ManiWAV: Learning Robot Manipulation from In-the-Wild Audio-Visual Data

Zeyi Liu[1] Cheng Chi[1,2] Eric Cousineau[3] Naveen Kuppuswamy[3]
Benjamin Burchfiel[3] Shuran Song[1,2]

[1]*Stanford University* [2]*Columbia University* [3]*Toyota Research Institute*

Abstract: Audio signals provide rich information for the robot interaction and object properties through contact. This information can surprisingly ease the learning of contact-rich robot manipulation skills, especially when the visual information alone is ambiguous or incomplete. However, the usage of audio data in robot manipulation has been constrained to teleoperated demonstrations collected by either attaching a microphone to the robot or object, which significantly limits its usage in robot learning pipelines. In this work, we introduce ManiWAV: an 'ear-in-hand' data collection device to collect in-the-wild human demonstrations with synchronous audio and visual feedback, and a corresponding policy interface to learn robot manipulation policy directly from the demonstrations. We demonstrate our system's capabilities through three contact-rich manipulation tasks that require either passively sensing the contact events and modes, or actively sensing the object surface materials. In addition, we show that our system can generalize to unseen in-the-wild environments, by learning from diverse in-the-wild human demonstrations. Project website: https://maniwav.github.io.

1 Introduction

Selecting and executing good contact is at the core of robot manipulation. However, most vision-based robotic systems nowadays are limited in their capability to sense and utilize contact information. In this work, we propose a robotic system that learns contact through a common yet under-explored modality – audio. Our first insight is that audio signals provide **rich** contact information. During a manipulation task, audio feedback can reveal several key information about the interaction and object properties, including:

- *Contact events and modes*: From wiping on a surface to flipping an object with spatula, audio feedback captures salient and distinct signals that can be used for detecting contact events and characterizing contact modes.
- *Surface materials*: Audio signals can be used to characterize the surface material through contact with the object. In contrast, either image sensors or vision-based tactile sensors require high spatial resolution to capture the subtle texture difference (e.g. the 'hook' and 'loop' side of velcro tapes).
- *Object states and properties*: With indirect contacts, audio signals can provide complementary information about the object state and physical properties beyond visual observation.

Second, audio data is **scalable** for data collection and policy learning. This is because acoustic sensors (i.e. contact microphones) are cheap, robust, and readily available to purchase. Audio signals also have standardized coding formats that can be easily integrated into existing video recording and storage pipelines (e.g. MP4 files). These nice properties make it possible to collect audio in the wild with low-cost data collection devices, such as a hand-held gripper, without the need for a robot. On the other hand, alternative ways to sense contact, such as tactile sensors, are relatively more expensive, fragile, and require expert knowledge to use.

Given the richness and scalability of audio data, we propose a versatile robot learning system, **ManiWAV**, that leverages audio feedback for contact-rich robot manipulation tasks. Building upon the portable hand-held data collection device UMI [1], we redesign one gripper finger to embed a *piezoelectric contact microphone*, that senses audio vibrations through contact with solid objects. The audio signals can be easily streamed to the GoPro camera through a mic port and stored synchronously with vision data in MP4 files. To learn from the collected demonstrations, one key challenge is to bridge the audio domain gap between in-the-wild data and actual robot deployment due to test-time noises (Fig. 1 b). To achieve this goal, we propose a

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

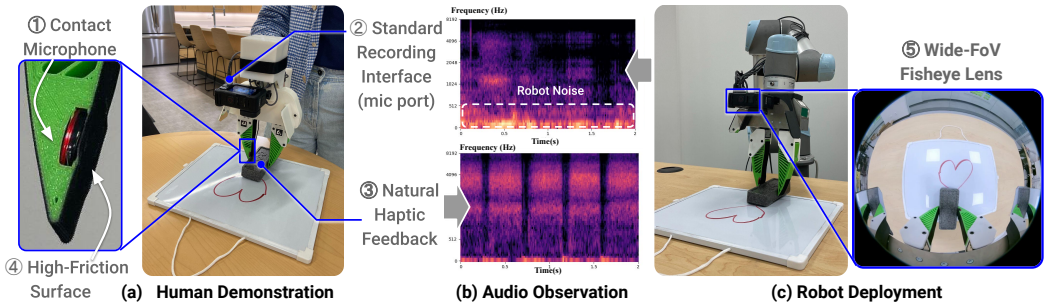


Fig 1: **Ear-in-hand gripper for in-the-wild data collection**. (a) The handheld design naturally provides *haptic feedback* to the demonstrator during contact-rich tasks (e.g., wiping), which is otherwise hard to obtain via teleoperation. Contact microphone captures high-frequency audio feedback that is recorded simultaneously with images. High friction tape is applied on top to augment the signals. (b) shows the domain gap between training and deployment data. (c) shows policy learned from in-the-wild data directly deployed on the robot.

data augmentation strategy that encourages learning of task-relevant audio representation. In addition, we propose an end-to-end sensorimotor learning network to encode and fuse the vision and audio data, with a diffusion policy [2] head for action prediction.

We demonstrate the capability of our proposed system on three contact-rich manipulation tasks: wipe shape from whiteboard, flip bagel with spatula, and strap wires with velcro tape. We also show that our system generalizes to unseen in-the-wild environments by leveraging in-the-wild data collected from diverse environments.

2 Method

Ear-in-Hand Hardware Design. Our data collection device is built on top of Universal Manipulation Interface (UMI) [1]. We redesign the 3D printed parallel jaw gripper on the device to embed a piezoelectric contact microphone under high-friction grip tape wrapped around the finger. The microphone is connected to the 3.5mm external mic port on the GoPro camera media mod. Fig. 1 (a) shows the hand-held gripper design. Audio is recorded at 48000 Hz and stored with 60Hz image data synchronously as MP4 files. During robot deployment, the same parallel jaw gripper with embedded microphone is mounted on a UR5 robot arm, shown in Fig. 1 (c).

We propose an end-to-end closed-loop sensorimotor learning model that takes in RGB images and audio, and output 10-DoF robot actions (end effector positions, end effector orientation represented in 6D [3], and 1D gripper openness).

Audio Data Augmentation. One key challenge is that the audio signals received during real time robot deployment is very different from the data collected by the hand-held gripper, resulting in a large domain gap between our training and test scenarios, as illustrated in Fig. 1 (b). This is mostly because 1) nonlinear robot motor noises during deployment, 2) out-of-distribution sounds generated by the robot interaction. (e.g. accidentally collide with an object).

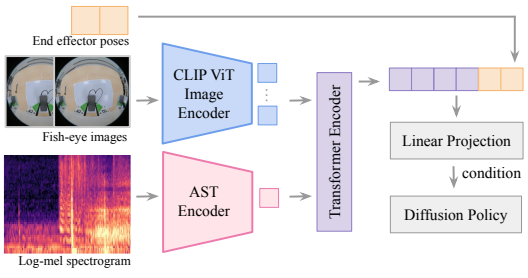


Fig 2: **Network Architecture.**

To address the domain gap, the key is to augment the training data with noises and guide the model to focus on the invariant task-relevant signals and ignore unpredictable noises. In particular, we randomly sample audio as background noises from ESC-50 [4]. The sounds are normalized to the same scale as the collected sound in the training dataset. We also record 10 samples of robot motor noises under randomly sampled trajectories with same contact microphone location as deployment time. The background noises and robot noises are overlayed to the original audio signal, each with a probability of 0.5.

We use a CLIP-pretrained ViT-B/16 model [5] to encode the RGB images. The images are resized into 224x224 resolution with random crop and color jitter augmentation. We use the audio spectrogram transformer (AST) [6] to encode the audio input. The audio singal is converted to a log mel spectrogram using FFT

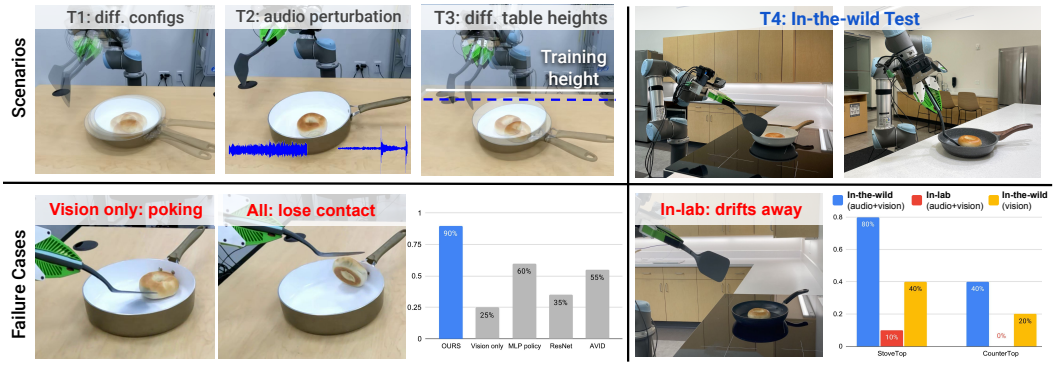


Fig 4: **Flipping Evaluation.** Up: On the left, we show the in-lab test scenarios. We train the policy with in-lab demonstrations collected in the same environment as inference time. On the right, we show the two unseen environments for the in-the-wild generalization test. Bottom: Typical failure cases and task success rate.

size and window length of 400, hop length of 160, and 64 mel filterbanks. The log-mel spectrograms are linearly normalized to range $[-1,1]$. We fuse the vision and audio features using a transformer encoder in a similar fashion as Li et al. [7] to leverage the attention mechanism to weigh the features adaptively at different stages of the task. We concatenate the output features and downsample the dimension to 768 with a linear projection layer. Finally we concatenate the end effector poses (20 Hz) from the past 2 timesteps to the audio-visual feature. To model the multimodality intrinsic to human demonstrations, we choose to use a diffusion model with UNet encoders as proposed by Chi et al. [2] as the policy head, conditioned on the observation representation mentioned above in each denoising step. The entire model (Fig. 2), including the above mentioned encoders, is end-to-end trained using the noise prediction MSE loss on future robot trajectories of 16 steps.

3 Evaluation

Flipping Task. The robot is tasked to flip a bagel in a pan from facing down to facing upward using a spatula. We collected two types of demonstrations for this task: 115 in-lab demonstrations and an additional 274 in-the-wild demonstrations collected in 6 different environments using different pans. Details of test scenarios and baselines can be found in the Appendix. The quantitative result and typical failure cases are visualized in Fig. 4. Our key findings are: 1) Transformer audio encoder outperforms a CNN-based encoder. We find that training a transformer encoder from scratch yields better performance compared to using a CNN-based encoder, as shown in the [ResNet] and [AVID] baseline. This is likely because the self-attention mechanism in transformer allows the model to focus more on the frequency regions of task-relevant signals in the spectrogram. 2) In-the-wild data enables generalization to unseen in-the-wild environments. As shown in Fig. 4, a policy trained on in-the-wild data significantly outperforms a policy trained on in-lab data in two unseen environments, as the scene diversity in in-the-wild data allows the policy to generalize better to new environments.

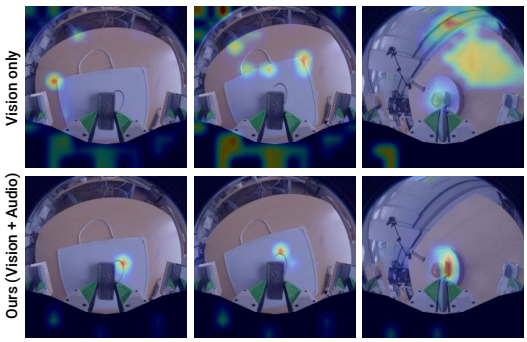


Fig 3: **Attention Visualization.** We find that a policy co-trained with audio attends more on the task-relevant regions (shape of drawing or free space inside the pan). In contrast, the vision only policy often overfits to background structures (e.g., edge of the table) to infer contacts.

Wiping Task. In this task, the robot is tasked to wipe a shape (e.g. heart, square) drawn on a whiteboard. Details of test scenarios and baselines can be found in the Appendix. We collect 120 demonstrations. The quantitative result and typical failure cases are visualized in Fig. 5. We find that: 1) Contact audio improves robustness and generalizability, when the [Vision only] policy fails to generalize to unseen table heights and unseen erasers. We visualize the attention map of the vision encoder in Fig. 3 and find that the model trained with audio attends better to task-relevant features. 2) Noise augmentation is an effective strategy to bridge

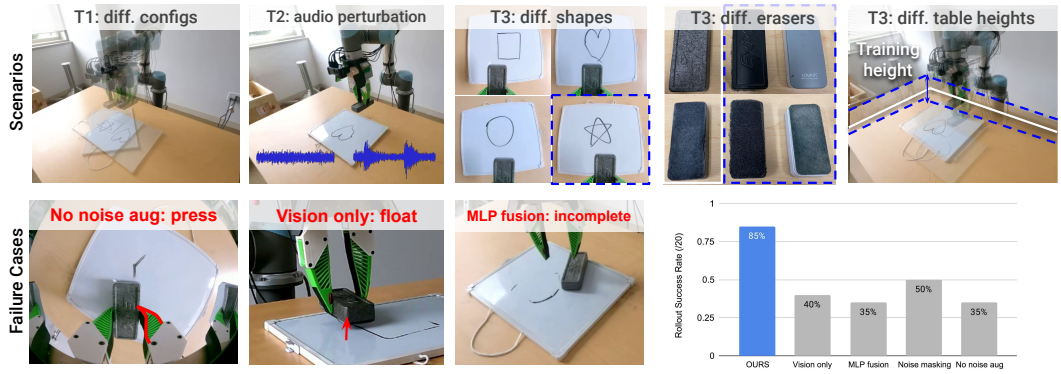


Fig 5: **Wiping Evaluation.** Up: Different test scenarios. Bottom: Typical failure cases and task success rate. [Vision only] policy often fails to maintain proper contact (e.g., either press too hard into the broad or float). [MLP fusion] policy often fails to fully wipe out the drawing and terminates early.

Fig 6: **Taping Evaluation.** In the first row, we show the task definition. The second row shows the typical failure case for each method and the overall task success rate.

the audio domain gap and increase the system's robustness to out-of-distribution sounds. Without noise augmentation, the robot is less robust to noises during test time and does not generalize well to unseen table heights, unseen erasers, and unseen shape. Lastly, we show 3) the advantage of using a transformer to fuse the vision and audio features compared to using an MLP. A typical failure in [MLP fusion] is that the robot fails to wipe in the area of the shape and stops when the shape is not completely wiped off.

Taping Task. The robot is tasked to choose the 'hook' tape from several tapes (either 'hook' or 'loop') and strap wires by attaching the 'hook' tape to a 'loop' tape underneath the wires. We collect 200 demonstrations in total, with a 'sliding' primitive where we use the tip of the gripper finger to slide along the tape. We find that: 1) Contact microphone is sufficiently sensitive to different surface materials. As shown in Fig. 6, the vision only policy makes random decisions and yields a 20% success rate. Similarly, the system that uses an environment microphone achieves similar results as the [Vision only] method, as it fails to pick up the subtle differences between the surface material. In contrast, by leveraging the contact microphone, our method is able to reliably guide the robot to pick up the correct tape. 2) Training-time noise augmentation is more effective than test-time noise reduction. This is because the noise cancellation algorithm causes the signal to be deprecated, resulting in domain gap between training and testing. On the other hand, our noise augmentation method preserves the frequency distribution of the original signals.

3.1 Limitations and Future Directions

Even though the contact microphone can pick up a wide range of audio signals and is robust against environment noises in the background by design, it may not be useful in scenarios where the interaction does not generate salient signals (e.g. for deformable objects such as cloth or quasi-static tasks), and can easily become imperceptible due to robot motor noises during deployment. On the policy learning front, the current policy does not leverage the fact that audio signals are received at a higher frequency than images and can be used to learn more reactive behaviors. Future work can consider a hierarchical network architecture [8] that infers higher frequency actions from audio inputs.

References

[1] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[2] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[3] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[4] K. J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press. ISBN 978-1-4503-3459-4. doi:10.1145/2733373.2806390. URL http://dl.acm.org/citation.cfm?doid=2733373.2806390.

[5] 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.

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

[7] 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.

[8] S. Saxena, M. Sharma, and O. Kroemer. Mrest: Multi-resolution sensing for real-time control with vision-language models. *arXiv preprint arXiv:2401.14502*, 2024.

[9] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev, and J. Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *Advances in neural information processing systems*, 31, 2018.

[10] J. Mejia, V. Dean, T. Hellebrekers, and A. Gupta. Hearing touch: Audio-visual pretraining for contact-rich manipulation. *arXiv preprint arXiv:2405.08576*, 2024.

[11] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017. doi:10.1109/ICASSP.2017.7952261.

[12] P. Morgado, N. Vasconcelos, and I. Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12486, 2021.

[13] 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.

[14] T. Sainburg. timsainb/noisereduce: v1.0, June 2019. URL https://doi.org/10.5281/zenodo.3243139.

Appendix

A Method Details

A.1 Audio Latency Calibration

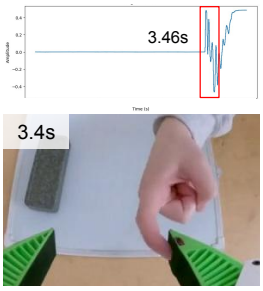


Fig 7

Similar to using a clapperboard in film production, we tap on the contact microphone and match the time between the frame when the finger is observed to be in contact with the microphone and the corresponding audio signal captured by the contact microphone. A visual illustration is shown in Fig. 7. The audio is received about 0.06s after the image is received. As a result, the total audio latency is 0.17 + 0.06 = 0.23s, where 0.17s is the calibrated image latency following the approach in [1].

A.2 Model Details

Image Augmentation. Each image is randomly cropped with a 95% ratio and then resized to its original resolution in the replay buffer. To make the learned model more robust to different lighting conditions at test time, we apply ColorJitter augmentation with brightness 0.3, contrast 0.4, saturation 0.5, and hue 0.08.

Spectrogram Parameters. There are several parameters when converting audio waveform to spectrogram that controls the time and frequency resolution of the resulting spectrogram image. Window length is the length of the fixed intervals in which STFT divides the signal; it controls the time and frequency resolution of the spectrogram. Hop length is the length of the non-intersecting portion of window length. The smaller the hop length is, the more times a particular audio segment will present in STFT, and the more elongated the time-axis of the resulting spectrogram will be. For a sample rate of 16kHz, we use the default window length, 400, as recommended by torchaudio. It is recommended that the hop length is around ½ of the window length; we choose 160 to slightly augment the signal pattern along the time-axis since contact signals are usually sparser in time compared to nature sounds or speech.

Transformer Encoder. We use one transformer encoder layer with 8 heads to fuse the vision and audio features. We set feedforward dimension to 2048 and dropout ratio to 0.0.

End-to-End Training Details. For each task, the entire model is end-to-end trained on 2 NVIDIA GeForce RTX 3090 GPUs for 60 epochs, with a batch size of 64. We use the AdamW optimizer with lr=1e-4, betas=[0.95, 0.999], eps=1.0e-8, weight_decay=1.0e-6, and apply EMA (Exponential Moving Average) on the weights.

B Evaluation Details

B.1 Test Scenarios and Baselines

B.1.1 Flipping

Test Scenarios: We run 20 rollouts for each policy. To ensure fair comparison, we use the same set of robot and object configurations for evaluations between different methods. We achieve the same object configuration by overlaying their position with respect to a captured image in the camera view. The test configurations can be grouped into four categories.

- T1: Variations in task configuration: different initial robot and object configurations (14 / 40).
- T2: Audio perturbation by playing different types of noises in the background (2 / 40).
- T3: Generalization to unseen table height (4 / 40).
- T4: Generalization to two unseen in-the-wild environments: a black stovetop and a white countertop, the later is more challenging due to unstructured background and lack of similar training data (20 / 40).

Comparisons: In this task, we focus on comparing our results with several ablations of the network design:

- Vision only: the original diffusion policy conditioned on image observations.

- MLP policy: using a MLP with three hidden layers (following Li et al. [7]) instead of action diffusion. The model takes the observation representation and outputs the future action trajectory.
- ResNet: uses a ResNet18 encoder to encode the audio log-mel spectrograms, with an additional CoordConv layer [9] following Li et al. [7].
- AVID: Following the approach by Mejia et al. [10], use a 9-layer CNN audio encoder pre-trained on AudioSet [11] using Audio-Visual Instance Discrimination (AVID) [12].
- In-the-wild: For in-the-wild evaluation (T4), we compare our model trained with in-the-wild demonstrations (blue bar in Fig. 4 bottom right chart), only in-lab demonstrations (red bar in Fig. 4 bottom right chart), and a [Vision only] baseline trained on in-the-wild data (yellow bar in Fig. 4 bottom right chart). Details on the in-the-wild dataset can be found in appendix.

B.1.2 Wiping

Comparisons: In addition to the vision only baseline, we evaluate the following alternatives for processing and learning from audio data:

- MLP fusion: uses an MLP with 2 hidden layers to fuse the vision and audio features instead of a transformer encoder. This approach was used by Du et al. [13].
- Noise masking: without noise augmentation but instead mask out the audio frequency below 500 Hz, which is the UR5 control frequency.
- No noise aug: without augmenting the audio data with noises during training.

Test Scenarios: We run 20 rollouts for each policy. In addition to the T1 (5 / 20) and T2 (4 / 20) test cases as described above, we also test generalization to unseen table heights, erasers and drawing shape T3 (11 / 20). A detailed breakdown of the test scenarios can be found in appendix.

B.1.3 Taping

Comparisons: In addition to vision only, we compare the following baselines:

- Env Mic: Instead of using a contact microphone, we mount a Rode VideoMic GO II directional microphone on the GoPro camera for both data collection and deployment to collect the audio signals.
- Noise Reduction: Instead of applying training time noise augmentation, we evaluated an alternative method that uses a test-time noise reduction algorithm. The algorithm estimates a noise threshold for each frequency and applies a smoothed mask on the spectrogram [14]. More details can be found in appendix.

Test Scenarios: We run 10 rollouts in total, each time the robot is presented 2-4 velcro tapes in random order with at least one is a 'hook' tape.

B.2 In-the-Wild Data Collection

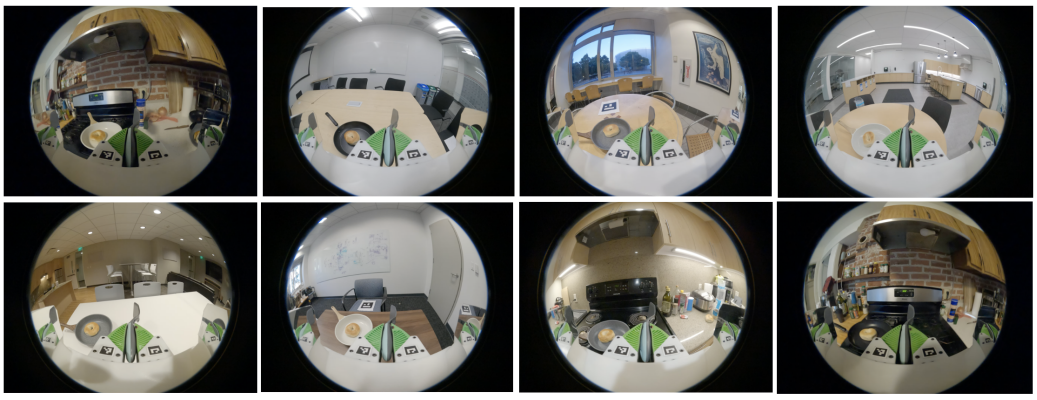


Fig 8: Example Scenes in the In-the-Wild Dataset.

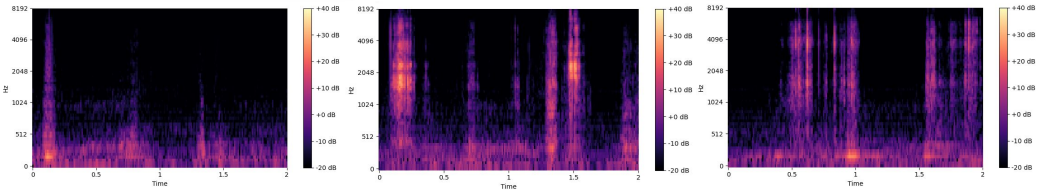


Fig 9: We visualize the audio spectrogram in the pouring task when the robot 'shakes' the cup to sense whether there are objects inside the cup. (a) shows the audio feedback of an empty cup, (b) and (c) shows the audio feedback for a cup with dice and with screws inside, respectively.

We collect 274 in the wild data for the bagel flipping task in total, including 52 in a conference room, 37 and 44 in two kitchens, 46 in an office, 76 on lounge tables, and 19 in a cafe, using 4 different pans. Examples of the environments are shown in Fig. 8.

B.3 Result Details

B.3.1 Pouring Task

A breakdown of the success rate for each substep in the pouring task is shown in Tab. 1. 'Grasp' is successful if the robot grasps the white cup stably in its end effector, 'Pour' is successful if the robot pours all objects in the white cup to the pink cup on the table, 'Place' is successful if the robot places the white cup on the table after pouring. By using audio feedback to infer the object state (whether there's object in the cup or not), our method is able to reliably guide the policy to pour objects and place the cup, whereas the baselines either never executes the pour action or the place action, resulting in low substep success rate. We also find that the policy can generalize to unseen objects such as screws. In Fig. 9, we visualize the audio spectrogram when the robot 'shakes' the cup. We can observe that the spectrogram for an empty cup is distinctive from a non-empty cup, and a cup with screws generates a similar audio pattern as a cup with dice upon shaking. We hypothesize that the audio features for the cup with screws and the cup with dice are also close in the audio feature space, leading to similar policy behavior. Videos of the policy rollouts can be found on the project website.

	Grasp	Pour	Place
OURS	90	**100**	**90**
Vision only	100	0	8.3
1s audio	91.7	30	0
10s audio	100	60	16.7

Table 1: Success Rate Breakdown.

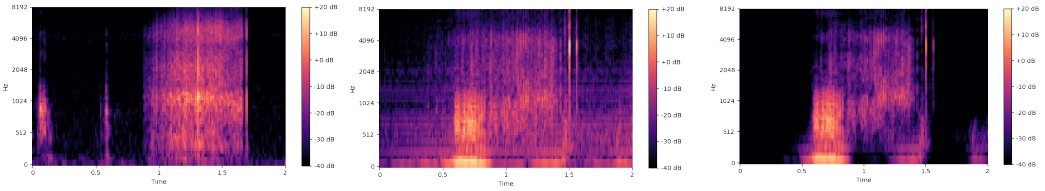
B.3.2 Taping Task

A breakdown of the success rate for each substep in the taping task is shown in Tab. 2. 'Touch' is successful if the robot slides along the tape while maintaining contact, 'Sense' is successful if the robot chooses the correct tape, 'Pick' is successful if the robot successfully grasps the tape. 'Place' is successful if the robot successfully places the tape on top of the wires. By leveraging audio feedback to infer the object surface material (whether the tape is a 'hook' or 'loop'), our method is able to reliably guide the policy to choose the correct tape whereas the baselines make random decisions, as shown in the 'Sense' step success rate. Videos of the policy rollouts can be found on the project website.

	Touch	Sense	Pick	Place
OURS	90	**90**	80	80
Vision only	80	30	90	80
Env Mic	80	30	90	40
Noise Reduction	80	40	90	90

Table 2: Success Rate Breakdown.

Noise Reduction Algorithm. We use the non-stationary noise reduction method introduced in [14]. The algorithm computes a spectrogram from the audio waveform, and then apply IIR filter forward and backward on each frequency channel to obtain a time-smoothed version of the spectrogram. A mask is then computed base on the spectrogram from estimating a noise threshold for each frequency band of the signal/noise. And finally, a smoothed, inverted version of the mask is applied on the original spectrogram to cancel noise. In our experiments, we apply this algorithm directly on the real time audio signals before feeding it to the model for inference. In Fig. 10, we show spectrogram visualization of the real time signal (b) and the signal after



(a) Training signal (b) Real time signal (c) Signal after noise reduction

Fig 10: Spectrogram Visualization.

reduction (c). Even though the noise reduction seems to successfully remove most of the robot noises and some other background noises, it does not preserve the original signal well, and still result in domain gap as comparing to the training signal (a).

B.3.3 Wiping Task

We listed out each test scenario in the wiping task evaluation and success (1) / failure (0) for each method.

	Test Scenario	OURS	Vision only	MLP fusion	Noise masking	No noise aug
0	T1 (square)	1	1	0 (incomplete)	0 (press)	1
1	T1 (heart)	1	1	0 (incomplete)	1	0 (press)
2	T1 (circle)	1	0 (incomplete)	1	1	1
3	T3 (star)	1	1	1	0 (press)	0 (press)
4	T3 (+1 inch)	1	0 (float)	0 (incomplete)	1	0 (incomplete)
5	T3 (+5 inch)	1	0 (press)	0 (incomplete)	0 (press)	0 (press)
6	T3 (changing height)	1	0 (float)	0 (float)	0 (press)	1
7	T3 (changing height)	1	0 (press)	0 (incomplete)	0 (press)	0 (press)
8	T2 (white noise)	1	1	0 (press)	1	1
9	T2 (white noise)	1	1	1	1	1
10	T2 (construction noise)	1	1	1	0 (press)	0 (press)
11	T2 (music)	0 (float)	1	0 (incomplete)	1	0 (incomplete)
12	T1 (board orientation)	0 (float)	0 (float)	0 (float)	0 (float)	0 (incomplete)
13	T1 (board position)	1	1	0 (incomplete)	1	1
14	T3 (unseen eraser)	1	0 (float)	0 (press)	0 (press)	0 (press)
15	T3 (unseen eraser)	1	0 (incomplete)	1	1	0 (press)
16	T3 (-1 inch)	1	0 (float)	1	1	0 (float)
17	T3 (-1 inch)	1	0 (float)	0 (incomplete)	0 (press)	0 (press)
18	T3 (-2 inch)	0 (float)	0 (float)	1	1	0 (float)
19	T3 (-2 inch)	1	0 (float)	0 (incomplete)	0 (press)	1
	Success rate	85%	40%	35%	50%	35%

Table 3: Wiping Test Scenario Breakdown.

'Float' means the robot keeps floating above the board without contacting the board before it wipes off the shape. 'Press' means the robot exerts too much force downward and causes the gripper to bend against the board. 'Incomplete' means the robot fails to follow the shape, either stops wiping early or wipes in wrong location.

As we can see from Tab. 3, 'Float' and 'Press' are most common in the [Vision only] baseline especially when the table height is different than training, likely due to the fact that it's insufficient to infer contact from the top-down view wrist-mount camera image alone (as shown in Fig. 11).

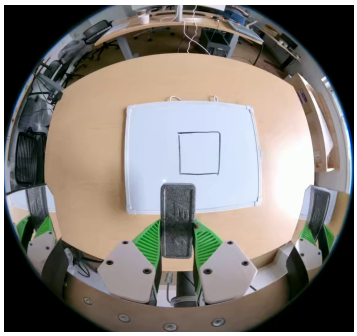


Fig 11: Camera view in the wiping task.

The most common failure case in [MLP fusion] is 'incomplete', where the robot stops wiping and releases the eraser before the shape is completely wiped off. We hypothesize that this is because simply fusing vision and audio features with MLP layers lose information that's crucial for inferring the stage of the task.

Without noise augmentation, the policy exhibits various unexpected behaviors including 'press', 'incomplete', and 'float', because of the big domain gap between training and testing. The policy achieves better performance by simply masking out the robot noise frequency range in the [Noise masking] baseline, however, it still fails half of the time and most of the failure cases are 'pressing' too hard against the board.

B.3.4 Flipping Task

We listed out each test scenario in the bagel flipping task evaluation and success (1) / failure (0) for each method.

	Test Scenario	OURS	Vision only	MLP policy	ResNet	AVID
0	T1	1	0 (lose)	1	0 (poke)	1
1	T1	1	0 (lose)	0 (displace)	1	0 (lose)
2	T1	1	0 (poke)	1	0 (displace)	0 (displace)
3	T1	1	0 (lose)	0 (poke)	1	1
4	T1	1	1	1	1	0 (lose)
5	T1	0 (lose)	0 (lose)	1	0 (displace)	1
6	T1	1	0 (lose)	1	1	1
7	T1	1	0 (poke)	0 (displace)	0 (poke)	1
8	T1	1	1	1	0 (displace)	0 (poke)
9	T1	1	1	0 (lose)	1	1
10	T1	1	1	1	0 (displace)	1
11	T1	1	0 (poke)	0 (poke)	0 (displace)	1
12	T1	1	0 (poke)	1	1	1
13	T1	1	0 (lose)	0 (lose)	0 (poke)	1
14	T2 (clap)	1	0 (poke)	1	1	0 (displace)
15	T2 (construction noise)	1	0 (poke)	1	0 (stuck)	1
16	T3 (unseen height)	1	1	1	1	0 (displace)
17	T3 (unseen height)	0 (lose)	0 (poke)	0 (lose)	0 (displace)	0 (lose)
18	T3 (unseen height)	1	0 (lose)	1	0 (poke)	0 (displace)
19	T3 (unseen height)	1	0 (displace)	0 (lose)	0 (displace)	0 (displace)
	Success rate	90%	25%	60%	35%	55%

Table 4: Flipping Test Scenario Breakdown.

'Poke' means that the robot pokes the spatula on the side of the bagel instead of inserting the spatula between the pan and the bottom of the bagel. 'Lose' means the robot loses contact with the bagel before it is flipped, as a result, the bagel falls back to its original side. 'Displace' means that the spatula is displaced in the robot end effector as comparing to its initial pose, as a result of the robot keeps moving down instead of switching to slide the spatula along the bottom of the pan.

For all T1 scenarios, we randomize the robot initial pose and object positions (e.g. bagel and pan). We can observe that the [Vision only] policy does not generalize well across initial configurations and most failure cases are either poking on the side of the bagel (since it's hard to infer from the image alone if the spatula is contacting the bottom of the pan), or losing contact with the bagel early before it can be flipped.

Using a [ResNet] and [AVID] audio encoder results in the spatula to 'displace' most of the times, likely because the model is not sensitive enough to the sound feedback of spatula touching the bottom of the pan, and as a result keeps moving downward and causes the spatula to displace.