Audio-SmolVLA

Shivam Garg Adam Malyshev
University of Maryland, College Park
{ sgarg23, amaly944 }@umd.edu

Abstract

We investigate how small multimodal robot models can be extended with audio understanding, a capability missing from current efficient vision-language-action (VLA) systems. Our approach introduces an audio projection layer and compares multiple training strategies for integrating this new modality into smolVLA. Through audio-responsive tasks, we evaluate whether audio can be incorporated without degrading the model's pretrained visuomotor-language capabilities while improving performance in settings where audio information is useful.

1. Introduction

Robotics aims to emulate human dexterity and perception to automate work in unpredictable settings [38]. While traditional analytical control provides safety, it lacks the semantic understanding required for unstructured environments. This has driven the field toward data-driven approaches, including Reinforcement Learning (RL) and Imitation Learning (IL). However, these methods often face challenges regarding sample efficiency and generalization [12, 26].

The rise of foundation models has catalyzed the development of Vision-Language-Action (VLA) models, which combine the reasoning of VLMs with robotic control. Among monolithic approaches, continuous control methods—dominated by diffusion [47] and flow-matching [18]—have outperformed discrete token methods in quality, albeit often at a higher computational cost. Recent open-source initiatives, such as smolVLA [46], utilize efficient flow-matching to bridge this gap, enabling high-performance policies on accessible hardware.

A significant limitation of current VLAs is their reliance on vision, ignoring audio, a modality crucial for sensing beyond the line of sight (e.g., alarms, occluded events) and enriching human-robot interaction. Integrating audio into these generalized models via end-to-end retraining is computationally expensive. Therefore, efficient fusion methods that project modalities into frozen backbones [19, 48] are essential.

Our work addresses this gap by fusing audio perception into smolVLA. We employ a projection architecture to map audio embeddings into the VLA's language space, fine-tuning the model to react to audio stimuli while preserving its original manipulation capabilities. By utilizing the efficiency of smolVLA, we demonstrate inference on consumer hardware and extended observation histories on the SO101-Arm platform [20].

2. Related Works

2.1. VLAs

In this related works section we directly discuss VLA models, with FuSe and VLAS directly integrating an audio modality.

1. FuSe [19] is the most related work that we would like to build off of. In this work they describe a generally applicable recipe for fusing of additional sensory modalities apart from vision, including audio, that lack large datasets by using natural language as a common embedding space and then finetuning VLA models on this multimodal sensory data while preserving pre-trained knowledge. Jones et al. propose a loss function that sums action MSE, modality contrastive loss, and a generative loss to effectively join the semantic knowledge of the backbone with unseen sensor modalities. The generative loss is computed from annotated language instructions and the output of the decoder of the VLM backbone (or an added generative head single transformer if absent). This study uses the Octo [39] VLA and a custom pretrained VLA using PaliGemma [3] VLM as a backbone demonstrating the process can work with any model.
2. VLAS [54] is another study that incorporates the audio modality within a VLA model. Zhao et al. approach is to focus solely on the speech modality for robot manipulation instructions and uses an end-to-end approach. They build off the LLaVa [30] VLM directly, project the embeddings of a speech encoder into the language embedding space, then finetune the entire model to produce discrete action tokens. The study also incorporates a Voice RAG system in order to enable the storage of

personal knowledge about various speakers and thus enable the performance of tailored manipulation tasks to the speaker.

3. smolVLA [46] is a VLA model designed around maximizing efficiency. It is a small parameter count model ($\approx$ 500 M) using a small flow matching transformer as an action expert on top of a smolVLM2 [37] backbone, a VLM also designed with efficiency in mind. Shukor et al. achieve this via careful design considerations such as skipping half of the VLM layers, constraining visual tokens, and interleaving self-attention blocks with cross-attention blocks in the action expert on the hidden states of the VLM. Additionally the easily accessible S0100 Arm [20] platform was developed in order to collect a diverse high quality community contributed dataset for pretraining. Shukor et al. also introduce an asynchronous inference stack that allows for fast, reduced latency, and resource-efficient inference as well as the ability to separate action execution from the more resource intensive observation processing and action prediction computation. The result of the work is a VLA model that achieves performance similar to models $10\times$ the size that can be finetuned on a single GPU, deployed on consumer-grade GPUs or even just CPUs.
4. $\pi_{0.5}$ [18] is one of the more performant VLA models in recent time. Intelligence et al. focus on creating the most generalizable robot manipulation via a massive high-quality varied dataset and a co-training technique in a discrete manner during pretraining for multiple high-level semantically relevant subtasks. This first co-training stage pretrains the VLM model to output discrete FAST [41] action tokens, provide low-level language descriptions of subtasks, and bounding boxes on a large multimodal web and robotic dataset. This pretrains fine-grained semantic understanding of broad manipulation tasks and environments. The second stage trains a flow matching action expert on the discretized action text tokens to create continuous actions that are more suited for real-time inference along with verbal instructions from people. $\pi_{0.5}$ uses a large PaliGemma [3] VLM model to achieve this and use high-end expensive manipulator platforms. The verbal instructions are fed as transcriptions to the model during in-the-wild action expert training to improve the model's ability to create high-level subtask outputs for a trained policy.

These works directly cover novel VLA models and current state of audio integration into VLA models.

2.2. Efficient Fusion

These works focus on fusing together modalities efficiently by leveraging pretrained capabilities rather than unified architectures.

The first approach is a linear projection fully autoregressive approach where visual tokens are derived from projecting visual embeddings to visual tokens in the language embedding space, directly passed to the model alongside text tokens. Introduced by LLaVa [30] where CLIP [42] embeddings are linearly projected into language embedding space of a Vicuna [8] language decoder then carefully finetuned on a synthetically generated high-quality visual instruction dataset. This approach has been extended to the audio domain with SLAM-ASR [31] an LLM based ASR model, using a HuBERT X-Large [16] self-supervised speech representation model with Vicuna-7B [8] as an LLM backbone, demonstrating emerging ASR performance from an LLM model and crucially show that unfreezing the speech encoder during training hurts performance. This seems to be the most common approach for fusion due to its high performance and lightweight training used most notably in recent work such as Mellow [9] which achieves state-of-the-art performance on audio understanding using HTSAT [6] a small transformer based audio encoder and a simple projection layer to smolLM2 [2] language embedding space. This is also notably the fusion strategy used in the FuSe [19] recipe. However this approach's limitation is that when no pooling strategy is applied it can produce a large quantity of visual tokens that can overwhelm context limits [22].

The second approach is an attention strategy. Notably used by the Flamingo [1] VLM model using interleaved gated cross-attention dense layers with frozen language model blocks. The layers attend to vision embeddings for a variable amount of image or video features derived from a frozen vision encoder, which are mapped to a set of fixed visual tokens using a Perceiver Resampler to reduce computational complexity. This architecture was repeated for the Idefics [21] model.

A third approach is the Querying Transformer (Q-Former) a self-attention autoregressive approach, introduced as part of the BLIP 2 [25] model. The Q-Former uses a set number of learnable query embeddings that interacts with an image transformer which cross-attends to a frozen image encoder and self attends to a language transformer part. This approach significantly reduces parameter count compared to the gated cross-attention layers thus making it more efficient. This technique was then applied to the audio modality in SALMONN [48] adapting the Q-Former to have a window-level query.

Via the study done for Idefics2 [23] VLM , the authors showed a fully autoregressive approach with LoRA [17] finetuning on the LM backbone can outperform the more complex and parameter heavy cross-attention approach, but when the LM is frozen then the autoregressive approach performs worse [22].

Notably in the Video-LLaMa series [53] [7] [52] first a Q-Former was used to aggregate image and audio information then projected, but Video-LLaMa2 and Video-LLaMa3

removed the need for a Q-Former in favor of simple MLP projector.

These works cover fusion strategies that boot strap pretrained models for training and parameter efficient emergent multi-modal understanding and reasoning.

2.3. Efficient Multimodal Models

These works introduce models focused on parameter efficient multi-modal reasoning models and introduce various techniques to maximize performance and achieve comparable performance with much larger models. Core to efficiency is leveraging pretrained encoders as solutions without them cause degradation in performance and require longer training [22]. Largely efficiency is achieved by leveraging the efficient strategies covered in the previous section, selecting strategies carefully in terms of performance and computational efficiency tradeoffs. PaliGemma [3] uses a linear projection of the SiGLIP [49] vision encoder, but for vision modality this can create a large number of visual tokens that can overwhelm context and degrade performance. Thus for smaller models capped visual tokens seem to not hinder performance while improving efficiency, done with a learned transformer pooling such as the Perciever Resampler [1] or Q-Former [25] [23]. However in the audio realm learned pooling does not seem to be necessary [31] [9] [13] and projection layers seem to do fine. The visual domain also requires token compression strategies, notably pixel shuffle which benefit small VLMS when applied aggresively [37]. Almost all small efficient models benefit greatly from using high quality data and splitting training into multiple stages, especially to achieve performance on complex tasks such as Q&A, increasing in quality and in the portion of the model that is unfrozen, [22]. For training stage some works emphasize an initial stage to align the projector or adapter module [27] [25] [13], while other works directly start with next-token prediction tasks [9] [31].

2.4. Audio Models

These works specifically cover the fusion of audio models into large language model backbones.

A primary component of our work is selecting an audio encoder that provides strong semantic representations while remaining computationally lightweight. CLAP [10] learns audio representations aligned with natural language captions, producing semantically meaningful embeddings. HTS-AT, used in Mellow [9], is a lightweight hierarchical audio transformer that achieves strong performance on AudioSet while keeping model size and training time low. We intend to explore and evaluate the efficiency and performance of each of these different audio encoding strategies, along with ResNet-style encoders as they are used to adapt audio into smolVLA.

Mapping audio embeddings into the LLM space is typically done with a small projector network. Salmonn [48] demonstrates a general-purpose audio module for frozen LLMs, while Mellow [9] shows that partial unfreezing of the language model is often necessary to stabilize reasoning. Because of this, we will explore different finetuning strategies on smolVLA, from only LoRA on attention weights to fully unfreezing the smolVLM2 backbone.

Larger systems such as Qwen3-Omni [51] and Audio Flamingo 3 [13] show token interleaving, temporal position encoding, and staged training impact performance for multimodal reasoning. While much bigger than smolVLA, they show important aspects of efficient fusion and training without introducing complexity that could disrupt smolVLA's pretrained capabilities.

3. Approach

The primary challenge in developing generalist audio-visual robot policies is the scarcity of datasets that pair robotic actions with heterogeneous sensory data like audio [19]. While end-to-end training is effective for vision, learning audio representations solely from limited robotic demonstrations often leads to poor generalization or catastrophic forgetting of pre-trained capabilities.

To address this, we introduce a two-stage training strategy that explicitly decouples the learning of audio semantics from the learning of robotic control. Building upon the efficient modality fusion techniques demonstrated in FuSe [19] and the lightweight audio-text alignment of Mellow [9], our approach leverages natural language as a bridging modality.

Our method proceeds in two distinct phases. In Stage 1, we semantically align a frozen audio encoder with the pretrained language embedding space of the VLM backbone using large-scale, non-robotic audio datasets. This ensures the model can construct "comprehendible" audio representations in the VLA's native text embedding space, before any robotic control is attempted. Once the backbone is capable of "hearing," we freeze the alignment components and move on to Stage 2: training the action expert on carefully annotated audio-visual-motor data. This stage focuses exclusively on grounding these new audio signals into physical actions, allowing the policy to learn correlations between sound and control without simultaneously struggling to learn feature extraction.

By leveraging the flow matching-based SmolVLA [46] architecture, which utilizes the efficient SmolVLM2-500M [37] backbone. We also investigate whether this efficiency gain would allow an audio observation history greater than 0.4s as compared to the FuSE [19] study which used the diffusion-based Octo [39] model, as well as potentially higher action sampling frequency than 5 Hz.

3.1. Architecture

We design our architecture to extend the efficient, multimodal capabilities of SmolVLA into the audio domain while adhering to strict computational constraints suitable for low-cost robotic platforms. Our system consists of three modular components: an efficient audio perception module, an adapted vision-language backbone, and a flow-matching action expert.

3.1.1. Audio Perception Module

To process raw audio signals efficiently, we employ DyMN (Dynamic MobileNet) [45], a lightweight audio encoder pretrained on AudioSet [11]. We select the 10.57M parameter DyMN encoder over larger, transformer-based alternatives (e.g., 31M parameter HTS-AT [6]: the selected choice of audio encoder in Mellow [9]) because we hypothesized that a smaller parameter count is crucial to maintain a balance of encoder-LM allocation, extending the insights from the visual modality found in the smolVLA [46] study. The encoder outputs a 960-dimensional embedding vector representing the semantic content of the audio clip.

To integrate these audio representations into the discrete token space of SmolLM2 [2], the language backbone of SmolVLM2 [37], we introduce a trainable Audio Projector, following both FuSe [19] and Mellow's [9] architectures. This module consists of a Multi-Layer Perceptron (MLP) with a Linear $\rightarrow$ GeLU $\rightarrow$ Linear structure. The projector effectively functions as a "translator" that converts sound into soft prompt tokens compatible with the SmolVLM2's [37] attention mechanisms.

3.1.2. Adapted Vision-Language Backbone

We utilize SmolVLM2-500M [37] as our core reasoning backbone. This model combines a SigLIP [49] vision encoder with the SmolLM2 [2] language model, optimized for efficiency through strategies like aggressive visual token compression (pixel shuffle).

To enable audio understanding without disrupting the delicate pre-trained distributions of the text and vision components, we employ Low-Rank Adaptation (LoRA) [17]. Rather than unfreezing the entire language model, which risks catastrophic forgetting, we inject trainable low-rank matrices into the query and value projection layers of the self-attention blocks. This aligns with the findings in Mellow [9], which show that partial unfreezing of the language model stabilizes reasoning and accelerates model convergence during training.

3.1.3. Flow-Matching Action Expert

For robotic control, SmolVLA [46] appends a Flow-Matching Action Expert to SmolVLM2 [37]. This expert is a transformer-based decoder that interleaves cross-attention layers (attending to SmolVLM's [37] hidden states) with causal self-attention layers. It is trained to predict "chunks" of continuous joint-position actions (N=50) by denoising random noise conditioned on the multimodal context provided by SmolVLM [37]. This architecture allows the policy to synthesize high-frequency control signals from the semantic understanding produced by SmolVLM2 [37].

3.2. Training Methodology

Our training pipeline is designed to sequentially build capabilities: first establishing semantic perception, then grounding that perception in physical control. We employ a two-stage process that leverages distinct datasets and objective functions tailored to each modality.

3.2.1. Stage 1: Semantic Alignment (Audio-Text)

The objective of the first stage is to bridge the modality gap between the frozen audio encoder (E_a) and the pretrained language model (SmolLM2 [2]). We train the Audio Projector (P) and SmolLM2's LoRA adapters (θ_{LoRA}) while keeping the audio encoder and the base weights of smolVLM2 [37] frozen.

We utilize AudioSet [11], a large-scale dataset of 10-second audio clips paired with ontology labels. To ensure robustness, we process raw waveforms through a custom pipeline that resamples to 32kHz and extracts Log-Mel spectrograms, matching the input requirements of the DyMN [45] encoder.

To achieve robust alignment, we employ a Dual-Loss Strategy that optimizes for both geometric structure and functional generation:

1. Contrastive Loss (L_{con}): We apply a symmetric InfoNCE loss [44] on normalized embeddings to align the global geometry of the audio and text latent spaces. This pushes the projected audio embedding e_a closer to its corresponding text caption embedding e_t while distancing it from unrelated captions in the batch. This ensures high-level semantic clustering (e.g., separating "Industrial sounds" from "Nature sounds").

2. Generative Loss (L_{gen}): Geometric alignment alone does not guarantee that embeddings act as valid inputs for the LLM. We therefore add a Cross-Entropy loss on the VLM's next-token prediction task. We prompt the model with the projected audio embedding followed by a text anchor (e.g., "The sound of...") and train it to autoregressively generate the correct caption. This forces the projector to learn the specific magnitude and distribution statistics required to trigger the LLM's attention mechanisms effectively.

The final objective for Stage 1 is to minimize the weighted sum: $L_{stage1} = L_{con} + L_{gen}$.

3.2.2. Stage 2: Policy Learning (Audio-Action)

In the second stage, we freeze the entire perception stack, including the Audio Encoder, Projector, and the now-aligned VLM (with LoRA), to prevent forgetting of the

learned audio semantics. We then finetune the base smolVLA action expert on a robotic manipulation dataset incorporating audio fusion elements.

3.2.3. Audio Integration in Framework

We relied on HuggingFace's LeRobot open-source Python robotics framework [4] for all aspects concerning robotic manipulation: data collection, finetuning of action expert, and evaluation. The current framework does not support audio and only utilizes visual observations for LeRobot datasets and the featured policies. We thus had to first integrate audio into the framework. After doing some research we uncovered this is a feature in early development for the open-source project to integrate audio and tactile observations [40]. We joined this effort by utilizing the code and adapting it to fix many issues. We found and fixed many problems such as a microphone recording issue that lead to deadlocking and the inability to handle single channel microphones [36]. We plan after more refinement to contribute our work to the development of the feature.

3.2.4. Data Collection

Due to the high variance between environments and sensors, smolVLA requires fine-tuning data specific to an environment. Thus we collected a custom dataset using the SO-101 arm [20] on the task "If bell rings pick up the tape and put in the bin". The LeRobot community recommends over 50 training human-expert demonstration episodes for a particular task, thus we followed this recommendation and collected 50 episodes of the task with a limit of 60 seconds per episode [33]. The environment uses two cameras: one overhead camera on top of the workspace of resolution 640x480, and another front facing camera of resolution 1024x576, and one front mounted single-channel 44.1 kHz microphone. The episodes are recorded via tele-operation with a SO-101 Leader arm that directly controls the SO-101 Follower arm that actually does the task 8. The dataset is recorded in the experimental LeRobot Dataset format which includes audio 9. The particular task was chosen for simplicity for the action expert to grasp and simple logic in the instruction to ensure to not overwhelm the capabilities smolLM2-360M backbone, especially since the action expert is conditioned on only the first half of the layers that likely have less abstract understanding. We discuss the challenges of our task selection in 4.

We aimed for roughly 2/3 of the dataset to be positive examples, where a bell sound is played then the pick and place task is executed. The remaining third we aimed for negative examples where no bell sound is played and no action is performed for the entire duration of the episode. We also introduced examples of adversarial sounds from piano playing, to dog barking to isolate the response to bell ringing. We also tried to randomize the duration before a bell ring is heard.

3.3. Audio-SmolVLA Implementation

We adapted the open-source code of smolVLA within the LeRobot library [4] [46] for our implementation by extending its classes to incorporate audio in the forward pass. First we adapted the existing efficient audio pre-processor to closely follow the pre-processor used by DyMN [45]. During the forward pass observations (images + audio + motor states) are sampled at 10 Hz. Audio is chunked in a 1 second ring buffer. We project the audio using our audio projector into the embedding space of the smolLM2 language expert, and concatenate the embeddings along with the image embeddings, language instruction embeddings, and sensimotor states in the prefix that is fed through smolVLA. The frozen SmolVLM2 model [37] then processes the multimodal context (Sensorimotor State + Vision + Audio + Text instructions) and each layer's hidden features are cached from the first half layers. The Action Expert then uses these states as conditioning to diffuse the random noise into a sequence of joint actions via Flow Matching [28]. This stage relies on the standard flow-matching objective, minimizing the distance between the predicted vector field and the ground-truth action trajectory.

3.4. Inference

We ran inference on the model through the record dataset logic with the policy field specified as our custom "audiosmolvla" policy [32]. We did not have the time to test or fix asynchronous inference, and we found we did not require it for remote server inference 4.

4. Results

We evaluate our method by first analyzing the quality of the audio signal propagated into the smolLM2 embedding achieved in Stage 1, followed by the performance of the full audio-responsive policy in Stage 2.

4.1. Stage 1 Results

The primary objective of Stage 1 was to bridge the modality gap between the frozen DyMN audio encoder [45] and the SmolVLM [37] backbone. We assess the success of this alignment through three lenses: quantitative retrieval metrics, latent space topology, and qualitative generative grounding.

4.1.1. Quantitative Alignment Performance

To measure the geometric alignment of the projected audio embeddings, we evaluated the model on a held-out test split of AudioSet [11]. We computed Recall@k metrics using an in-batch retrieval task (Batch Size = 64), where the model must identify the correct text caption for a given audio clip among 63 negatives.

Our model achieved a Recall@5 of 92.3%, indicating a strong ability to discriminate between semantic concepts in

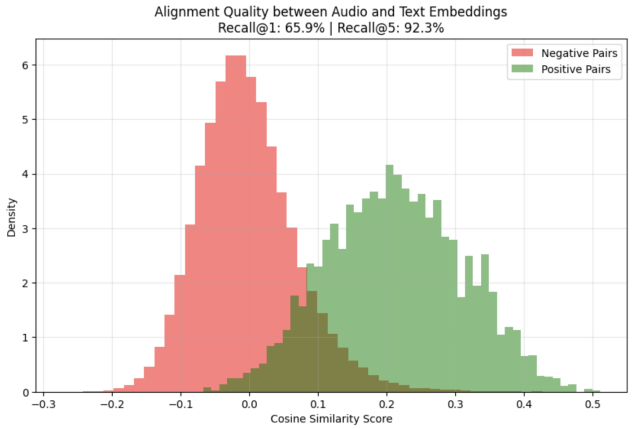

Figure 1. Density of Cosine Similarity scores for Positive vs. Negative Audio-Text pairs on the AudioSet test set. The clear separation between the distributions indicates robust discriminative capability.

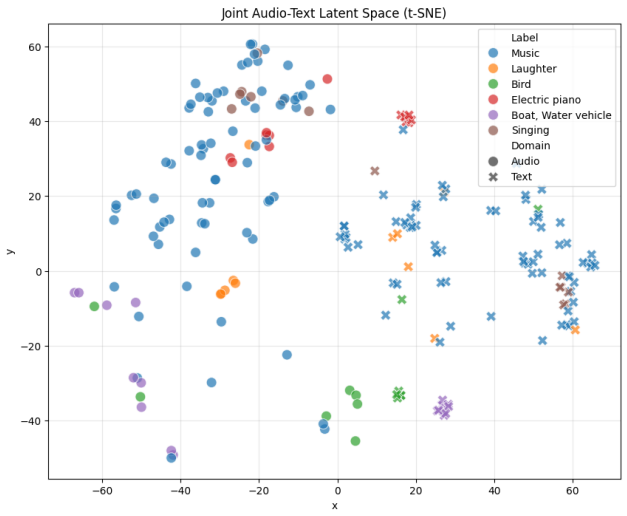

Figure 2. t-SNE visualization of the Joint Audio-Text Latent Space. Circles represent projected audio embeddings; Crosses represent ground-truth text embeddings. Colors indicate semantic classes. Note the "reflection" effect, where the internal structure of the audio clusters mirrors the text clusters.

the joint embedding space. As shown in Figure 1, the distribution of cosine similarity scores for positive pairs (audio-caption matches) is distinctively separated from negative pairs. This validates that the Audio Projector successfully mapped the audio features into the correct "angular" neighborhoods of the VLM's embedding space, satisfying the contrastive loss objective.

4.1.2. Latent Space Topology

While contrastive loss aligns vector directions, it does not guarantee that audio and text embeddings occupy the same manifold. To investigate the structure of the learned joint space, we visualized embeddings from the test set using t-SNE [50] (Figure 2). By doing so, we were able to observe two key phenomena.

First, the relative geometry of semantic clusters is preserved across modalities. For instance, "Electric Piano" embeddings (red) cluster at the "north" pole of both the Audio and Text distributions, while "Boat, Water vehicle" embeddings (purple) cluster at the "south." This confirms that the projector successfully learned the internal structural relationships between concepts. Additionally, compared to the tightly-clustered visualized text label embeddings, the audio embeddings are relatively more dispersed. This could be both interpreted positively or negatively: in a positive light, it could represent the model's ability to reflect the inherent in-class variance of the audio modality and retains the fine-grained acoustic details. On the other hand, it could signal that the model hasn't learned the nuanced complexities of the class relationships and cannot confidently classify the in-class examples with the same label.

Secondly, we observe a distinct separation between the audio (circles) and text (crosses) clusters. The audio embeddings seem to hold a similar structure to the text label embeddings, only projected over a plane. This "modality gap" suggests that while the semantic directions are aligned, the magnitude distributions of the projected audio tokens differ from the native text tokens. However, since the Action Expert in Stage 2 relies on relative embedding states rather than absolute text generation, the preservation of topological structure is the critical success factor.

4.1.3. Generative Grounding & Understanding

To verify that the alignment translates to functional understanding, we tasked the VLM with captioning raw audio clips using the prompt "The sound of...". This tests whether the projected audio embeddings can effectively drive smolVLM's [37] attention mechanism to generate accurate tokens.

10 generation examples are summarized in Figure 3, demonstrate strong first-token accuracy. The model correctly identifies distinct continuous signals, generating "Engine knocking" for engine sounds (Row 7) and "Wood" for tool sounds (Row 10). This proves the model is not merely guessing based on priors but is attending to the specific spectral features of the input. However, we also observed a tendency for "hallucination" after the correct initial identification, where the model lists related concepts (e.g., adding "Music" after correctly identifying "Crash"). Additionally, the model struggled with short impulse sounds like explosions, often defaulting to the "Music" cluster. For example, the model predicted sounds like "Bass" and "Thump" instead of "Explosion" sounds (Row 8).

For the purpose of robotic control, the strong grounding

	Ground Truth	Model Generation	Labels
0	I can hear Smash, crash.	Music, and Music of Latin America.\n\nA cappe...	[Smash, crash]
1	The sound of Water tap, faucet, Hands, Inside,...	Inside, small room, Speech, and Water.\n\nExa...	[Sink (filling or washing), Water tap, faucet,...
2	The sound of Heart murmur, and Speech.	Speech, and Pinging.\n\n\nA musical instrumen...	[Heart murmur, Speech]
3	Radio, and Music.	Music, and Electronic dance music.\n\nDrum an...	[Music, Radio]
4	The sound of Beep, bleep, Speech, and Female s...	Speech, and Speech synthesizer.\n\n\nSpeech s...	[Beep, bleep, Female speech, woman speaking, S...
5	A sound scene featuring Speech, Rattle, and Mu...	Rail transport, Railroad car, train wagon, an...	[Music, Rattle, Speech]
6	Detected audio events: Sniff, and Speech.	Speech, and Purr.\n\n\nDogs, cats, Domestic a...	[Sniff, Speech]
7	Detected audio events: Engine, Idling, and Eng...	Engine knocking, Engine, and Engine pistons, ...	[Engine knocking, Engine, Idling]
8	Detected audio events: Explosion.	Music, and Thump, thud.\n\n\nBass drum, tom, ...	[Explosion]
9	A sound scene featuring Radio.	Outside, rural or natural, and Punch.\n\n\nIn...	[Radio]
10	This clip contains Sawing, and Wood.	Wood, and Drying, Resting.\n\nThe word, as a ...	[Sawing, Wood]

Figure 3. Zero-shot audio captioning results for 10 samples. The model demonstrates strong semantic grounding, correctly identifying specific sound sources such as engines and tools, verifying that the audio embeddings act as valid soft prompts for the VLM.

of the initial semantic concepts (e.g., distinguishing ”Engine” from ”Nature”) is important, however for the use case of dealing with potential catastrophes (which often include crashing noises), our current iteration smolVLM [37] does not contain a good enough understanding of short audio signals.

4.2. Stage 2 Results

4.2.1. Training Results

The LeRobot community recommends 20,000 training steps for good performance with finetuning the smolVLA. Due to restrictions on compute resources we were able to achieve only 3,000 training steps for two versions of ”audiosmolvla”. One version we wrapped audio embeddings with special start and end audio tokens. These special added tokens to the language tokenizer were also projected to have similar embeddings to the word ”audio” to have semantic similarity without learning. The other version we removed special audio tokens, which follows how the base pretrained smolVLA [46] handles images with no image special tokens, with pretrained projector weights from stage 1 3 not being isolated. Due to an implementation error the special token approach also had randomly initialized audio projector weights.

We can see that after 3,000 steps the special audio tokens approach achieved a slightly lower training loss 5 than the approach without special audio tokens 4.

4.2.2. Evaluation

We are only able to provide qualitative assessment of evaluation for each trained policy. We recorded two evaluation datasets, where the SO-101 Follower arm actions are dictated by the policy as opposed to tele-operation controls from the SO-101 Leader. The policy without special audio tokens [34], performed adequately on the pick and place task with some errors and incompletions as well as slightly noisy actions. In terms audio performance, the policy seemed to be extremely sensitive to audio with no real distinction for the particular bell sound, with talking causing the arm to start the pick and place task. The policy trained with special audio tokens [35] seemed to hesitate more before starting the pick and place waiting for the bell sound, and we have one recorded instance where a clap did not cause the arm to move but the bell did. However, the policy would eventually move the arm without the bell cue. These datasets can be visualized with the Hugging Face space [24], although without audio.

The reasoning for the poor performance is likely in great part to the limited training steps. Especially with the new modality introduction we believe that more than the recommended base finetuning 20,000 steps are required. We believe also the dataset also should be extended past the recommended 50 steps in order to encapsulate the new modality better, as we see both policies being influenced more heavily on the base smolVLA pick and place fine tuning.

Another consideration is that the task could also be inherently flawed as waiting for an audio cue before starting a task poses alignment challenges where the policy can just learn to wait. The use of conditional logic in the instruction maybe overwhelming the reasoning capabilities of the simpler first half of the smolVLM backbone. We initially proposed a similar task to the one described in FuSe [19] where the policy learns to pick up an object with the same color as the button that plays a specific sound, but we believe this would have been too complex for the smolVLA model in terms of reasoning as its pretraining is almost exclusively pick and place, sorting, and stacking.

4.2.3. Challenges

The special tokens were removed in an attempt to improve training efficiency as both models had a speed of 400 steps/hour on a High RAM A100 in Google Colab [14], compared to the base smolVLA finetuning which is cited by the LeRobot to progress at a rate of 4,000 steps/hour. We believe there might be some inefficiencies with our added audio encoder and pre-processing steps that seem to not work well with Hugging Face’s accelerate library [15]. We provide the GPU utilization graphs for both the special token approach 7 and without 6, which show sparse GPU usage a likely indicator of inefficient CPU or I/O operations.

Figure 4. Action expert action chunk regression L1 loss over training steps for audiosmolvla with pretrained audio projector without special audio tokens from Weights & Biases

Figure 5. Action expert action chunk regression L1 loss over training steps for audiosmolvla with randomly initialized audio projector with special audio tokens from Weights & Biases

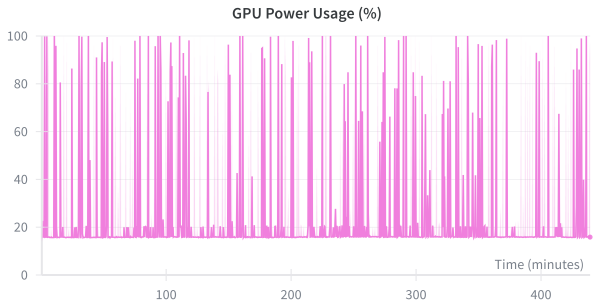

Figure 6. GPU utilization for audiosmolvla with pretrained audio projector and no audio special tokens from Weights & Biases

5. Future Works

Our inefficiency problems need to be solved to achieve the desired training steps we believe are necessary of over 20,000 steps in order for the policy to learn the audio modality effectively. We also believe a longer dataset needs to be recorded for a better designed task for audio. We believe maybe a task of responding to a bell via dropping an object would be better suited or tasks where audio is required in

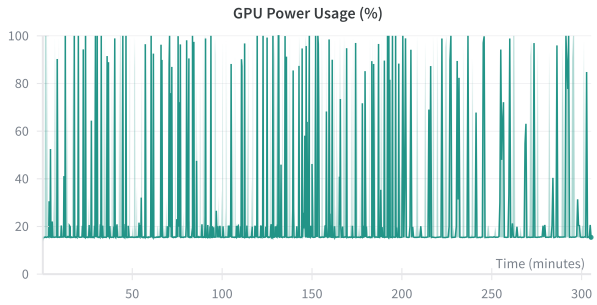

Figure 7. GPU utilization for audiosmolvla with randomly initialized audio projector and audio special tokens from Weights & Biases

tandem with vision such as plugging in an electrical socket [5]. We were unable to also evaluate our proposed policy on the LIBERO benchmark [29] used on the base smolVLA policy [46] in order to quantitatively determine the preservation of pretrained capabilities.

We want to also conduct a complete ablation study on audio modality projection. Fully compare the performance and efficiency on the permutation of: special audio tokens, pretraining of projector, pretraining of smolVLM backbone, CNN vs. Transformer based audio encoders.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. 2, 3

[2] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. 2, 4

[3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica,

Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. 1, 2, 3

[4] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024. 5

[5] CarolinePascal. Lerobot dataset: plug socket single padding. https://huggingface.co/datasets/CarolinePascal/plug_socket_single_padding, 2024. 8

[6] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection, 2022. 2, 4

[7] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 2

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2

[9] Soham Deshmukh, Satvik Dixit, Rita Singh, and Bhiksha Raj. Mellow: a small audio language model for reasoning, 2025. 2, 3, 4

[10] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap: Learning audio concepts from natural language supervision, 2022. 3

[11] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events, 2017. 4, 5

[12] Majid Ghasemi, Amir Hossein Moosavi, and Dariush Ebrahimi. A comprehensive survey of reinforcement learning: From algorithms to practical challenges, 2025. 1

[13] Arushi Goel, Sreyan Ghosh, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang gil Lee, Chao-Han Huck Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, and Bryan Catanzaro. Audio flamingo 3: Advancing audio intelligence with fully open large audio language models, 2025. 3

[14] Google. Google colaboratory, 2024. Accessed: 2024-05-20. 7

[15] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022. 7

[16] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units, 2021. 2

[17] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 2, 4

[18] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. 1, 2

[19] Joshua Jones, Oier Mees, Carmelo Sferrazza, Kyle Stachowicz, Pieter Abbeel, and Sergey Levine. Beyond sight: Finetuning generalist robot policies with heterogeneous sensors via language grounding, 2025. 1, 2, 3, 4, 7

[20] Rob Knight, Pepijn Kooijmans, Remi Cadene, Simon Alibert, Michel Aractingi, Dana Aubakirova, Adil Zouitine, Russi Martino, Steven Palma, Caroline Pascal, Francesco Capuano, and Thomas Wolf. Standard open so-100 & so-101 arms. https://github.com/TheRobotStudio/SO-ARM100, 2024. Online. 1, 2, 5

[21] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023. 2

[22] Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions, 2024. 2, 3

[23] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models?, 2024. 2, 3

[24] LeRobot Team. Lerobot dataset visualizer. https://huggingface.co/spaces/lerobot/visualize_dataset, 2024. Hugging Face Space. 7

[25] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 2, 3

[26] Zezeng Li, Alexandre Chapin, Enda Xiang, Rui Yang, Bruno Machado, Na Lei, Emmanuel Dellandrea, Di Huang, and Liming Chen. Robotic manipulation via imitation learning: Taxonomy, evolution, benchmark, and challenges, 2025. 1

[27] Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, Nadine Chang, Karan Sapra, Amala Sanjay Deshmukh, Tuomas Rintamaki, Matthieu Le, Ilia Karmanov, Lukas Voegtle, Philipp Fischer, De-An

Huang, Timo Roman, Tong Lu, Jose M. Alvarez, Bryan Catanzaro, Jan Kautz, Andrew Tao, Guilin Liu, and Zhiding Yu. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models, 2025. 3

[28] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. 5

[29] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. 8

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 2

[31] Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, and Xie Chen. An embarrassingly simple approach for llm with strong asr capacity, 2024. 2, 3

[32] Adam Malyshev and Shivam Garg. LeRobot Model: audiosmolvla. https://huggingface.co/AdamMalyshev/audiosmolvla, 2025. 5

[33] Adam Malyshev and Shivam Garg. LeRobot Dataset: Bell ring put tape in bin. https://huggingface.co/datasets/AdamMalyshev/bell_ring_put_tape_in_bin, 2025. 5

[34] Adam Malyshev and Shivam Garg. LeRobot Dataset: audiosmolvla evaluation on bell_ring_put_tape_in_bin task. https://huggingface.co/datasets/AdamMalyshev/eval_bell_ring_put_tape_in_bin_test, 2025. 7

[35] Adam Malyshev and Shivam Garg. LeRobot Dataset: audiosmolvla_random_init evaluation on bell_ring_put_tape_in_bin task. https://huggingface.co/datasets/AdamMalyshev/eval_bell_ring_put_tape_in_bin_random_init_tests, 2025. 7

[36] Adam Malyshev and Shivam Garg. lerobot-audio: Audio lerobot fork. https://github.com/adam-malyshev/lerobot-audio/tree/fix/audio_recording, 2025. Branch: fix/audio$_{r}ecoring$. 5

[37] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models, 2025. 2, 3, 4, 5, 6, 7

[38] Saeid Nahavandi, Roohallah Alizadehsani, Darius Nahavandi, Chee Peng Lim, Kevin Kelly, and Fernando Bello. Machine learning meets advanced robotic manipulation, 2023. 1

[39] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. 1, 3

[40] Caroline Pascal. lerobot: Lerobot audio dataset fork. https://github.com/CarolinePascal/lerobot/tree/user/CarolinePascal/2025_03_24_audio_dataset, 2025. Branch: user/CarolinePascal/2025_03_24_audio_dataset. 5

[41] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. 2

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[43] Rerun.io. Rerun: An open source sdk for logging, storing, querying, and visualizing multimodal and multi-rate data. https://github.com/rerun-io/rerun. 11

[44] Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. Infonce: Identifying the gap between theory and practice, 2024. 4

[45] Florian Schmid, Khaled Koutini, and Gerhard Widmer. Dynamic convolutional neural networks as efficient pre-trained audio models, 2023. 4, 5

[46] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. 1, 2, 3, 4, 5, 7, 8

[47] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 1

[48] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models, 2024. 1, 2, 3

[49] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. 3, 4

[50] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 6

[51] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-omni technical report, 2025. 3

[52] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 2

[53] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2

[54] Wei Zhao, Pengxiang Ding, Min Zhang, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation, 2025. 1

6. Appendix

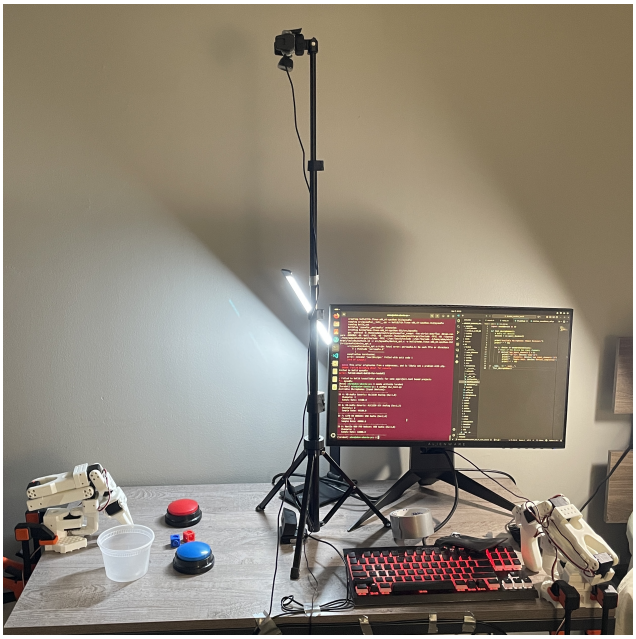

Figure 8. Data collection setup with SO-101 Leader arm pictured on the right, SO-101 Follower arm pictured on left, and cameras with microphone in middle.

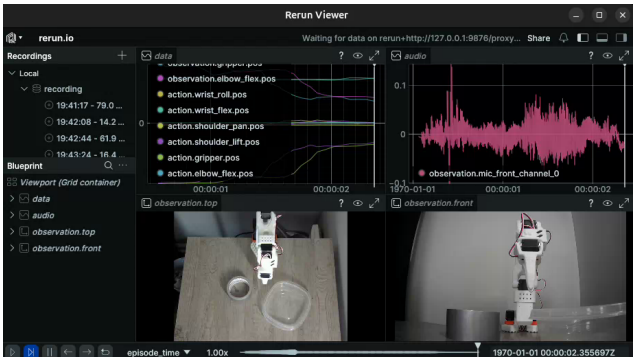

Figure 9. Screenshot of live data collected when recording dataset visualized with the rerun tool [43]

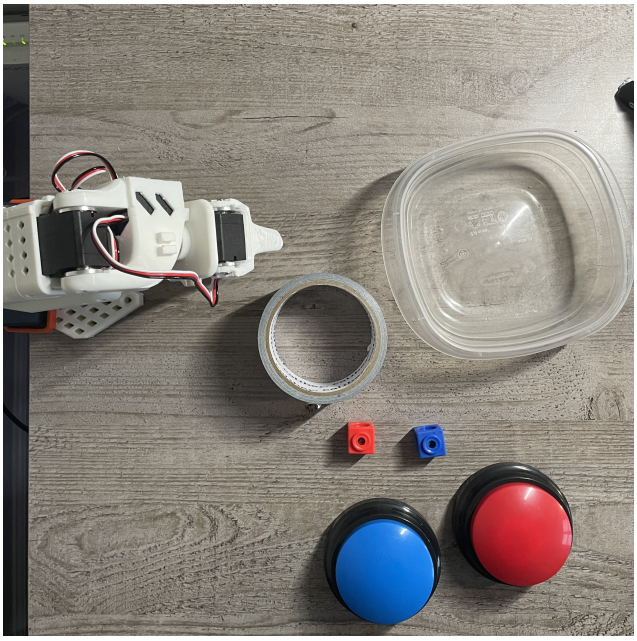

Figure 10. Subset of objects used and attempted to be used in the dataset recording. The small cubes proved too small and difficult for the manipulator and two camera setup.