A Algorithm for fine-tuning an agent

Algorithm 1 Fine-tuning the Dif-VAR and an RL agent

Inputs: A trained Dif-VAR $\mathbf{V}$, a trained policy π_θ , a subset of the original training data $\mathcal{D}_{old}$
Collect a small set of visual-audio pairs $\mathcal{D} = \{(\mathbf{I}_i, \mathbf{S}_i)\}_{i=1}^{U}$
$\mathcal{D}_{new} = \mathcal{D}_{old} \bigcup \mathcal{D}$
for a sampled minibatch $\{(\mathbf{I}_i, \mathbf{S}_i)\}_{i=1}^{N}$ from $\mathcal{D}_{new}$ **do** ▷ Fine-tune Dif-VAR
    Calculate empty intent label e_i by checking if $\mathbf{S}_i = \mathbf{0}_{l \times m}$
    Calculate image and sound embeddings: $\mathbf{h}^I, \mathbf{z}^I, \mathbf{h}^S, \mathbf{z}^S \leftarrow \mathbf{V}(\mathbf{I}_i, \mathbf{S}_i)$
    Calculate $\mathcal{L}_{\text{SSC}}$ by Eq. 7
    Calculate loss by $\mathcal{L}_{\text{finetune}} = \alpha_1 \mathcal{L}_{\text{SSC}} + \alpha_2 \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}_{\text{BCE}}(b^I(\mathbf{h}_j^I), e_j)$
    Update $\mathbf{V}$ to minimize $\mathcal{L}_{\text{finetune}}$
for $k = 0, 1, 2, ...$ **do** ▷ Self-supervised RL fine-tuning
    Sample a sound command $\mathbf{S}_g$ from $\mathcal{D}$ as goal
    for $t = 0, 1, ..., T$ **do**
        Receive RGB image $\mathbf{I}_t$ and robot state $\mathbf{M}_t$
        Calculate image and sound embeddings: $\mathbf{v}_t^I, \mathbf{v}_g^S \leftarrow \mathbf{V}(\mathbf{I}_t, \mathbf{S}_g)$ by Eq. 3
        Calculate reward $r_t = \mathbf{v}_t^I \cdot \mathbf{v}_g^S$
        if $\mathbf{S}_t$ **then**
            Calculate embeddings: $\mathbf{v}_t^S \leftarrow \mathbf{V}(\mathbf{S}_t)$
            $r_t = r_t + \mathbf{v}_t^S \cdot \mathbf{v}_g^S$
        Store $\{r_t, \mathbf{I}_t, \mathbf{M}_t, \mathbf{v}_t^I, \mathbf{v}_g^S\}$ in a memory buffer $\mathcal{D}_{RL}$
    Update π_θ with data from $\mathcal{D}_{RL}$ using PPO
    Clear $\mathcal{D}_{RL}$
return $\mathbf{V}, \pi_\theta$

B Robotic environment descriptions

The Row and Desk environments are developed in PyBullet [50] and focus mainly on manipulation tasks. In contrast, the iTHOR environment is developed in AI2-THOR [51] and is challenging in perception and designed for mobile robots.

B.1 Row

Four objects are placed in a line at a random location unknown to the robot on the table. A robot arm needs to move its gripper and stay above the object corresponding to a given command based on RGB images. The camera is placed at a fixed location on the side of the table such that it can capture the gripper and the objects from a distorted perspective. The relative positions of the gripper tip and the objects are initialized randomly at the beginning of an episode. A sound command only mentions the orindal information about the target object, and the robot needs to develop spatial reasoning skills to approach the target object using the relative positional information observed from the camera.

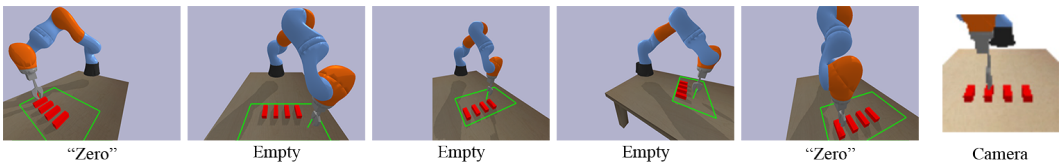

Figure 6: Visualization of the Row environment using Kuka-iiwa robot arm with paired images and voices from the Wordset. In this case, "zero" means the leftmost block, "one" means the second block from the left, and so on. The red and green rays are just for illustration purposes. The possible locations of the blocks are limited to the green rectangle and the end-effector location is indicated by the vertical ray. The rightmost figure shows the camera view.

B.2 Row - real

This environment is modified from the original Row environment. The Four objects are a mug, a soup can, a pudding box, and an orange. Different objects may require distinct grasping poses. See Fig. 7 for examples. At the end of an episode, the gripper performs a grasp by lowering its height from its current position, closing the fingers, and lifting the object up. For domain randomization, we randomize the background, camera viewpoint, and relative offset among the objects.

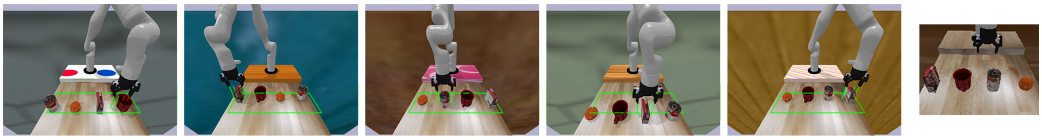

Figure 7: Visualization of the Row environment with paired images and voices from the Synthetic dataset under domain randomization setting. The grasping poses for the mug and the pudding box are different. The red and green rays are just for illustration purposes. The rightmost figure shows the camera view.

B.3 Desk

The Desk environment is modified from the CALVIN dataset [52]. A Franka Panda robot arm is placed in front of a desk with a sliding door and a drawer that can be opened and closed. On the desk, there is a button connected to an alarm clock, a switch to control a light bulb, and a pill case. The tasks of the robot include turning on or off the light bulb by manipulating the switch, pressing the button to mute the alarm clock and turn the LED of the clock into red, and picking up the pill case that could be on the top of the desk or inside a closed drawer. When the pill case is located inside a closed drawer, the robot needs to open the drawer before picking up the pill case. The sound commands come from FSC, ESC-50, and the Synthetic dataset.

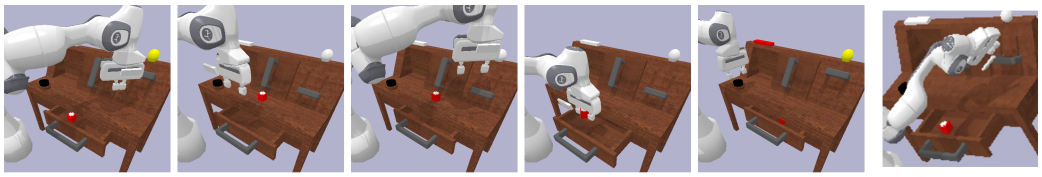

Figure 8: Visualization of the Desk environment with paired images and voices. The rightmost figure shows the camera view.

B.4 iTHOR

Our iTHOR environment uses real full-sentence speech commands to simulate a real-world application of household robots. The environment has 30 different floor plans of living rooms, each with its own set of decorations, furniture, and arrangements. The robot is given goal tasks such as switching the floor lamp or television on or off. The robot must navigate through the environment and interact with the intended object given RGB images and a noisy local discrete occupancy grid as the robot states. The complexity of the environment requires the agent to associate complicated speech commands with high-fidelity visual observations, without a floor plan map. The floor plans can be visualized and interacted with in https://ai2thor.allenai.org/demo/.

Figure 9: Visualization of the iTHOR environment with paired images and voices from the FSC dataset.

C Sound Data

Table 3: Sound signals used in the experiments.

Dataset	Sound	Examples
FSC	activate light	"Turn on the lights," "Lamp on"
	deactivate light	"Switch off the lamp," "Lights off"
	activate music	"Put on the music," "Play"
	deactivate music	"Pause music," "Stop"
	bring shoes	"Get me my shoes," "Bring shoes"
GSC	"0," "1," "2," "3"	"zero," "one," "two," "three"
	names of 4 objects	"house" "tree," "bird," "dog"
NSynth	C_4, D_4, E_4, F_4	Various instruments, tempo, and volume
US8K	bark, jackhammer	Sound recorded in the wild
ESC-50	Clock alarm	Alarm sound emitted from various alarm clocks
Synthetic	bring pill case	"Pass over the pill box for me," "Give me the pill case"
	first object	"I would like the first object," "Give me the leftmost object"
	second object	"Would you mind giving me the second object from the left," "Bring the third object from the right to me"
	third object	"Take the third object," "Bring me the second object from the right" "Take the third object from the left"
	fourth object	"Give the rightmost object to me ," "Hand over the fourth object"

D Visualization of task execution

D.1 Row

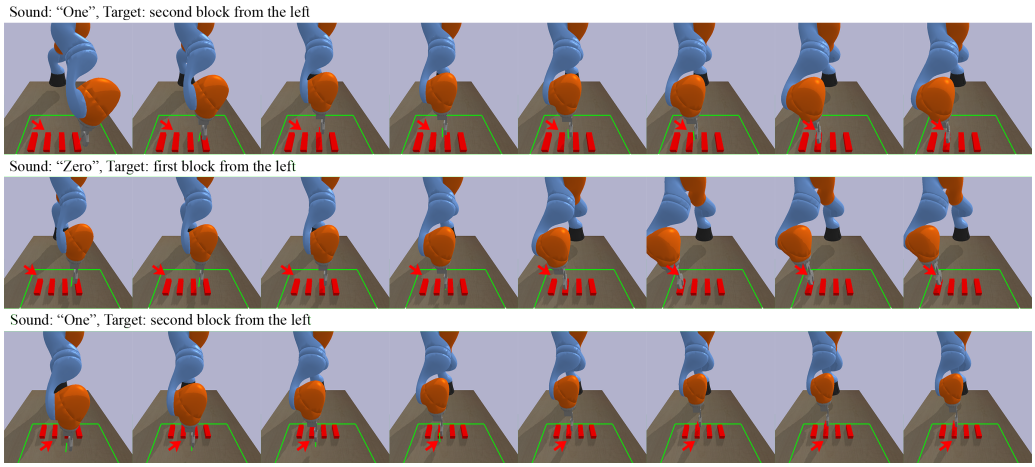

Figure 10: Visualization of the task execution in the Row environment after training without fine-tuning. The sounds come from Wordset dataset. Kuka moves its gripper to the target block successfully in all episodes.

D.2 Desk

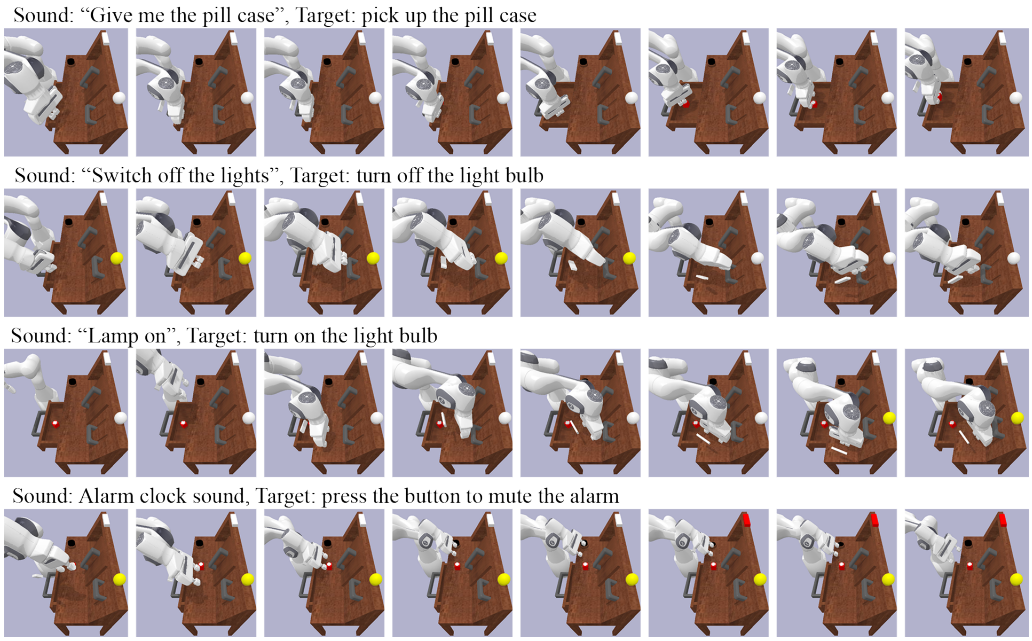

Figure 11: Visualization of the task execution in the Desk environment after training without fine-tuning.

D.3 iTHOR

Figure 12: Visualization of the task execution in the iTHOR environment after training without fine-tuning. The sounds come from FSC dataset. iTHOR agent finishes household tasks successfully in all episodes.

D.4 iTHOR fine-tuning

Figure 13: Visualization of the task execution in the iTHOR environment before and after the fine-tuning in unseen floor plans and the sound commands given by new speakers.

D.5 Desk fine-tuning

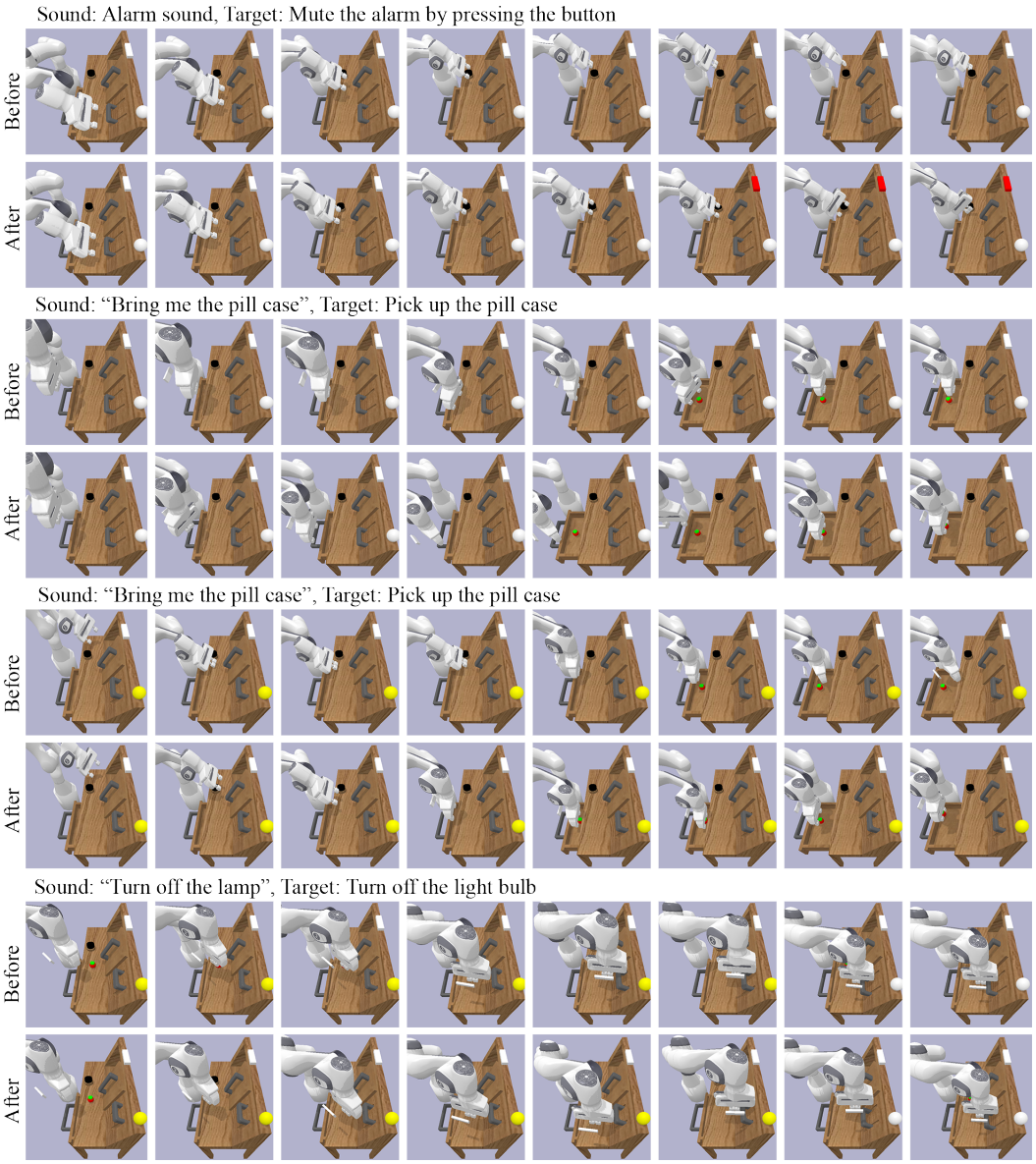

Figure 14: Visualization of the task execution in the Desk environment before and after the fine-tuning with a unseen desk and the sound commands given by new speakers. The appearance of the desk and the pill case are different from the original desk. The location of the light bulb, the button, the LED, and the drawer are different from the original desk.

E Time efficiency

In this section, we evaluate the time efficiency of all the methods. All the models are running on a single Nvidia GTX 1080 Ti GPU and a Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz. We report the average time in second (s) for the model to take one action in the iTHOR environment with the FSC dataset. The average is calculated from 12500 samples.

- ANR: 0.041s
- E2E: 0.018s
- VAR: 0.024s
- Dif-VAR: 0.022s

F About ASR+NLU+RL (ANR) pipeline

- Accuracy of intent prediction of ASR+NLU.
  FSC dataset: 86.0%; Wordset: 87.0%.