the policy to more effectively initiate drum hits. This curriculum helps the policy learn to initiate external contact, as we decompose the problem into first following a motion, and then learning reactive behaviors to continue following the trajectory with the external forces. This curriculum shares similarities with DexMachina [54], which uses virtual object controllers to prevent early failures caused by gravity. In contrast, our curriculum targets contact-related randomness – a more challenging source of instability – and is simpler, requiring no modifications to object assets.

Task Rewards. Similar to prior works [14], we add a sparse hit reward to check whether the drum is hit at a specified time.

C. Experimental Setup

1) Hardware Setup: We use two 7-DOF Franka Panda arms and two 20-DOF Tesollo DG-5F hand in both simulated bimanual and real-world unimanual tasks. We run policy inference at 20 Hz, and we use a PID joint position controller that runs at 100 Hz.

We showcase real-world drumming with a unimanual, drum pad and cymbal setup. We create a digital twin of the real world by matching the drum, cymbal, and stick positions and sizes. We apply randomization to improve robustness for sim-to-real.

To track the stick proprioception, we paint the end of the drumstick. We use color segmentation, depth readings from a RealSense camera, camera intrinsics, and camera extrinsics to project it into robot frame.

2) Evaluation Metrics: For bimanual song playing, we evaluate performance using two metrics: (1) the F1 score for song performance, and (2) the stick-hold ratio, defined as the fraction of time the stick remains held in hand across the total duration, which reflects the effectiveness of in-hand control. For dexterous fine control, we use (1) trajectory error, measuring how accurately the policy follows fine-grained trajectories, and (2) energy consumption, capturing the overall efficiency of the system.

3) Drum Songs: We use MIDI files to specify drum sequences, following prior work [15], [14]. For simulation experiments, we construct **Easy** and **Hard** tracks from an open-source MIDI drum dataset covering six different genres, using a bimanual setup with multiple drums. Easy tracks consist of repetitive rhythmic patterns designed to evaluate long-horizon stability, while Hard tracks involve more complex rhythms requiring coordinated hits across different drums.

We also design a Finger-Driven Control task that focuses on high-speed single-drum playing to test fine-grained finger dexterity.

D. Real-World Bimanual Song-Playing

We further showcase a real-world bimanual drum performance on our website. We use two songs—*Seven Nation Army* and *Everlong*—performed on a simplified setup with

two drums (snare and hi-hat). The slower song emphasizes larger arm movements, while the faster one requires higher finger dexterity. The results shown in the videos demonstrate that our framework is able to play real, bimanual songs robustly.

E. Ablations for Arm Control

We study the contribution of each component in the arm control pipeline through an ablation analysis of motion planning and residual RL. As shown in the accompanying videos, the full system that combines motion planning with residual RL achieves precise and stable drumming performance, reaching an F1 score of 1.0. When removing the residual RL policy, the system relies solely on kinematic planning and fails to perform fine-grained corrections during drum contact. This results in inaccurate hits, limited adaptability in finger control, and a reduced F1 score of 0.8. Further removing motion planning and training a policy purely with RL from scratch significantly degrades performance. In this setting, the policy struggles to coordinate large-scale arm movements with dexterous hand control, leading to frequent missed hits and an F1 score of 0.5. These results highlight that motion planning provides a strong structural prior for global coordination, while residual RL is critical for local correction and precise stick control.

F. Song Processing

Like prior works [15], [14], we specify songs by importing MIDI files into the environment. MIDI is a widely used representation that encodes instruments and timing in a concise, discrete format.

a) Bimanual Song Playing.: We obtain MIDI songs from an open-access website[1] . From this collection, we select six different genres of music, and for each track, we extract both an Easy and Hard segment. To ensure playability for the robot policy, all songs are slowed down by a factor of three. The Easy tracks feature repetitive loops, which we evaluate over 400 timesteps (20 sec) to test long-horizon control. The Hard tracks require hitting multiple drums with dynamic rhythms, and we evaluate these with 200 timesteps (10 sec). All tracks are performed using a bimanual setup.

b) Finger-Driven Control Experiment: For our Finger-Driven Control experiment (Section IV-C), we use an exercise that emphasizes finger dexterity by playing a single drum at a very high speed of up to 240 beats per minute (i.e., 4 hits per second). We train with 100 timesteps in a uni-manual setup.

c) Real-world Bimanual Song Playing: For real-world bimanual playing, we select two songs: *Seven Nation Army* by The White Stripes, with approximately 120 beats per minute (BPM), and *Everlong* by Foo Fighters, a faster and more dexterous piece with approximately 160 BPM. The first song primarily emphasizes larger arm movements, whereas the second requires greater finger dexterity due to its higher tempo and denser rhythmic patterns.

[1]https://mididrumfiles.com/tag/midi-files/

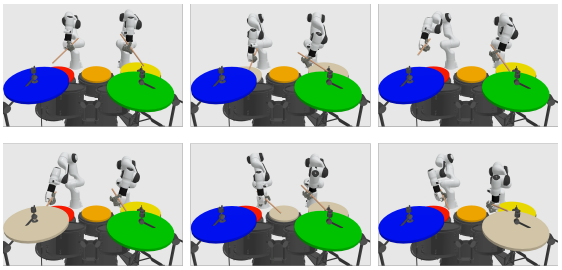

Fig. 6. **Bimanual Song-Playing Rollout.** We visualize 6 frames across a single song trajectory, with lighter colored drums and cymbals corresponding to a hit. Every song requires multiple combinations of drums to be hit.

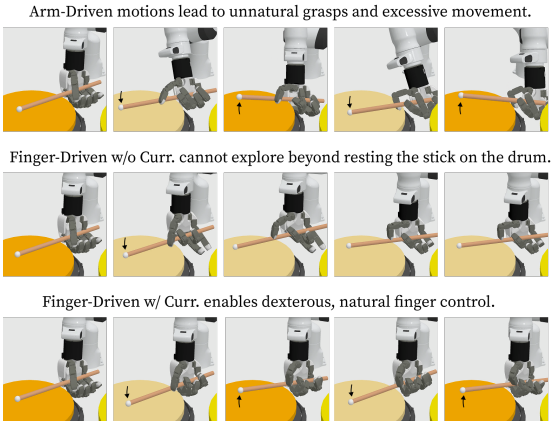

Fig. 7. **Sample Rollouts for the Dexterous Exercise.** For fast-playing exercises, the arm-driven policy moves to unnatural positions, leading to energy-intensive and potentially dangerous positions. Finger-driven control without the contact curriculum is unable to effectively learn dexterous control, whereas adding the contact curriculum leads to the most natural, effective drum hits. The lighter-colored drum head denotes that the drum is being hit by the stick.

To simplify the physical setup, we restrict the drum configuration to only a snare and a hi-hat and adapt the drum sheet accordingly. Specifically, notes corresponding to left or right cymbals are mapped to the hi-hat, while notes corresponding to toms are mapped to the snare.

G. Simulation Training

We train our policies with Proximal Policy Optimization [55], running for 60M steps on bimanual tasks and 40M steps on unimanual tasks. The training setup uses a discount factor of $\gamma = 0.8$, and 1024 parallel environments. We adopt generalized advantage estimation (GAE) with $\lambda = 0.9$ and employ a 3-layer MLP policy network with hidden dimensions of size 512. For the contact curriculum, contact between the drum stick and the drum pad is disabled during the first 10,000 steps.

For sim-to-real training, we incorporate domain randomization. At each step, uncorrelated Gaussian noise sampled from $\mathcal{N}(0, 0.05^2)$ is added independently to the proprioception and stick positions in the observation space. The stick's friction coefficient is perturbed with noise drawn from a uniform distribution $\mathcal{U}(-0.2, 0.2)$. In addition, control gains are scaled by a random factor sampled from $\mathcal{U}(0.9, 1.1)$ at environment initialization, and this factor is kept fixed for

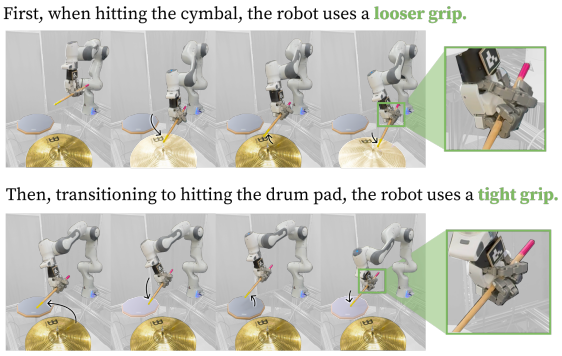

Fig. 8. **Real World Rollouts.** We visualize part of one closed-loop trajectory. The lighter colors for the cymbal and drum show when the robot hits them with the stick, and the arrows visualize the direction the stick was moving in. In this continuous rollout, we two cymbal hits and two drum hits. Notably, for the cymbal, which is not fixed and can rotate around the world vector, the robot strikes with a relatively loose grip (top). However, after hitting the drum pad, which is stable, the hand adjusts and forms a firmer grip (bottom), as shown by the movement of the index finger and thumb.

each environment throughout training.

H. Limitations and Future Work

We are excited about future directions for dexterous drumming. For one, DexDrummer cannot play multi-drum songs at human speed, and we slow down our songs in order to make it feasible. Future directions may explore how to play to drum tracks in real-time. Similarly, current experiments test performance for up to 400 timesteps (20 seconds), whereas real-world songs are often 3-5 minutes. Improving speed, robustness, and control is paramount to improved song performances.

For our real-world experiments, we currently only show a bimanual performance with two drums. Future work may showcase dexterous drumming with a full drum set. Furthermore, we do not incorporate sim-to-real techniques besides domain randomization, and further research into reducing the sim-to-real gap may lead to stronger song-playing.