The Sound of Simulation: Learning Multimodal Sim-to-Real Robot Policies with Generative Audio

Renhao Wang **Haoran Geng** **Tingle Li**

Feishi Wang **Gopala Anumanchipalli** **Trevor Darrell**

Boyi Li **Pieter Abbeel** **Jitendra Malik** **Alexei A. Efros**

University of California, Berkeley

Abstract: Robots must integrate multiple sensory modalities to act effectively in the real world. Yet, learning such multimodal policies at scale remains challenging. Simulation offers a viable solution, but while vision has benefited from high-fidelity simulators, other modalities (e.g. sound) can be notoriously difficult to simulate. As a result, sim-to-real transfer has succeeded primarily in vision-based tasks, with multimodal transfer still largely unrealized. In this work, we tackle these challenges by introducing MULTIGEN, a framework that integrates large-scale generative models into traditional physics simulators, enabling multisensory simulation. We showcase our framework on the dynamic task of robot pouring, which inherently relies on multimodal feedback. By synthesizing realistic audio conditioned on simulation video, our method enables training on rich audiovisual trajectories—without any real robot data. We demonstrate effective zero-shot transfer to real-world pouring with novel containers and liquids, highlighting the potential of generative modeling to both simulate hard-to-model modalities and close the multimodal sim-to-real gap. Code, models and data available at: https://multigen-audio.github.io

Keywords: generative modeling, real2sim, sim2real, multimodal learning

1 Introduction

Multimodal perception is essential for robust and adaptive human behavior, whether it be grasping an object by sight and feel, or pouring a drink while relying on auditory cues. For robots to achieve similar generalization and adaptability, they must also learn to integrate multiple sensory inputs effectively. However, acquiring large-scale multimodal datasets for robot learning is a significant challenge. Synchronized video, audio, tactile and action data require precise calibration, expensive hardware, and labor-intensive setup. As a result, even recent efforts to collect robot datasets at unprecedented scale still lack diverse sensory constellations [1, 2].

A promising alternative to costly real-world data collection is simulation-based learning, where robots acquire skills in synthetic environments before adapting them to the real world via sim-to-real transfer. This strategy has enabled impressive progress in vision-based policy learning, particularly for tasks like locomotion and basic manipulation (e.g. grasping or pick-and-place). However, more dexterous and dynamic behaviors—which often require multimodal feedback—remain out of reach. Sim-to-real transfer for such multimodal tasks has yet to be fully realized, due in large part to the difficulty of simulating non-visual modalities. Sound, for example, is notoriously hard to model in simulation: its physical propagation depends on complex wave dynamics and material interactions, making high-fidelity audio simulation both expensive and impractical at scale.

In this work, we address both challenges—multimodal simulation and sim-to-real transfer—with MULTIGEN, a framework which augments traditional physics-based simulators with large-scale

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

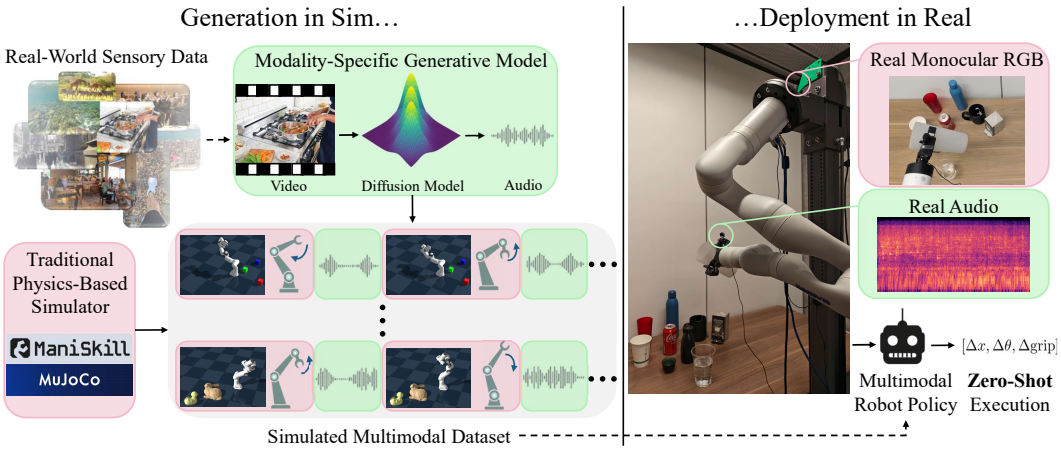


Figure 1: **Overview of our MULTIGEN framework.** We train a generative model on real-world sensory data to capture modalities that would otherwise be difficult to simulate (e.g. audio). Augmenting traditional simulators with these generative models enables generating synthetic multimodal data at scale, and learning multimodal policies that can translate *zero-shot* to the real world.

pretrained generative models. Specifically, MULTIGEN is a hybrid pipeline that runs a generative model in parallel with a physics engine, allowing for the synthesis of realistic, task-relevant sensory signals that complement visual and action data. To demonstrate the effectiveness of this approach, we instantiate MULTIGEN on the task of robot pouring, an inherently multimodal task that strongly relies on auditory feedback due to liquid occlusion, transparency, and visual ambiguity. Our generative model is pretrained on diverse real-world audiovisual data and finetuned on in-the-wild human pouring videos, enabling it to produce task-relevant sounds that enhance policy learning. We show that MULTIGEN enables high-quality audio synthesis that more faithfully matches real-world pouring sounds compared to standard data augmentation techniques. We also demonstrate that policies trained with MULTIGEN transfer effectively to real-world pouring of novel liquid types across diverse container geometries in a zero-shot manner—without requiring any real-world robot sensory or action data. These results highlight the potential of MULTIGEN to close the multimodal data gap, paving the way for learning more perceptually rich robotic systems at scale.

2 Related Work

Multimodal robot learning. There has been increasing interest in integrating multiple sensory modalities for robotics. Prior work has heavily explored leveraging vision and touch across various tasks, such as grasping objects [3, 4], representation learning [5], or for fine motor control [6, 7, 8, 9]. But audio sensing has received relatively little attention in robotics. Most prior works incorporating sound focus on speech-based interaction [10, 11] or environment and action recognition [12, 13]. Few studies leverage audio directly for manipulation, such as using impact sounds for material classification [14, 15] or pouring sounds to estimate liquid quantity [16]. More recently, interest in contact-rich tasks with heavy occlusion and clear acoustic signal, such as dynamic pouring or object search and retrieval from confined spaces, have brought policies employing vision, tactile and audio to the fore [17, 18, 19]. While these approaches all use teleoperated datasets of limited size, our work in contrast leverages generative models for synthesizing realistic multimodal data at scale.

Simulation of sound. Traditional approaches to simulating sound in robotics and graphics rely on physics-based methods, such as numerical wave propagation [20, 21] or ray tracing [22]. While these methods can achieve high-fidelity audio, they are computationally expensive and often require detailed material properties [23, 24], making them impractical for large-scale robot learning.

More recently, generative models have shown promise for synthesizing realistic audio from visual and contextual inputs [25, 26]. Advances in audiovisual learning have enabled models to generate synchronized soundtracks for silent videos [27, 28], as well as infer material properties from impact sounds [14, 29, 30]. However, these generative approaches have primarily been explored in media

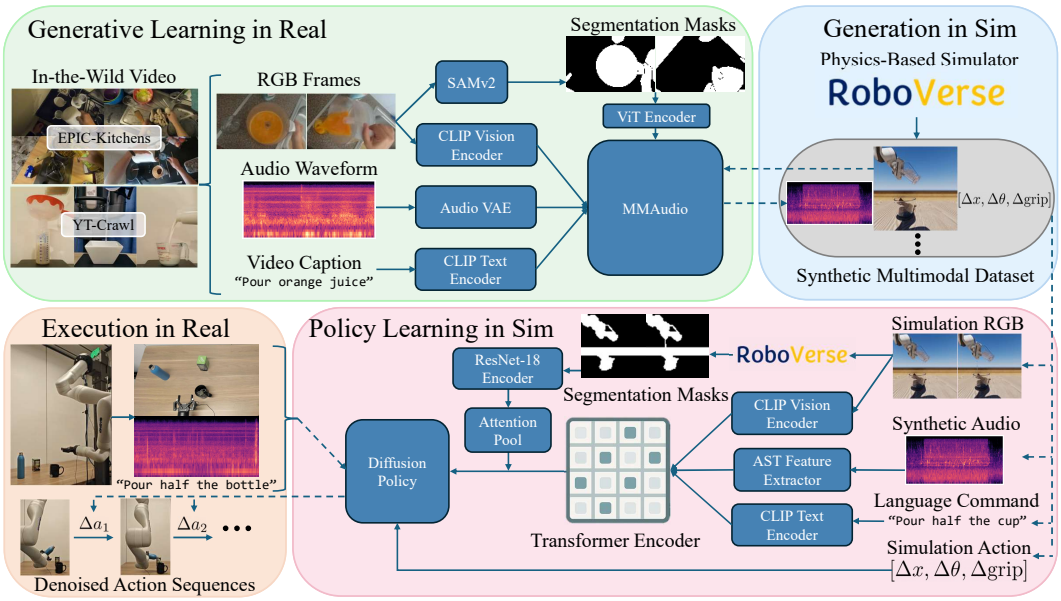


Figure 2: **Components in the MULTIGEN instantiation for robot pouring.** We first finetune a video-to-audio diffusion model (e.g.MMAUDIO) on in-the-wild video. Conditioning on simulation video then enables generating multimodal simulation trajectories replete with audio. We then train a policy (e.g. Diffusion Policy) on this multimodal dataset, before evaluating zero-shot on a real setup.

synthesis and computer vision, with limited prior work applying them to robotics. Our work is the first to leverage generative audio models within a multimodal robot learning framework, demonstrating their utility in sim-to-real transfer for tasks like pouring.

Generative models in simulation. Generative models have increasingly been integrated into simulation frameworks to enhance data diversity and realism. LucidSim [31] investigates replacing visual domain randomization with text-to-image generation models. Gen2Sim [32] generates 3D object models and their corresponding URDFs for scaling simulation assets. Eureka [33] uses the vast general knowledge of LLMs to generate reward functions for training policies via RL in sim. Our work departs from this trend of using generative models for augmenting existing data or capabilities. Instead, we show how they can be used for introducing entirely new sensory streams in simulation.

3 MultiGen

3.1 Framework Overview

MULTIGEN consists of two main components:

1. **A physics-based simulation engine** which provides a controllable environment for robot learning. The simulator models the visual scene, rigid-body dynamics, fluid interactions, robot kinematics, etc., and allows for the collection of visual and proprioceptive inputs.

2. **A generative model** that conditions on simulator information and generates complementary sensory signals, such as audio, that are traditionally difficult to simulate.

These two components interact in a hybrid pipeline, where the physics engine provides structured inputs (e.g. visual renders, robot and object information), which are then used by the generative model to synthesize additional sensory signals. This interplay ensures that the generated data is physically consistent with the scene while bypassing more computationally expensive simulation processes. We hypothesize that with sufficiently powerful components —i.e. a high-fidelity physics engine and a well-trained generative model—zero-shot transfer is achievable without finetuning on real-world data. An overview of our proposed framework is depicted in Fig. 2.

To illustrate the viability of this hypothesis, we instantiate MULTIGEN on a concrete multimodal task: robot pouring. This task inherently requires multimodal perception due to occlusion effects, visual

ambiguities, and the strong dependence on auditory feedback for estimating liquid flow and container fill level. For our simulation environment, we leverage the newly released RoboVerse framework, which provides high-quality physics simulation for robotic manipulation. For our generative model, we build upon MMAudio, a large-scale pretrained video-to-audio model, which we adapt to generate realistic pouring sounds from simulated visual inputs.

3.2 Generative Audio Model

A primary challenge in integrating generative models for multimodal robot learning is selecting the appropriate conditioning signals. While language-conditioned audio models like AudioLDM [34] can generate diverse sounds based on text prompts, they lack the fine-grained control necessary for continuous tasks like pouring. Key variables such as pour height, container volume, liquid properties, and material interactions require pixel-level information, which language alone struggles to specify.

To address this, we use MMAudio, a state-of-the-art video-to-audio model that predicts audio from raw video frames [28]. MMAudio employs a multimodal transformer architecture that jointly processes visual, audio, and textual features using cross-attention mechanisms. The model is trained generatively with a conditional flow matching loss. Further details are available in [28].

MMAudio allows the generation of sound directly from the visual scene, capturing crucial task-relevant cues. However, we find that its zero-shot performance is inadequate for our purposes (see ablations in Section 4.4). We identify two key limitations:

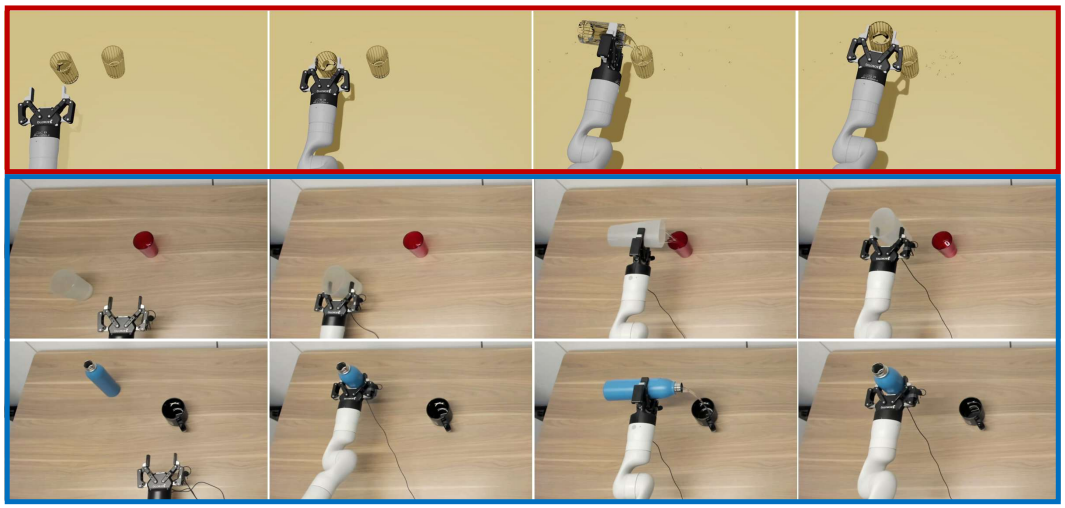
> **1. MMAudio is trained on diverse but suboptimal data.** The model has been exposed to large-scale audiovisual datasets, but much of this data consists of noisy, lower-frequency, Foley sounds dramatically different from liquid sounds.
>
> **2. Pouring is a long-tail audio event.** The pretraining corpuses of MMAudio contain relatively few pouring instances [35, 36, 37]. This leads to poor representations of task-specific acoustic cues, such as pitch variation as a container fills, dependencies on liquid viscosity, and transient impulse sounds at the start and stop of a pour.

To address these limitations, we finetune MMAudio on a curated dataset of real-world pouring sounds collected from (1) EPIC-Kitchens: a large-scale first-person cooking dataset containing diverse instances of liquid manipulation [38, 39], and (2) our own YouTube crawl. Here, we extracted high-quality pouring clips from Internet video sharing platforms, featuring various liquids (e.g., water, coffee, soda, soup) and containers (e.g., glass, plastic, ceramic). Our audiovisual clips range from 5 to 60 seconds, and total 1031 videos in total. This finetuning allows the model to better capture the nuances of real-world pouring sounds, improving generalization to novel pouring conditions.

However, this adaptation introduces a new challenge: the domain gap between real-world and simulated video. Since MMAudio has been trained primarily on natural video, its performance may degrade when applied to synthetic RGB frames from simulation, which exhibit a different visual distribution. To bridge this gap, we introduce a semantic segmentation conditioning mechanism using SAMv2 (Segment Anything Model v2) [40]. Specifically, for each video, we extract an initial RGB frame of size $H \times W \times 3$, and obtain a segmentation mask tensor of size $H \times W \times C$, where C corresponds to a predefined set of task-relevant object classes (e.g., cup, mug, water, juice). We then use SAMv2 to propagate these masks across the entire video sequence, ensuring consistent segmentation. Finally, we condition MMAudio on both the RGB frames and the segmentation masks, allowing the model to focus on semantically meaningful regions. Concretely, we introduce an additional set of projection and cross-attention parameters to inject into the transformer backbone (details in appendix). This conditioning strategy improves robustness to the sim-to-real visual gap, and enables our model to produce more physically accurate and task-relevant audio signals.

3.3 Simulation Framework

To effectively train multimodal robotic policies in simulation, our framework requires a simulator that meets several key desiderata. First, while works such as [41] or [31] seek to compensate for poor



(i) Start (ii) Grasp (iii) Pour (iv) End

Figure 3: **Comparison of various simulation trajectories and real-world executions.** Trajectories motion-planned in our photorealistic simulator mirror execution traces in the real world.

visual simulation via generative models, our main focus is to inject real world modalities to enable entirely new sensing capabilities. Thus, we prioritize a **photorealistic simulation environment** that provides high visual fidelity out of the box. Second, the simulator must be **computationally efficient** to support large-scale trajectory generation. Liquid simulation, in particular, is traditionally expensive due to the complexity of modeling soft-body and fluid dynamics, which often rely on particle-based solvers. Ideally, the simulator must also support parallelized data collection, enabling multiple instances to run simultaneously with efficient physics simulation and rendering. Third, the simulator should provide a **diverse and customizable environment**, both in terms of natural diversity in assets and the ability to introduce controlled variability.

To meet these requirements, we use RoboVerse, a recently released scalable simulation platform that supports high-fidelity rendering and highly-parallelized execution. RoboVerse also offers a diverse set of predefined assets, including cups, bottles, and liquid types. To further enhance diversity, we randomly scale object dimensions (e.g. cup openings, bottle heights) to diversify object geometries. RoboVerse is also equipped with a powerful DR library that allows for control over a wide-ranging set of crucial simulation parameters.

Environment setup. We begin by constructing a simulation environment that coarsely mirrors real-world conditions. We approximate camera placement based on real-world setups but maintain loose constraints (e.g. precise camera extrinsics and lighting conditions are not required). Unlike traditional high-fidelity digital twin approaches, which aim for near-perfect reconstruction [42], our hybrid approach merely requires approximate modeling, as we will rely on structured domain randomization (DR) to improve generalization. Concretely, to bridge the sim-to-real gap, we apply DR to key simulation parameters, including: liquid properties (density, viscosity, color, transparency, surface friction interactions), camera pose (perturbations in position and rotation), lighting conditions (randomized intensity, shadows, and reflections), and background and table textures. A full specification for our DR setup can be found in the appendix. These variations improve generalization to unseen pouring scenes, and ensure that policies are robust to naturally occurring real-world variations.

Generating vision and action. To generate diverse and structured pouring trajectories, we use a motion planning-based pouring strategy. Every trajectory consists of a pouring container and a receiving container, with the robot initialized from the same home position. The pouring container is randomly pre-filled to a $70\% \pm 5\%$ capacity. We sample a random grasp point within a 5% offset from the container center, ensuring a consistent but slightly varied grasp strategy. Then, the robot moves the container above the receiving container, ensuring alignment with its opening geometry. This pour height is again randomly sampled within predefined limits to ensure variability. Next, one of four

discrete pouring amounts is sampled (one-quarter, one-half, three-quarters, or full pour). Based on the initial fill level and the selected pouring fraction, we compute a deterministic tilt angle to achieve the desired liquid transfer. Finally, once pouring is completed, the robot untilts the container, and the episode terminates. We show in Fig. 3 these various keypoints in both trajectories generated in simulation, as well as real-world execution traces from learned policies. For simplicity, we use motion planning to generate trajectories in simulation, but our framework could be adapted to use any number of approaches depending on the chosen task (e.g. reinforcement learning or other pretrained policies with verification mechanisms, or virtual teleoperation for collecting human-guided demonstrations.)

Generating audio. To generate synchronized multimodal data, we run a parallelized simulation pipeline with two threads. The first thread executes the motion-planned pouring trajectory, recording RGB frames, proprioception, and action sequences. However, the RGB frames from cameras corresponding to the real-world setup may be heavily occluded depending on the pouring scenario, or misaligned with the camera poses of our finetuning dataset. Thus, we crucially also collect video from a dedicated virtual camera positioned frontal to the receiving container. This view is in distribution with the real-world pouring videos used for finetuning our video-to-audio generative model. The second thread then leverages this virtual stream to process completed trajectories and generate audio.

4 Experiments

In this section, we begin by describing our policy learning and evaluation methodology, as well as our real-world setup. We then dive into results which definitively answer two main questions: (1) First, to what extent do policies learned in simulation via MULTIGEN transfer to the real world? (2) Second, how well does the generated audio from MULTIGEN align with real-world pouring sounds, both perceptually and in terms of task-relevant metrics?

4.1 Evaluation Methodology: Baselines and Benchmarks

We choose a state-of-the-art policy learning approach to train and evaluate our framework. Namely, we learn a diffusion policy [19, 43], a denoising diffusion model which conditionally generates actions given state, vision and audio inputs. Importantly, our policy is trained from scratch using only proprioception, vision, and audio data generated entirely by MULTIGEN, as described in Section 3.3. To systematically analyze the impact of multimodality, we also train a variant which omits audio.

For evaluation, we establish a benchmark spanning diverse pouring conditions. Our evaluation considers variations in container materials (e.g. plastic, paper, metal), liquid types (e.g. water, juice, soda, hot liquids), and occlusion levels (e.g. transparent vs. opaque containers). By introducing realistic distribution shifts, we assess the adaptability and robustness of policies trained in simulation. The full suite of conditions can be seen in Table 1, with details available in the appendix. For each condition, we choose three pairs of random starting positions for the pouring and receiving containers. For each position pair, we deploy the policy four times, once with each of the language instructions described in Section 3.3, leading to twelve total evaluations for each condition. All policies are evaluated zero-shot, meaning *no demonstrations or real robot data have been employed.*

To compensate for differences in relative difficulty or target volume across language commands and container setups, our evaluation metric is Normalized Mean Absolute Error (NMAE):

$$NMAE = \frac{|\text{ Actual Poured Amount - Desired Target Amount }|}{\text{Desired Target Amount}}. \tag{1}$$

Note that the desired target amount is given by the known initial volume in the pouring container and the language command (e.g. pour half). We average over the three distinct locations per command, before averaging across the four commands.

4.2 Physical Setup and Implementation Details

Robot hardware and setup. Our experiments are conducted on a Kinova Gen3 robot equipped with a Robotiq 2F-85 adaptive gripper. Monocular RGB is provided via a single Logitech BRIO 4K web

Opaque	water- white cup- red cup	coffee- metal thermos- paper cup	sake- sake carafe- sake cup	water- metal thermos- metal mug
DIFFUSION POLICY(V)	0.54 ± 0.23	0.54 ± 0.20	0.68 ± 0.30	0.47 ± 0.29
DIFFUSION POLICY(V + A)	$\mathbf{0.44} \pm \mathbf{0.19}$	$\mathbf{0.33} \pm \mathbf{0.17}$	$\mathbf{0.43} \pm \mathbf{0.28}$	$\mathbf{0.37} \pm \mathbf{0.15}$
Transparent	water- white cup- plastic cup	juice- plastic bottle- plastic cup	soda- metal can- plastic cup	juice- plastic bottle- glass mug
DIFFUSION POLICY(V)	0.53 ± 0.20	0.46 ± 0.20	0.43 ± 0.20	0.49 ± 0.21
DIFFUSION POLICY(V + A)	$\mathbf{0.42} \pm \mathbf{0.27}$	$\mathbf{0.38} \pm \mathbf{0.21}$	$\mathbf{0.39} \pm \mathbf{0.22}$	$\mathbf{0.43} \pm \mathbf{0.18}$

Table 1: **Main evaluation results on our pouring benchmark (lower is better).** We report average normalized mean absolute error (NMAE) and 1 std. dev. error. The top four tasks all involve opaque containers, and the bottom four tasks involve translucent containers. Results are computed over twelve seeds (four language commands evaluated for three random locations each.)

camera mounted in an egocentric position. Audio is obtained via a MAONO omnidirectional USB lapel microphone mounted at the end effector, capturing 24-bit audio at 192 kHz. For all tasks, we generate 6-DoF Cartesian space delta end-effector commands at a policy frequency of 10 Hz. The MoveIt IK library converts these commands to a desired 7-DoF joint action. A full depiction of our workspace and data setup can be found in the appendix.

Policy architecture details. For our audiovisual diffusion policy, we follow [19] and use a CLIP-pretrained ViT-B/16 model [44] to encode RGB frames. We use an audio spectrogram transformer (AST) [45] to encode audio. Audio is downsampled from 24-bit, 192 kHz into 16-bit, 16 kHz, before conversion to a log-mel spectrogram via FFT with temporal window 400, hop length 160 and 64 mel filterbanks. Complete training hyperparameters are available in the appendix.

4.3 Results: Zero-Shot Sim-to-Real Transfer

Our main results are presented in Table 1, demonstrating strong sim-to-real transfer. In particular, policies trained with MULTIGEN achieve high success rates in real-world pouring tasks, suggesting that our simulation pipeline produces transferable policies without requiring real-world data for training (average NMAE: **0.46**). Critically, the multimodal vision + audio diffusion policy variant outperforms the vision-only baseline, highlighting the importance of auditory feedback in our pouring task. Specifically, including audio induces a **23.3%** average reduction in NMAE. Pouring to and from opaque containers tends to benefit the most of audio, with an average NMAE reduction of **29.4%** (compared to a **16.2%** reduction for transparent containers). This is intuitive: in settings where visual feedback is limited, audio plays a crucial role in estimating flow rate and container fill levels.

4.4 Results: Assessing Audio Generation Quality

In this section, we rigorously evaluate the impact of integrating a generative audio model into the simulation pipeline. Our goal is to quantify the benefits of MULTIGEN by comparing it to explicit data augmentation techniques and measuring how well the generated audio supports policy learning. We evaluate across three key axes:

1. **Diversity:** How diverse is the synthetic audio data generated by MULTIGEN?
2. **Fidelity:** How well does MULTIGEN audio match real-world pouring sounds?
3. **Usefulness:** Can MULTIGEN enable scalable learning of multimodal policies?

To establish a baseline, we collect 10 pouring demonstrations in the real-world, using a random selection of pouring and receiving containers. Pouring heights and initial container locations are randomized (details available in the appendix.) Then, we follow the data augmentation protocol from MANIWAV [19], where background noises are overlaid onto training audio. Specifically, [19] augments audio from human demonstrations with i) randomly sampled environment noises from

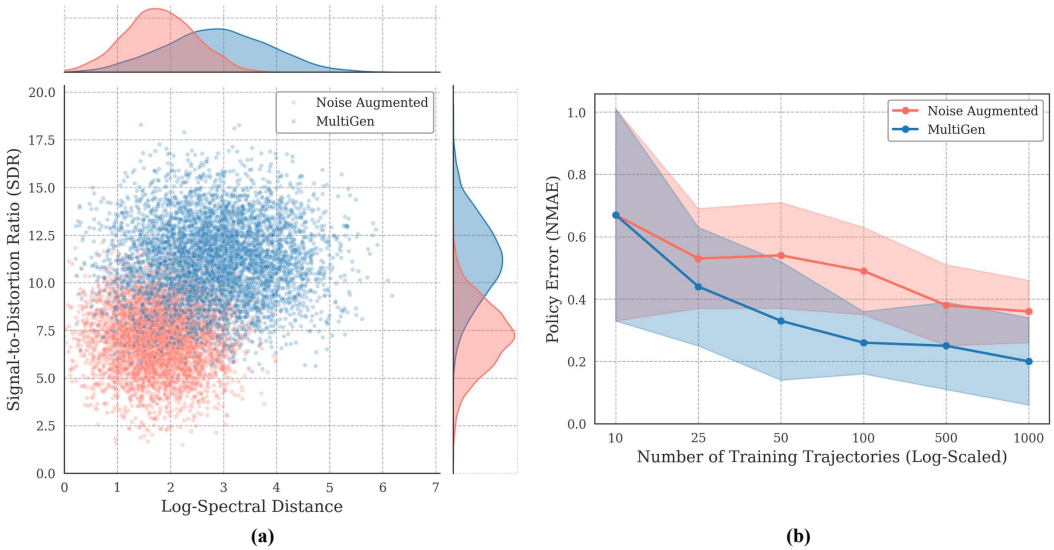


Figure 4: **Comparison between MULTIGEN and standard data augmentation used in robotics. (a) Left:** MULTIGEN produces audio that is both more diverse (higher Log-Spectral Distance) and more accurate (higher Signal-to-Distortion Ratio) than traditional additive noise augmentation. **(b) Right:** This higher quality audio allows policies trained with MULTIGEN to demonstrate better scaling properties (i.e. lower policy error across increasing dataset sizes.)

ESC-50 [46] and ii) robot motor sounds from 10 task-specific trajectories. These are overlayed onto the original audio signal with 50% probability. We generate 1000 samples using this augmentation approach, and 1000 samples from our finetuned generative audio model. For the latter, we condition on RGB and SAMv2 masks from the 10 pouring demonstrations to generate the corresponding audio.

To assess diversity, we compute the *minimum* log-spectral distance (LSD) between each generated sample and the collected demonstrations. This allows us to measure how much variability each approach introduces relative to real-world audio. To assess fidelity, we measure the signal-to-distortion ratio (SDR). We then plot these two quantities against each other, as shown in Fig. 4a. We observe that MULTIGEN generates a significantly broader range of audio samples, producing a wider marginal distribution over spectral distances compared to explicit augmentations. Crucially, despite greater diversity, MULTIGEN samples remain physically consistent, exhibiting higher SDR than augmented audio, even for samples far from the training distribution. This suggests that MULTIGEN achieves strong generalization, synthesizing diverse yet dynamically accurate audio.

Finally, to validate the impact on policy learning, we train a diffusion policy using either dataset. We compare both policies on a subset of the benchmark described in Section 4.1, including two opaque and two transparent pour settings. As shown in Fig. 4b, results indicate that MULTIGEN-trained policies significantly outperform augmentation-based policies. Moreover, MULTIGEN-trained policies exhibit stronger scaling with increasing number of generated trajectories, compared to naïve augmentation. Taken together, these ablations confirm that MULTIGEN generates more realistic, task-relevant audio, and that this property directly leads to improved multimodal policy learning.

5 Conclusion

In this work, we introduced MULTIGEN, a novel framework for integrating generative multimodal simulation into robot learning. By augmenting physics-based simulators with large-scale generative models, we demonstrated that sim-to-real policy learning can leverage rich sensory feedback beyond vision and proprioception. We instantiated MULTIGEN in the context of robot pouring, a task where auditory cues are critical. MULTIGEN enabled zero-shot sim-to-real transfer, facilitating generalization to challenging real-world pouring without requiring real robot sensory or action data. Our results highlight the broader potential of generative models to fundamentally expand simulation capabilities by injecting entirely new sensory modalities into training environments.

6 Limitations

While MULTIGEN demonstrates strong multimodal sim-to-real transfer for robot pouring, a few limitations remain. First, our approach relies on projecting real-world conditions into simulation, requiring approximate alignment between simulated and real environments. Although we only enforce loose constraints (e.g. approximate camera placement and correct robot embodiment), this still necessitates some manual setup. However, ongoing advancements in real-to-sim techniques can further automate this process, making it more scalable and reducing human effort. Second, our trained policies exhibit limited generalization to extreme container geometries. For instance, significantly larger or irregularly shaped containers can introduce unmodeled physical dynamics (e.g., unexpected liquid sloshing) that degrade performance. Expanding the diversity of simulated training assets and incorporating adaptive policies that reason about novel container shapes could mitigate this issue. Third, while our policies successfully execute pouring tasks in controlled real-world settings, they do not explicitly account for collision avoidance when navigating cluttered environments. In practical deployments, additional obstacle-aware motion planning or integrated perception modules would be required to prevent unintended collisions with surrounding objects.

Acknowledgments

The authors would like to thank Qiyang Li and Phillip Isola for helpful discussions on experimental design. Philipp Wu also contributed significantly to an earlier version of this work. Yuvan Sharma assisted with real-world setup. RW is supported in part by the Toyota Research Institute and ONR MURI. PA holds concurrent appointments as a Professor at UC Berkeley and as an Amazon Scholar. This paper describes work performed at UC Berkeley and is not associated with Amazon.

References

[1] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 1

[2] A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 1

[3] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018. 2

[4] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine. The feeling of success: Does touch sensing help predict grasp outcomes? *arXiv preprint arXiv:1710.05512*, 2017. 2

[5] F. Yang, C. Feng, Z. Chen, H. Park, D. Wang, Y. Dou, Z. Zeng, X. Chen, R. Gangopadhyay, A. Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26340–26353, 2024. 2

[6] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International conference on robotics and automation (ICRA)*, pages 8943–8950. IEEE, 2019. 2

[7] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv preprint arXiv:2404.16823*, 2024. 2

[8] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023. 2

[9] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine. Manipulation by feel: Touch-based control with deep predictive models. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 818–824. IEEE, 2019. 2

[10] J. Thomason, A. Padmakumar, J. Sinapov, N. Walker, Y. Jiang, H. Yedidsion, J. Hart, P. Stone, and R. Mooney. Jointly improving parsing and perception for natural language commands through human-robot dialog. *Journal of Artificial Intelligence Research*, 67:327–374, 2020. 2

[11] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn. Yell at your robot: Improving on-the-fly from language corrections. *arXiv preprint arXiv:2403.12910*, 2024. 2

[12] D. Gandhi, A. Gupta, and L. Pinto. Swoosh! rattle! thump!–actions that sound. *arXiv preprint arXiv:2007.01851*, 2020. 2

[13] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9701–9707. IEEE, 2020. 2

[14] S. Clarke, N. Heravi, M. Rau, R. Gao, J. Wu, D. James, and J. Bohg. Diffimpact: Differentiable rendering and identification of impact sounds. In *Conference on Robot Learning*, pages 662–673. PMLR, 2022. 2

[15] S. Clarke, T. Rhodes, C. G. Atkeson, and O. Kroemer. Learning audio feedback for estimating amount and flow of granular material. *Proceedings of Machine Learning Research*, 87, 2018. 2

[16] H. Liang, S. Li, X. Ma, N. Hendrich, T. Gerkmann, F. Sun, and J. Zhang. Making sense of audio vibration for liquid height estimation in robotic pouring. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5333–5339. IEEE, 2019. 2

[17] H. Li, Y. Zhang, J. Zhu, S. Wang, M. A. Lee, H. Xu, E. Adelson, L. Fei-Fei, R. Gao, and J. Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation. *arXiv preprint arXiv:2212.03858*, 2022. 2

[18] M. Du, O. Y. Lee, S. Nair, and C. Finn. Play it by ear: Learning skills amidst occlusion through audio-visual imitation learning. *arXiv preprint arXiv:2205.14850*, 2022. 2

[19] Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, B. Burchfiel, and S. Song. Maniwav: Learning robot manipulation from in-the-wild audio-visual data. In *8th Annual Conference on Robot Learning*, 2024. 2, 6, 7

[20] T. Tsuchiya. Numerical simulation of sound wave propagation with sound absorption using digital huygens' model. *Japanese Journal of Applied Physics*, 46(7S):4809, 2007. 2

[21] A. Rungta, C. Schissler, R. Mehra, C. Malloy, M. Lin, and D. Manocha. Syncopation: Interactive synthesis-coupled sound propagation. *IEEE transactions on visualization and computer graphics*, 22(4):1346–1355, 2016. 2

[22] C. Schissler and D. Manocha. Interactive sound propagation and rendering for large multi-source scenes. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2016. 2

[23] C. Chen, C. Schissler, S. Garg, P. Kobernik, A. Clegg, P. Calamia, D. Batra, P. Robinson, and K. Grauman. Soundspaces 2.0: A simulation platform for visual-acoustic learning. *Advances in Neural Information Processing Systems*, 35:8896–8911, 2022. 2

[24] C. Matl, Y. Narang, D. Fox, R. Bajcsy, and F. Ramos. Stressd: Sim-to-real from sound for stochastic dynamics. *arXiv preprint arXiv:2011.03136*, 2020. 2

[25] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12, 2016. 2

[26] H. K. Cheng, M. Ishii, A. Hayakawa, T. Shibuya, A. Schwing, and Y. Mitsufuji. Taming multi-modal joint training for high-quality video-to-audio synthesis. *arXiv preprint arXiv:2412.15322*, 2024. 2, 15

[27] T. Li, R. Wang, P.-Y. Huang, A. Owens, and G. Anumanchipalli. Self-supervised audio-visual soundscape stylization. In *European Conference on Computer Vision*, pages 20–40. Springer, 2024. 2

[28] L. Ruan, Y. Ma, H. Yang, H. He, B. Liu, J. Fu, N. J. Yuan, Q. Jin, and B. Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10219–10228, 2023. 2, 4

[29] Z. Zhang, Q. Li, Z. Huang, J. Wu, J. Tenenbaum, and B. Freeman. Shape and material from sound. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[30] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016. 2

[31] A. Yu, G. Yang, R. Choi, Y. Ravan, J. Leonard, and P. Isola. Learning visual parkour from generated images. In *8th Annual Conference on Robot Learning*, 2024. 3, 4

[32] P. Katara, Z. Xian, and K. Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6672–6679. IEEE, 2024. 3

[33] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023. 3

[34] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023. 4

[35] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024. 4

[36] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020. 4

[37] C. D. Kim, B. Kim, H. Lee, and G. Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132, 2019. 4

[38] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020. 4

[39] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. 4

[40] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4

[41] A. Pashevich, R. Strudel, I. Kalevatykh, I. Laptev, and C. Schmid. Learning to augment synthetic images for sim2real policy transfer. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2651–2657. IEEE, 2019. 4

[42] Y. Jiang, S. Yin, K. Li, H. Luo, and O. Kaynak. Industrial applications of digital twins. *Philosophical Transactions of the Royal Society A*, 379(2207):20200360, 2021. 5

[43] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 6

[44] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[45] Y. Gong, Y.-A. Chung, and J. Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021. 7

[46] K. J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015. 8

[47] Y. Park and P. Agrawal. Using apple vision pro to train and control robots, 2024. 14

[48] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 17

A Details on Evaluation Benchmarks

In this section, we detail the physical setups involved in our main benchmark (i.e. the evaluations in Table 1), and our ablation (i.e. the 10 base pouring demonstrations used in Section 4.4, and the evaluation setting for Fig. 4b.)

A.1 Main Benchmark

Water-White Cup-Red Cup. The pouring container is a tall opaque plastic cup with total volume 491 mL. The receiving container is a plastic red cup with total volume 284 mL. The pouring liquid is tap water.

Coffee-Metal Thermos-Paper Cup. The pouring container is a tall, thin blue metal thermos with total volume 500 mL. The receiving container is a disposable paper cup with total volume 287 mL. The pouring liquid is hot instant coffee.

Sake-Sake Carafe-Sake Cup. The pouring container is an irregularly shaped sake carafe with total volume 216 mL. The receiving container is a small sake cup with total volume 35 mL. The pouring liquid is room temperature Junmai sake.

Water-Metal Thermos-Metal Mug. The pouring container is a tall, thin blue metal thermos with total volume 500 mL. The receiving container is a black metal mug with total volume 300 mL. The pouring liquid is hot water.

Water-White Cup-Plastic Cup. The pouring container is a tall opaque plastic cup with total volume 491 mL. The receiving container is a clear disposable plastic cup with total volume 356 mL. The pouring liquid is tap water.

Juice-Plastic Bottle-Plastic Cup. The pouring container is a SimplyOrange bottle with total volume 450 mL. The receiving container is a clear disposable plastic cup with total volume 356 mL. The pouring liquid is orange juice with pulp.

Soda-Metal Can-Plastic Cup. The pouring container is asmall round Coca Cola soda can with total volume 222 mL. The receiving container is a clear disposable plastic cup with total volume 356 mL. The pouring liquid is Coca Cola soda.

Juice-Plastic Bottle-Glass Mug. The pouring container is a SimplyOrange bottle with total volume 450 mL. The receiving container is a black glass mug with total volume 364 mL. The pouring liquid is orange juice with pulp.

A.2 Ablation Benchmark

To collect 10 pouring demonstrations, we choose the following subset from the previously described 8 main evaluation settings.

- water-white cup-red cup (x2)
- water-metal thermos-metal mug (x2)
- water-white cup-plastic cup (x2)
- juice-plastic bottle-plastic cup (x2)
- juice-plastic bottle-glass mug (x2)

For each setting, we randomly choose a starting location for the pouring and receiving containers, which are both distinct from the locations used in the main evaluation benchmark. We also randomly choose one of the 4 different pour commands (quarter, half, three-quarters, all). Demonstrations are collected via Apple Vision Pro teleoperation [47].

For evaluation, we choose 2 transparent container and 2 opaque container settings, namely:

1. water-white cup-plastic cup

2. juice-plastic bottle-plastic cup
3. water-white cup-red cup
4. water-metal thermos-metal mug

Similar to our main benchmark, we evaluate 3 random positions per setting, across 4 distinct pouring commands (quarter, half, three-quarters, all). Note that the evaluation setting is completely within distribution with respect to the original 10 pouring demonstrations.

B Generative Model Details

In this section, we provide additional details on the MMAudio video-to-audio generative model, including architectural modifications and finetuning design choices.

B.1 MMAudio Architecture

The MMAudio model follows a modular, multimodal architecture for conditional video-to-audio generation. Concretely, video, audio, and text streams are encoded separately via modality-specific encoders and fused in a series of multimodal transformer blocks to generate audio latents autoregressively. For further details on the original MMAudio design, please refer to Cheng et al. [26].

Segmentation Pathway. We introduce a new input modality during finetuning: semantic segmentation masks aligned with each RGB frame. For each mask of shape $H \times W \times C$ (where C is the number of semantic classes), we apply a lightweight convolutional encoder to extract spatial features:

- Two 3×3 convolution layers with ReLU activation,
- Global average pooling to obtain a per-frame embedding $s_t \in \mathbb{R}^{d_{\text{seg}}}$ (we use $d_{\text{seg}} = 512$),
- A projection layer to map s_t to the transformer embedding dimension (1024).

Injection Strategy. We inject the projected segmentation embedding s_t into the model as an additional global conditioning vector alongside the existing visual and text-based conditioning. Specifically, s_t is concatenated with the pooled visual and text tokens and added to the conditioning stream used by all multimodal transformer blocks. This avoids modifying the original CLIP encoder or the SyncFormer path, preserving the pretrained backbone while allowing the model to leverage segmentation information during finetuning.

This injection is implemented using a separate projection MLP and attention-based fusion mechanism, where the transformer dynamically integrates s_t alongside the timestep embedding and existing global context. By gating the contribution of s_t (via learned scalars), the model can modulate its reliance on segmentation during finetuning.

For ease of understanding this injection implementation, we provide the PyTorch pseudo-code below:

Code Block 1: Segmentation mask fusion module. A learnable gate modulates the contribution of segmentation embeddings relative to the original global conditioning vector.

```
class SegmentationFusion(nn.Module):
    def __init__(self, dim):
        super().__init__()
        self.gate = nn.Linear(dim * 2, 1)

    def forward(self, c_g, seg_embed):
        """
        c_g: (B, T, D) global conditioning vector, see Cheng et al.
        seg_embed: (B, T, D) segmentation embedding
        returns: (B, T, D) fused conditioning vector
        """
        x = torch.cat([c_g, seg_embed], dim=-1)     # (B, T, 2D)
        alpha = torch.sigmoid(self.gate(x))         # (B, T, 1)
        return alpha * seg_embed + (1 - alpha) * c_g
```

B.2 Ablation: Zero-Shot vs. Finetuned

One natural hypothesis is that the pretrained MMAudio model is sufficient to generate high-quality pouring sounds out-of-the-box. To test this, we conducted a controlled ablation comparing the *zero-shot* performance of the pretrained model against the *finetuned* variant described in Section 3.2. Our goal was to measure the extent to which finetuning improves the fidelity, diversity, and downstream utility of the generated audio.

Finetuning Setup. Our finetuning dataset and input modalities are described in Section 3.2. Our finetuning architecture is described in Appendix B.1. We froze the CLIP vision encoder and updated only the following components:

- The multimodal transformer backbone (attention and projection layers),
- The audio decoder (diffusion UNet),
- The segmentation cross-attention module (newly added),
- The final linear projection layers for spectrogram generation.

Audio was represented as log-mel spectrograms using a 64-bin mel filterbank, 16 kHz sampling rate, 25ms window, and 10ms hop size. During training, the spectrogram was denoised from Gaussian noise over 100 diffusion steps. For optimization, we used AdamW with a learning rate of 1e-5, and a cosine decaying scheduler with 500-step warmup. The batch size was 32, and the losses involved the default MMAudio losses (L1 waveform loss + spectral convergence loss + perceptual loss using AudioCLIP embeddings). The diffusion sampler was DDIM with 50 diffusion timesteps. We finetuned for a total of 100K steps (~40 epochs).

Evaluation Setup. We use a subset of 100 simulation trajectories as the basis for our ablation. For each video, we generated synthetic audio using both the zero-shot and finetuned models, conditioned on RGB only for the former, and RGB + segmentation masks for the latter.

Quality Metrics. We evaluated generated audio across three axes:

- **Fidelity**: Measured via Signal-to-Distortion Ratio (SDR), computed between the generated waveform and ground-truth microphone audio. Higher is better.
- **Spectral Accuracy**: Assessed using Log-Spectral Distance (LSD) between generated and real spectrograms. Lower indicates better spectral alignment.
- **Diversity**: We compute the minimum LSD between each generated sample and the training set to assess the marginal spread of synthetic samples. This tests whether the model produces varied outputs or mode-collapses to common templates.

Utility Metrics. We further evaluated the practical utility of each model by training the same multimodal diffusion policy architecture on the generated trajectories from each variant. We then evaluated both policies in the real world on a 4-condition benchmark (2 opaque, 2 transparent pours) using the protocol described in Appendix A.2. Performance was measured using Normalized Mean Absolute Error (NMAE) of the poured volume.

Quantitative Results. The results are summarized below:

Metric	Zero-Shot	Finetuned	Relative Change
Signal-to-Distortion Ratio (SDR) ↑	7.8 dB	**11.3 dB**	+45%
Log-Spectral Distance (LSD) ↓	1.95	**1.43**	-26.7%
Audio Diversity (min-LSD to train set) ↑	0.67	**1.02**	+52.2%
Policy NMAE ↓	0.50	**0.37**	-26.0%

Qualitative Observations. Audio generated by the zero-shot model often exhibited unnatural characteristics—overly low-frequency hums, repetitive bubbling artifacts, or missing transient features such as pour onset and cutoff. These artifacts were substantially reduced in the finetuned model,

which produced audio with realistic variations in pitch, volume, and dynamics that aligned with the visual pouring cues. Spectrogram inspection revealed tighter alignment of temporal events and improved high-frequency detail in the finetuned outputs.

Conclusion. These results conclusively show that finetuning is essential for adapting pretrained generative audio models to domain-specific robotic tasks. Despite the general capabilities of MMAudio, its pretrained version lacks the task-specific inductive biases and visual grounding needed for physically plausible pouring sounds. Finetuning on a small, curated dataset enables the model to capture subtle acoustic dynamics critical for multimodal policy learning and sim-to-real transfer.

C Additional Training Details

This section contains additional training details to supplement Section 4.2. Our RGB observations are $224 \times 224 \times 3$ frames, with a frame stack history of 2 timesteps. Our audio observations are 533 samples per timestep, with a horizon of 120 timesteps. Finally, our proprioception observations are 7-DoF joint angles, with a history of 2 timesteps.

Our diffusion policy predicts a 10-dim action vector (6D rotation representation) with a horizon of 16 timesteps. We use a DDIM sampler with 50 diffusion timesteps. Our scheduler is the Glide cosine scheduler [48], with beta start and end at 0.0001 and 0.02, respectively. We train with a batch size of 64, using the AdamW optimizer and a learning rate of 0.0001. Learning rate is cosine decayed after 500 warmup steps, with a total of 500 training epochs (250 steps per epoch). We incorporate weight decay of 1e-6 on all trainable parameters.