FROM INSTRUCTION TO EVENT: SOUND-TRIGGERED MOBILE MANIPULATION

Anonymous authors
Paper under double-blind review

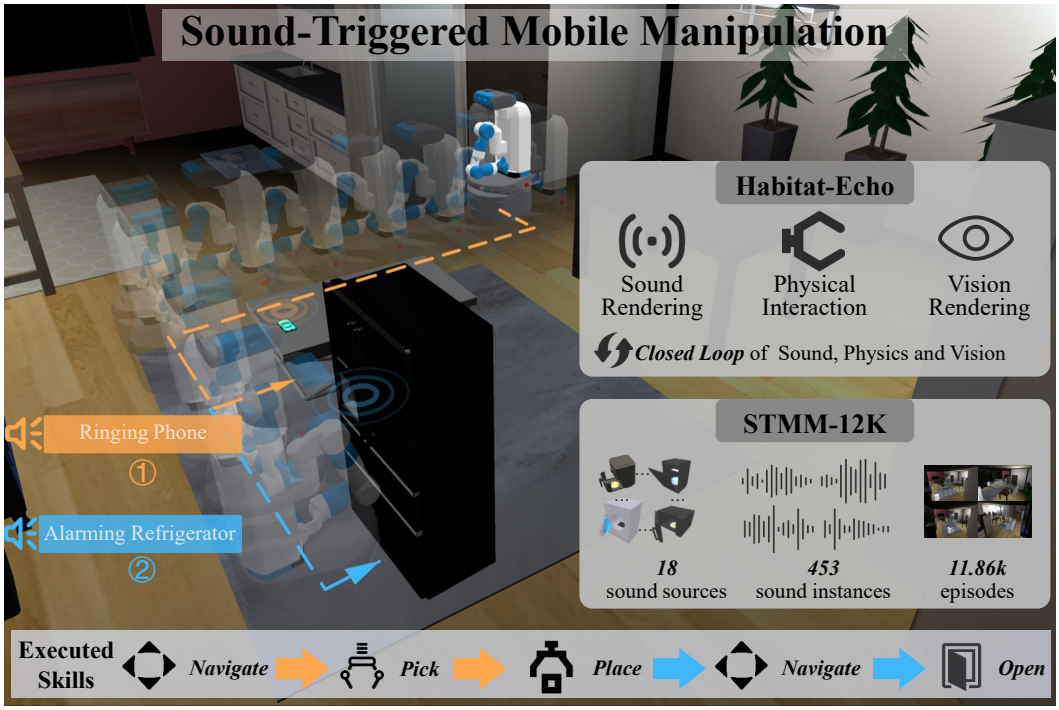


Figure 1: We introduce the problem of **Sound-Triggered Mobile Manipulation (STMM)**, where robots are required to infer task goals from environmental sound events, ground the corresponding sound-emitting targets, and execute appropriate physical interactions without explicit per-episode instructions. In multi-event scenes, robots must further prioritize and sequentially resolve concurrent events. To support this new task, we develop **Habitat-Echo**, a simulation platform that unifies sound rendering, visual perception, and physical interaction in a closed loop, and establish **STMM-12K**, a benchmark containing 18 sound-source categories, 453 sound instances, and 11.86k single-event and multi-event episodes.

ABSTRACT

Current mobile manipulation research predominantly follows an instruction-driven paradigm, where robots rely on predefined textual commands to execute tasks. However, this setting confines robots to a passive role, limiting robotic autonomy and the ability to react to dynamic environmental events. To address these limitations, we introduce Sound-Triggered Mobile Manipulation (STMM), where robots must actively perceive and interact with sound-emitting objects without explicit action instructions. To support STMM, we develop Habitat-Echo, a simulation platform that integrates sound rendering with physical interaction. We further propose a hierarchical baseline that translates high-level planning into low-level executions, where a task planner predicts a skill chain for policy models to execute sequentially. Experiments indicate the feasibility of perceiving auditory events and executing corresponding physical interactions without explicit instructions. Notably, in challenging multi-event scenarios, the robot successfully isolates the primary sources from overlapping acoustic interference to execute the first interactions, and subsequently proceeds to manipulate the secondary objects. These new challenges of planning-to-execution position STMM as a measurable research direction. *The code and datasets will be released upon acceptance.*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Recent advances in embodied AI have led to remarkable progress in mobile manipulation, enabling robots to navigate in unstructured environments and physically interact with surrounding objects (Song et al., 2025a; Bai et al., 2025; Yu et al., 2025; Chen et al., 2026). Such synergy between mobility and interaction empowers robots to perform long-horizon household tasks such as object rearrangement, multi-stage interaction, and sequential task execution (Ji et al., 2025; Huang et al., 2025; Li et al., 2025b; Escoriza et al., 2025; Song et al., 2025b; Lin et al., 2025), pushing embodied robots closer to general-purpose household assistance. Despite these advances, most existing mobile manipulation works are built upon an instruction-driven paradigm, where the task goal is explicitly specified by the user through instructions (Wu et al., 2025; Shah et al., 2025; Yan et al., 2025; Zhang et al., 2025; Liu et al., 2025a; Li et al., 2025a), and multimodal contextual cues (Wang et al., 2026). While this paradigm excels at completing user-specified chores, it also implicitly assumes that the task objective is known before execution begins. However, in real-world household environments, many tasks are triggered by environmental events that require a timely response. For example, a ringing refrigerator may prompt robots to approach and open it, and a ringing phone may indicate that the device needs to be turned off (see Figure 1). The unpredictable nature of these events makes it difficult for a user to monitor the scene and issue instructions in time. This limitation highlights the urgent need for *event-driven mobile manipulation*, where an intelligent robot should be able to proactively perceive and respond to environmental events.

Among various environmental signals, sound serves as a natural trigger for event-driven interaction. Unlike visual cues, sound is omnidirectional and independent of illumination. In fact, many of these events can be perceived acoustically long before they enter the visual field. Robots can therefore detect critical events even when the sounding target lies outside its field of view or under poor lighting conditions. Moreover, sound signals not only indicate the occurrence of an event, but also provide partial *semantic* cues about the event type and *spatial* cues about the location of the sound source. Motivated by this, we introduce the problem of **Sound-Triggered Mobile Manipulation (STMM)**, where robots must autonomously ground the sound source and execute appropriate interactions based on the sound event and scene context. Unlike conventional instruction-driven mobile manipulation, STMM requires robots to infer the task goal from environmental events rather than receive instructions or context directly from a human user. Concretely, we categorize sound-triggered mobile manipulation into two fundamental interaction types: (1) **object relocation**, where the robot localizes a sounding rigid object (*e.g.*, a phone) and alters its spatial position via pick-and-place skills, and (2) **state transition**, where the robot localizes a sounding functional or articulated object (*e.g.*, a refrigerator) and manipulates it to transition its internal state, *e.g.*, opening or closing. Building upon this, we investigate both single-event and more challenging multi-event settings. In the multi-event setting where multiple sound events occur simultaneously, robots must recognize concurrent events, determine the corresponding interactions, and compose them into an ordered execution plan according to a specified event priority. Overall, sound-triggered mobile manipulation requires robots to jointly reason about *what* event is happening, *where* the relevant sounding target is located, and *how* the correct interaction should be carried out, while additionally handling task scheduling in multi-event scenarios, making the problem particularly challenging.

However, existing embodied simulators either focus strictly on visual-physical interaction (*e.g.*, mobile manipulation simulator (Puig et al., 2024)) or visual-sound sensing (*e.g.*, sound navigation environment (Chen et al., 2020)). There is a lack of an integrated environment where robots can perceive spatial sound and physically interact with sound-emitting targets within a closed loop, hindering systematic research on sound-triggered mobile manipulation. To bridge this gap, we develop **Habitat-Echo**, a unified sound-visual-physical simulation platform. By integrating spatial sound rendering into existing interactive embodied environments and extending them with abundant sound-emitting interactive assets, Habitat-Echo enables a closed-loop simulation of sound perception, sound source grounding, and physical interaction within a single unified platform.

Built on Habitat-Echo, we further establish **STMM-12K**, the first benchmark for sound-triggered mobile manipulation, covering diverse sounding objects and interaction scenarios under both single-event and multi-event settings. To provide a first-step solution to this new problem, we design a hierarchical baseline framework that decomposes sound-triggered mobile manipulation into high-level acoustic reasoning and low-level physical execution. For high-level acoustic reasoning, a task planner is leveraged to translate sound-visual observations into an executable skill chain and organize

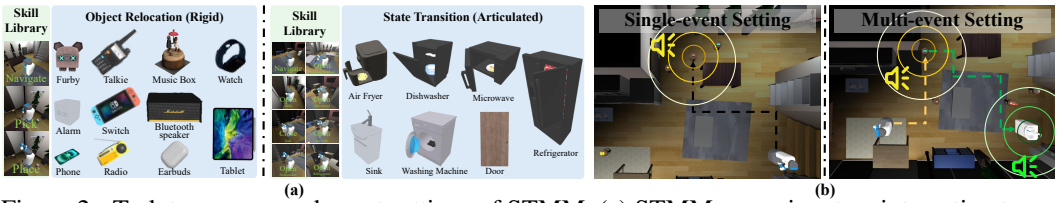


Figure 2: Task taxonomy and event settings of STMM. (a) STMM comprises two interaction types according to the physical properties of sound-emitting targets: *object relocation* for rigid sources and *state transition* for articulated sources. (b) Both interaction types are instantiated under *single-event* and *multi-event* settings, and the latter requires robots to resolve concurrent events sequentially according to a specified priority.

it according to the specified event priority. For low-level physical execution, a set of lightweight policy models is invoked sequentially to carry out each required skill according to the predicted skill chain. Each policy model focuses on a specific skill, such as navigation, pick, place, and open door, enabling specialized control for diverse interaction types. Experiments on STMM-12K shows: (1) completion success rates with ground-truth plans and privileged target positions reach 86.13% for object relocation but only 30.38% for state transition. (2) Supervised task planners achieve planning success rates up to 99.03% on object relocation but 69.50% on the multi-event setting. Together, these results highlight the challenges, particularly the cascade completion for state transitions and the task planning for the multi-event setting. Our contributions are summarized as follows:

- We propose Sound-Triggered Mobile Manipulation (STMM), a novel event-driven embodied task in which robots are required to respond to sound events physically without relying on explicit per-episode instructions. We define two interaction types, object relocation and state transition, and instantiate each under both single-event and multi-event settings.
- We develop Habitat-Echo, a unified embodied simulation platform that unifies spatial sound rendering, visual observation rendering, and interactive physical interaction in a close loop. Based on Habitat-Echo, we establish STMM-12K, the first benchmark for STMM containing 11.86k episodes (10.47k train / 1.39k test), 453 sound instances, and 18 sound sources.
- Experimental results on STMM-12K show the feasibility and challenges of our hierarchical baseline. Specifically, the baseline achieves a 71.05% completion success rate on single-event object relocation, showing the feasibility of relocating rigid sound sources. Meanwhile, single-event state transitions pose completion challenges (28.67%), while the multi-event setting presents additional task planning challenges (69.50%).

2 Related Work

Sound provides crucial spatial and contextual information for embodied agents. Existing embodied AI leverages sound primarily for either navigation or manipulation. Below, we review these two lines of work in turn. In Sound-Visual Navigation (Chen et al., 2020), methods address sound rendering (*e.g.*, semantic emissions (Chen et al., 2021), moving sources (Younes et al., 2023), distractors (Yu et al., 2022)) or cross-modal learning (Wang et al., 2022a; Kondoh & Kanezaki, 2025; Wang et al., 2022b), yet largely neglect physical interaction. Works in robotic manipulation utilize sound merely as auxiliary context, such as for process monitoring (Wang et al., 2025b), intent inference (Wang et al., 2026), or 2D masking (Guo et al., 2026). Unlike these approaches that treat sound as a complementary cue, STMM casts the sound event as the explicit task trigger and the sound source as the physical target, unifying grounding, navigation, and manipulation into a closed loop. *We defer further discussion to the supplementary material.*

3 Sound-Triggered Mobile Manipulation

3.1 Problem Formulation

Unlike conventional instruction-driven mobile manipulation, Sound-Triggered Mobile Manipulation (STMM) requires the robot to respond to sound events without a complete per-episode task instruction. We model STMM as a partially observable Markov decision process, $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \gamma \rangle$,

where $\mathcal{S}$, $\mathcal{A}$, and Ω denote the state, action, and observation spaces, respectively. $\mathcal{P}$ specifies the joint distribution of the next state and observation given the current state and action, while $\mathcal{R}$ and γ denote the reward function and discount factor, respectively. At each time step t, the robot receives a multimodal observation $O_t = \{V_t, S_t, P_t\} \in \Omega$, where V_t, S_t, and P_t denote visual observations, sound observations, and proprioceptive states, respectively. From these observations, the robot must predict the task goal $\mathcal{G}$ associated with sound events E_{sound} and execute actions $a_t \in \mathcal{A}$ to navigate and interact with the relevant sound sources. In the multi-event setting, event priorities are additionally provided (see Section. 3.2). Under our evaluation protocol, an episode is successful only if the required skill chain is correctly predicted and all constituent skills are successfully executed in order (see Section. 3.3). As shown in Figure 2, we categorize the interactions into two fundamental types according to the physical properties of the corresponding sound sources:

Object Relocation. For rigid sound sources (*e.g.*, a ringing phone in Figure 2a), the robot resolves the event by relocating the object. This requires localizing, approaching, and manipulating the source, which requires translating cross-modal observations into executable interactions.

State Transition. For articulated sound sources (*e.g.*, a beeping washing machine in Figure 2b), the robot resolves the event by transitioning the object's internal physical state. Unlike spatial relocation, this interaction requires operating along specific kinematic trajectories, imposing stricter constraints on local geometry and manipulation precision. For example, opening a washing machine necessitates precisely grasping and rotating its handle along the functional axis.

3.2 Task Setting

Beyond the interaction types defined above, we further design single-event and multi-event settings for STMM, which differ in the complexity of source localization and task execution.

Single-event Setting. In the single-event setting, only one sounding event is present in the scene. The robot is required to infer the event-conditioned goal from the multimodal observation, localize the relevant target, and execute the corresponding interaction to resolve the event. Depending on the physical properties of the sounding target, the required interaction can take the form of either object relocation or state transition.

Multi-event Setting. In the multi-event setting, multiple sound events occur simultaneously, resulting in acoustic interference and sequential decision-making challenges. Given an event-priority specification, the robot must identify the concurrent events, associate each event with the corresponding interaction, and execute the resulting skill chains in the prescribed order. After the first event is resolved, its corresponding sound disappears and the acoustic scene changes, requiring the robot to continue interacting with the remaining target under an updated environment state.

3.3 Skill Space

Directly mapping raw observations to low-level actions entangles high-level reasoning with physical execution, increasing learning burden and hindering proactive decision-making. To address this, STMM decouples the entangled process into reusable skill primitives, each serving as an atomic sensorimotor mapping for specific interactions. Specifically, the skill space is defined as:

$$\mathcal{K} = \{\texttt{Navigate}, \texttt{Pick}, \texttt{Place}, \texttt{CloseSink}, ..., \texttt{OpenRefrigerator}\}. \tag{1}$$

Here, `Navigate` denotes approaching the relevant sounding target. `Pick` and `Place` correspond to rigid-object relocation via the robot gripper. `CloseSink` and `Open Refrigerator` correspond to state-transition interactions. The detailed execution spaces and termination criteria are elaborated in the supplementary materials. Under this abstraction, different events can be solved through different skill compositions, *i.e.*, the *Skill Chain*. An event is considered resolved successfully if and only if all constituent skills are successfully executed in sequence. Object relocation is typically instantiated as:

$$[\texttt{Navigate} \rightarrow \texttt{Pick} \rightarrow \texttt{Place}], \tag{2}$$

whereas state transition is instantiated as:

$$[\texttt{Navigate} \rightarrow \texttt{CloseSink}]\ ,\ ....,\ [\texttt{Navigate} \rightarrow \texttt{OpenRefrigerator}]. \tag{3}$$

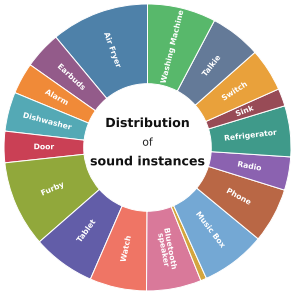
For multi-event scenes, the complete task is further represented as the ordered composition of multiple skill chains conditioned on a specified event priority. In this way, the skill chain serves as a structured bridge from latent event interpretation to executable embodied responses.

Simulator	Sound Rendering	Physical Interaction
Habitat (Szot et al., 2021)	✗	✓
AI2-THOR (Kolve et al., 2017)	✗	✓
Isaac (Makoviychuk et al., 2021)	✗	✓
SoundSpaces (Chen et al., 2020)	✓	✗
Habitat-Echo (Ours)	✓	✓

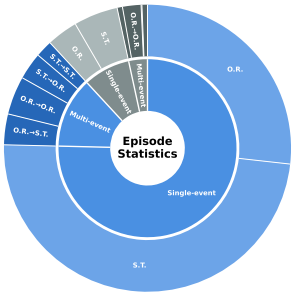
(a)



(b)



(c)



(d)

Figure 3: (a) Comparison of different simulators. Habitat-Echo supports both sound rendering and physical interaction. (b) Visual distractor objects. STMM-12k includes *77* additional visual distractor objects. (c) Distribution of sound instances. Instances are color-coded by category. (d) Episode statistics. Train and test splits. (10.47k and 1.39k episodes) O.R. denotes Object Relocation, and S.T. denotes State Transition. Interaction type of the first sound source → the second one.

4 Platform and Benchmark

4.1 Habitat-Echo Simulation Platform

Existing simulators either prioritize visual-physical manipulation (Kolve et al., 2017; Makoviychuk et al., 2021; Szot et al., 2021) or strictly focus on sound-visual navigation (Chen et al., 2020; 2022), leaving sound perception and physical interaction disconnected (see Figure 3a). To support STMM, we develop Habitat-Echo, which couples spatial sound rendering with interactive object dynamics under a unified simulation state, further enabling sound-visual-physical closed-loop interaction.

Sound-Physical Simulation. Habitat-Echo builds upon interactive indoor 3D environments (Szot et al., 2021), retaining visual rendering and physical interactions while integrating the spatial sound capabilities of SoundSpaces (Chen et al., 2020). At each time step, the simulator retrieves the Room Impulse Response (RIR) corresponding to the current source–receiver spatial configuration and computes binaural sound observations from the source waveform. As the robot moves, the resulting sound observation changes consistently with the evolving spatial relationship between the robot and the sound source. Habitat-Echo further supports simultaneous rendering of multiple active sources, allowing overlapping acoustic events to be perceived within the same scene. Importantly, acoustic events are coupled with the physical states of their corresponding objects. Once an event is physically resolved, its associated sound is removed from the subsequent observation, producing an updated acoustic field for subsequent interaction. This coupling allows robot actions to affect not only the physical environment but also future sensory observations, forming a sound–visual–physical closed loop.

Interactive Sound-Event Assets. To instantiate such closed-loop interactions, we extend the environment with a collection of sound-emitting interactive assets for STMM. These assets cover both interaction types in Section. 3, including graspable sounding rigid bodies (*e.g.*, phones, alarms) and articulated objects (*e.g.*, doors, sinks with faucet handles), providing an asset foundation for both object relocation and state transition tasks. To enhance physical realism, these assets include coupled dynamics (*e.g.*, water flow syncing with faucet handles) and populated interiors (*e.g.*, stocked refrigerators). These assets are explicitly designed to act as sound sources and manipulation interaction targets, thereby closing the loop between perception and action. We further populate scenes with silent instances from sound-source categories and unrelated visual distractors (see Figure 3b), preventing agents from resolving events solely through object-category priors.

Skill Interface. Habitat-Echo exposes standardized interfaces for the skill space $\mathcal{K}$ defined in Section. 3.3, including navigation, object relocation, and articulated-object manipulation. Different skills operate over their corresponding observation and action spaces, such as audio-visual observations for `Navigate` and joint-constrained actions for `OpenRefrigerator`. Each skill is equipped with consistent initialization, termination, and success criteria, allowing heterogeneous low-level policies to be composed into executable skill chains.

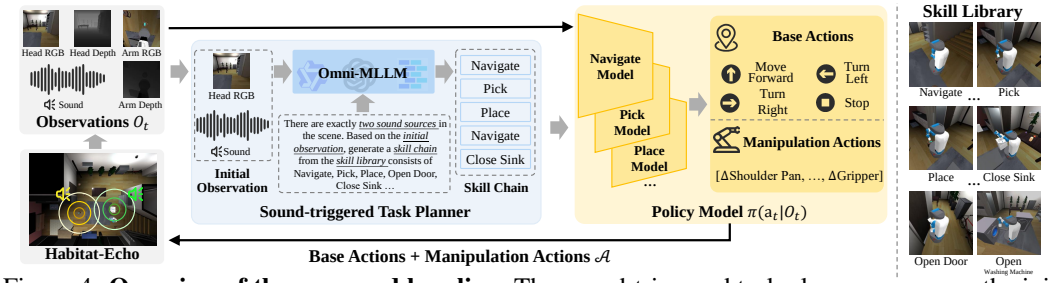


Figure 4: **Overview of the proposed baseline.** The sound-triggered task planner processes the initial observation to reason and generate a high-level skill chain from the skill library *(right)*. Guided by this chain, specialized policy models are sequentially activated to generate low-level actions and interact with Habitat-Echo.

4.2 STMM-12K Benchmark

STMM-12K is designed to evaluate whether an embodied agent can predict interaction goals from acoustic events and physically resolve events under diverse objects, sounds, and compositions. The benchmark follows three complementary design principles: (1) *interaction diversity*, covering both object relocation and state transition; (2) *acoustic generalization*, requiring robots to handle unheard instances of known sound events; (3) *event complexity*, ranging from single-event episodes with silent distractors to multi-event episodes with concurrent sounds and sequential interactions.

Dataset Construction. STMM-12K covers the two interaction types defined in Section. 3, *i.e.*, object relocation and state transition, under both single-event and multi-event settings. As shown in Figure 2, the benchmark contains 18 interactive sound sources in total, including 11 graspable rigid objects such as phones and alarms, and 7 articulated objects such as doors, sinks, and household appliances. Each category is associated with multiple real-world sound recordings, yielding 453 acoustic instances and providing diverse realizations of the same semantic event. To prevent agents from exploiting visual-category shortcuts, we introduce silent source-category distractors (see Figure 2) and unrelated visual distractors (see Figure 3b). The former require agents to distinguish acoustically active targets from visually plausible alternatives, while the latter increase scene clutter, jointly encouraging acoustic grounding rather than category-level visual recognition.

Episode Generation. We employ a procedural randomization process to generate episodes for training and testing. For each episode, a sound-source category is first sampled, followed by a specific recording instance from that category. The robot and target objects (except for doors) are then placed at randomized valid locations, while infeasible configurations are rejected to ensure start-to-goal traversability and physical accessibility for the required manipulation. In the multi-event setting, two sound sources are randomly placed and validated for physical accessibility and traversability. One is then randomly selected as the main source, and the priorities of sound sources are recorded. STMM-12K comprises 11.86k episodes in total, including 8.97k/0.99k train/test episodes for the single-event setting, and 1.50k/0.40k for the multi-event setting (see Figure 3d). The sound instances are strictly divided into training (311) and testing (142) sets.

Evaluation Protocol. We evaluate STMM at both the task planning and task completion levels. *Planning Success Rate (PSR)* measures whether the agent correctly predicts the event-conditioned skill chain. *Completion Success Rate (CSR)* adopts a stricter criterion that an episode is counted as successful only when the required plan is correct and every constituent skill is successfully executed in the prescribed order.

5 Hierarchical Baseline for STMM

The proposed baseline comprises two core components, *i.e.*, a sound-triggered task planner and policy models (see Figure 4). The sound-triggered task planner acts as the high-level reasoner, analyzing initial observations to predict a logical skill chain. Following this chain, the policy models serve as the low-level executors, sequentially translating each assigned skill into concrete actions to complete the corresponding subtasks.

5.1 SOUND-TRIGGERED TASK PLANNER

We propose a sound-triggered task planner that operates as the high-level reasoning module. This module is responsible for translating environmental cues into a plan of skills, *i.e.*, a skill chain (see Figure 4). The input for the task planner is twofold, comprising initial sensory observations and a skill library. Regarding observations, the planner utilizes the sound to reason about the target's category and relative orientation, while the head RGB provides an initial egocentric view to partially perceive the scene layout. Crucially, to prevent information leakage, we restrict the planner's input to strictly align with the observation space of the Navigate skill, ensuring no additional information is accessed. Complementing these observations, the skill library is provided as a textual prompt or category embeddings for the task planner. The library defines the semantic functionality of all available skills to support reasoning of the planner. By combining these multi-modal observations with the skill descriptions, the task planner outputs an ordered chain of skills. This chain acts as a high-level directive, guiding the sequential activation of the following policy models to complete the task.

5.2 SKILL-SPECIFIC POLICY MODEL

Policy models serve as low-level executors, translating the skill chain from the task planner into concrete actions and sequentially interacting with the environment. Policy models in the proposed baseline share a common architecture comprising modality encoders (Chen et al., 2020; Oquab et al., 2024), a temporal module, and actor-critic heads. The temporal module and actor-critic heads follow prior work (Chen et al., 2020; Gu et al., 2023), and the modality encoders are tailored to the specific observation space of different skills. For the Navigate skill, the observation space includes sound, head RGB, and head depth. In single-event and multi-event settings, sound observations are converted into binaural spectrograms. The navigation policy uses a vanilla CNN encoder (Chen et al., 2020), while multi-event navigation additionally incorporates a location predictor (Chen et al., 2021) for sound source localization. Manipulation skills include Pick, Place, Open Door, Open Refrigerator, Open Washing Machine and so on. The observation spaces consist of head and arm depth and are processed by a CNN encoder (Chen et al., 2020; Gu et al., 2023).

6 EXPERIMENT

Implementation Detail. We optimize Actor-Critic policies using PPO (Schulman et al., 2017) on two NVIDIA 4090 GPUs for 5×10^5 environment steps. Optimization relies on Adam (Kingma, 2014) with an initial learning rate of 3×10^{-4} (linear decay), a weight decay of 1×10^{-5}, and a 0.5 maximum gradient norm. Key PPO hyperparameters include a 128-step rollout, 2 update epochs with 2 mini-batches, a 0.99 discount factor, a 0.95 GAE parameter, and a 0.2 clipping range. The value and entropy loss coefficients are set to 0.5 and 0.01, respectively. The skill-specific training rewards, detailed execution and termination criteria are provided in the supplementary material.

Baseline. We combine individually trained low-level policies with various task planners to compose three baselines. (1) **Oracle**: Uses ground-truth skill chains, serving as the planning upper bound. (2) **Zero-shot Omni-MLLMs**: We evaluate training-free models including Qwen2.5-Omni (3B, 7B) (Xu et al., 2025a), MiniCPM-o-4.5 (Cui et al., 2026), and Qwen3-Omni-30B-Instruct (Xu et al., 2025b). (3) **Supervised Setting-Specific Planner (SSP)**: A lightweight network is trained to map initial observations to skill chains via sound category prediction.

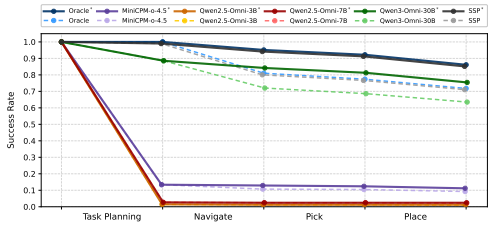
6.1 RESULT

Quantitative Results of Object Relocation in Single-Event setting. In Figure 5a and Figure 5b, the Oracle* baseline, which utilizes ground-truth planning and target position, achieves a completion success rate of 86.13%. Regarding task planning, Qwen3-Omni-30B achieves the highest accuracy of 88.56%, surpassing other zero-shot planners. Supervised Setting-Specific Planner (SSP) achieves a 99.03% planning rate and a 71.05% completion success rate.

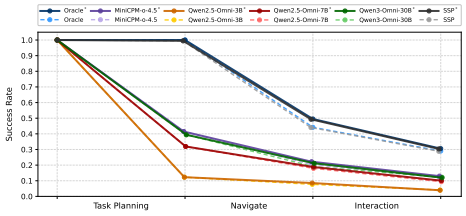
Quantitative Results for State Transition in the Single-Event Setting. Figure 5a and Figure 5c show the task planning and completion success rates. The Oracle* baseline (where * denotes adopting the target position) receives a completion success rate of 30.38%. SSP closely approaches this

Method	Single-event setting				Multi-event setting	
	Object Relocation		State Transition			
	Planning (↑)	Completion (↑)	Planning (↑)	Completion (↑)	Planning (↑)	Completion (↑)
Oracle	100.00	71.78	100.00	28.84	100.00	11.25
Oracle*	100.00	86.13	100.00	30.38	100.00	19.00
Qwen2.5-Omni-3B	1.46	0.97	12.29	4.10	26.25	2.50
Qwen2.5-Omni-7B	2.68	2.19	31.91	9.39	44.50	3.25
MiniCPM-o-4.5	13.38	9.25	***41.47***	12.12	***50.00***	***5.00***
Qwen3-Omni-30B	***88.56***	***63.50***	39.42	***12.29***	19.25	3.00
SSP	**99.03**	**71.05**	**99.49**	**28.67**	**69.50**	**7.75**
Qwen2.5-Omni-3B*	1.46	1.22	12.29	3.92	26.25	2.50
Qwen2.5-Omni-7B*	2.68	2.43	31.91	10.07	44.50	5.25
MiniCPM-o-4.5*	13.38	11.19	***41.47***	***12.97***	***50.00***	***8.75***
Qwen3-Omni-30B*	***88.56***	***75.43***	39.42	11.95	19.25	4.25
SSP*	**99.03**	**85.16**	**99.49**	**30.20**	**69.50**	**13.25**

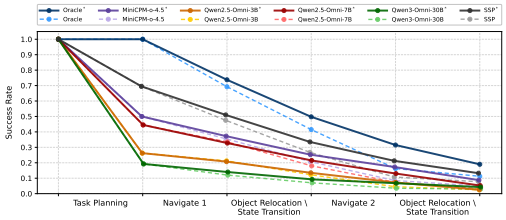
(a)



(b)



(c)



(d)

Figure 5: Results on STMM-12K. (a) Success rates of task planning and skill completion. Fully-supervised SSP achieves the best performance in both settings. **Best** and ***second best***. Progressive completion success rate of (b) object relocation (single-event), (c) state transition (single-event), and (d) multi-event setting. * denotes the target position that is provided for the navigation policy model.

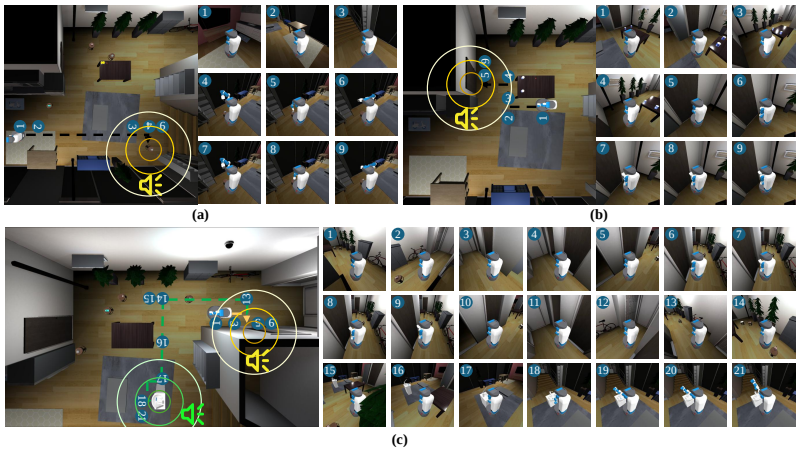


Figure 6: Qualitative Visualizations of Task Execution. Execution trajectories for (a) Object Relocation, (b) State Transition, and (c) Multi-event Setting. For each task, the left panel maps robot paths and sound sources from a top-down view, while the right panel shows third-person keyframes. Multi-event Setting requires sequential navigation and interaction with two distinct sound sources.

with 99.49% planning and 28.67% completion success rates. Among zero-shot Omni-MLLMs, MiniCPM-o-4.5 leads in planning (41.47%), outperforming Qwen2.5-Omni models and followed closely by Qwen3-Omni-30B (39.42%). This translates to completion rates of 12.12% and 12.29% for MiniCPM-o-4.5 and Qwen3-Omni-30B, respectively. Current zero-shot planners lag behind specialized supervised models in reliable task planning.

Quantitative Results of Multi-event Setting. We report the multi-event results in Figure 5a. Among zero-shot task planners, MiniCPM-o-4.5 achieves the highest task planning accuracy, followed by Qwen2.5-Omni-7B. The supervised Setting-Specific Planner (SSP) outperforms all zero-shot planners. Oracle* and SSP achieve completion success rates of 19.00% and 7.75%, respectively. We show the progressive completion success rate in Figure 5d. Under interference from the second sound source, Oracle* and SSP complete the first event with success rates of 49.75% and 26.75%, respectively. These results highlight the difficulty of the multi-event setting, where concurrent acoustic interference and cascading errors across sequential skills hinder completion (see Section. 3.2).

Qualitative Result. We visualize executions of the robot in single-event and multi-event settings (see Figure 6). In the object relocation (single-event), the robot successfully navigates to the target

Table 1: Ablation Study. (a) Effect of sound modality on navigate policy model. Cat. and Tgt. Pos. denote the category embedding and target position of the sound source, respectively. (b) Task Planning. Combining Head RGB and sound is the best. (c) Policy model for Navigate (Visual Modality). Using Head RGB and Head Depth yields the highest. (d) Policy model for Pick and Open Washing Machine (O.W.M.). Combining Head and Arm Depth outperforms single-view.

(a)

Model	Input Modality			S.R. (↑)
	Sound	Cat.	Tgt. Pos.	
I	✗	✗	✗	0.00
II	✗	✓	✗	0.00
III	✓	✗	✗	81.02
IV	✗	✗	✓	93.43
V	✓	✗	✓	**95.13**

(b)

Model	Input Modality		S.R. (↑)
	Head RGB	Sound	
I	✓	✗	52.31
II	✗	✓	98.30
III	✓	✓	**99.03**

(c)

Model	Input Modality		S.R. (↑)
	Head RGB	Head Depth	
I	✗	✓	72.74
II	✓	✗	78.58
III	✓	✓	**81.02**

(d)

Model	Input Modality		S.R. (↑)	
	Head Depth	Arm Depth	Pick	O.W.M.
I	✓	✗	92.70	68.91
II	✗	✓	92.94	56.75
III	✓	✓	**96.59**	**70.27**

and executes the Pick and Place skills (see Figure 6a). In the state transition (single-event), the robot accurately navigates to the sound-emitting door and completes the Open Door skill (see Figure 6b). The multi-event setting involves two simultaneous sound sources: a primary source and a secondary source (see Figure 6c). Following the specified event priority, the robot first interacts with the primary source despite interference from the secondary one and then interacts with the secondary source. More visualizations can be seen in the supplementary material.

6.2 Ablation Study and Further Discussion

Effect of Sound Modality for Navigate Skill. We ablate the input modalities of the navigation skill(see Table 1a), where sound is processed into binaural spectrograms for implicit positioning. Models lacking spatial data fail completely (Model I and II, 0.00%). Success scales up when using sound only (Model III, 81.02%) or explicit target positions (Model IV, 93.43%). Combining both achieves the optimal 95.13% (Model V). This confirms that spatial information is indispensable and sound serves as an effective positional cue complementary to explicit coordinates.

Effect of input Modality for Task Planning. As shown in Table 1b, we ablate the input modalities used for task planning in object relocation. We utilize Setting-Specific Planner (SSP) as the baseline. Relying exclusively on Head RGB yields the lowest success rate at 52.31%. Utilizing only sound signals substantially elevates the performance to 98.30%. Integrating both Head RGB and sound attains the optimal result of 99.03%. These results show the importance of auditory information in task planning, while visual input from Head RGB is supplementary.

Effect of Visual Modality for Navigate Skill. We evaluate the impact of different visual modalities, including head RGB and head depth for the Navigate skill (see Table 1c). Head-depth-only input performs worst, because depth carries no semantic texture and cannot be aligned with sound semantics; Head RGB alone performs better; and combining Head RGB with Head depth gives the best overall performance. This combination leverages complementary modalities: RGB provides sound-aligned semantics, while depth yields precise geometry.

Effect of Input Modality for Pick Skill and Open Washing Machine Skill. In Table 1d, integrating head and arm depth maximizes performance for both Pick (96.59%) and Open Washing Machine (70.27%) skills. Single-view input underperforms: for Pick, head depth alone (92.70%) suffers from manipulation blind spots, while arm depth (92.94%) lacks broader spatial awareness. The two views are complementary, fusing precise local geometry from the arm with global context from the head.

7 Conclusion

In this work, we introduce Sound-Triggered Mobile Manipulation (STMM), a new paradigm in which robots respond to sound events by physically interacting with the corresponding sound-emitting objects. To support this paradigm, we develop Habitat-Echo, a simulation platform integrating sound rendering with physical interaction, and establish a hierarchical baseline comprising a high-level task planner and low-level policies. Experiments show that the baseline localizes and interacts with sound sources in the single-event setting. In the multi-event setting, the baseline shows the feasibility of sequential interaction with multiple sound sources according to a prescribed priority, providing a starting point for further research on STMM.

8 AI USE STATEMENT

In this work, we used generative AI tools (*e.g.*, ChatGPT) to edit the research paper to improve readability, specifically for grammar and spelling checks. We have not used generative AI tools for creating scientific figures or any plagiarism. We have carefully reviewed all AI-assisted text revisions to ensure their accuracy. We take full responsibility for the final content of this work.

REFERENCES

Sherif I Abdelmaksoud, Mohammed H Al-Mola, Ghulam E Mustafa Abro, and Vijanth S Asirvadam. In-depth review of advanced control strategies and cutting-edge trends in robot manipulators: analyzing the latest developments and techniques. *IEEE Access*, 12:47672–47701, 2024.

Shuanghao Bai, Wenxuan Song, Jiayi Chen, Yuheng Ji, Zhide Zhong, Jin Yang, Han Zhao, Wanqi Zhou, Wei Zhao, Zhe Li, et al. Towards a unified understanding of robot manipulation: A comprehensive survey. *arXiv*, 2025.

Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, pp. 17–36, 2020.

Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *CVPR*, pp. 15516–15525, 2021.

Changan Chen, Carl Schissler, Sanchit Garg, Philip Kobernik, Alexander Clegg, Paul Calamia, Dhruv Batra, Philip Robinson, and Kristen Grauman. Soundspaces 2.0: A simulation platform for visual-acoustic learning. *NeurIPS*, 35:8896–8911, 2022.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, et al. Abot-m0. 5: Unified mobility-and-manipulation world action model. *arXiv*, 2026.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction, 2026.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv*, 2024.

Adrià López Escoriza, Nicklas Hansen, Stone Tao, Tongzhou Mu, and Hao Su. Multi-stage manipulation with demonstration-augmented reward, policy, and world model learning. In *ICML*, 2025.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. In *ICLR*, volume 2025, pp. 57607–57624, 2025.

Yunhai Feng, Jiaming Han, Zhuoran Yang, Xiangyu Yue, Sergey Levine, and Jianlan Luo. Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation. *arXiv*, 2025.

Jiayuan Gu, Devendra Singh Chaplot, Hao Su, and Jitendra Malik. Multi-skill mobile manipulation for object rearrangement. In *ICLR*, 2023.

Heyu Guo, Shanmu Wang, Ruichun Ma, Shiqi Jiang, Yasaman Ghasempour, Omid Abari, Baining Guo, and Lili Qiu. Omnivla: Physically-grounded multimodal vla with unified multi-sensor perception for robotic manipulation. *ICRA*, 2026.

Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. Roboground: Robotic manipulation with grounded vision-language priors. In *CVPR*, pp. 22540–22550, 2025.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *CVPR*, pp. 1724–1734, 2025.

Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *ICRA*, pp. 1470–1477, 2011.

Yash Kant, Arun Ramachandran, Sriram Yenamandra, Igor Gilitschenski, Dhruv Batra, Andrew Szot, and Harsh Agrawal. Housekeep: Tidying virtual households using commonsense reasoning. In *ECCV*, pp. 355–373, 2022.

Michael Katz, Harsha Kokel, Kavitha Srinivas, and Shirin Sohrabi. Thought of search: Planning with language models through the lens of efficiency. *NeurIPS*, 37:138491–138568, 2024.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv*, 2014.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv*, 2017.

Haru Kondoh and Asako Kanezaki. Embodied navigation with auxiliary task of action description prediction. In *ICCV*, pp. 7025–7036, 2025.

Shanshan Li, Jiawei Hou, Da Huang, Yanwei Fu, and Xiangyang Xue. Ali-ui: Enhancing complex vision-language navigation with alignment of unified map and instruction parsing. In *ACM MM*, pp. 3913–3922, 2025a.

Xiaofan Li, Chenming Wu, Zhao Yang, Zhihao Xu, Yumeng Zhang, Dingkang Liang, Ji Wan, and Jun Wang. Driverse: Navigation world model for driving simulation via multimodal trajectory prompting and motion alignment. In *ACM MM*, pp. 9753–9762, 2025b.

Sihao Lin, Zerui Li, Xunyi Zhao, Gengze Zhou, Liuyi Wang, Rong Wei, Rui Tang, Juncheng Li, Hanqing Wang, Jiangmiao Pang, et al. Vlnverse: A benchmark for vision-language navigation with versatile, embodied, realistic simulation and evaluation. *arXiv*, 2025.

Peiqi Liu, Yaswanth Orru, Jay Vakil, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv*, 2024.

Shubo Liu, Hongsheng Zhang, Qian Qiao, Qi Wu, and Peng Wang. Vln-chenv: Vision-language navigation in changeable environments. In *ACM MM*, pp. 3798–3807, 2025a.

Yanming Liu, Xinyue Peng, Jiannan Cao, Yuwei Zhang, Xuhong Zhang, Sheng Cheng, Xun Wang, Jianwei Yin, and Tianyu Du. Tool-planner: Task planning with clusters across multiple tools. *ICLR*, 2025b.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv*, 2021.

OpenAI. GPT-4V(ision) System Card. `https://openai.com/index/gpt-4v-system-card/`, September 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024, 2024.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, Vladimir Vondrus, Théophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *ICLR*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, 2017.

Rutav Shah, Albert Yu, Yifeng Zhu, Yuke Zhu, and Roberto Martín-Martín. Bumble: Unifying reasoning and acting with vision-language models for building-wide mobile manipulation. In *ICRA*, pp. 13337–13345, 2025.

Arth Shukla, Stone Tao, and Hao Su. Maniskill-hab: A benchmark for low-level manipulation in home rearrangement tasks. In *ICLR*, 2024.

Mingchen Song, Xiang Deng, Zhiling Zhou, Jie Wei, Weili Guan, and Liqiang Nie. A survey on diffusion policy for robotic manipulation: Taxonomy, analysis, and future directions. *Authorea*, 2025a.

Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *CVPR*, pp. 12078–12088, 2025b.

Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart Russell, and Pieter Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *ICRA*, pp. 639–646, 2014.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John M. Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel X. Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeurIPS*, pp. 251–266, 2021.

Marc Toussaint. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *IJCAI*, pp. 1930–1936, 2015.

Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *IEEE RAL*, 2024.

Fangyuan Wang, Shipeng Lyu, Peng Zhou, Anqing Duan, Guodong Guo, and David Navarro-Alarcon. Instruction-augmented long-horizon planning: Embedding grounding mechanisms in embodied mobile manipulation. In *AAAI*, number 14, pp. 14690–14698, 2025a.

Hanqing Wang, Wei Liang, Luc V Gool, and Wenguan Wang. Towards versatile embodied navigation. *NeurIPS*, 35:36858–36874, 2022a.

Renhao Wang, Haoran Geng, Tingle Li, Feishi Wang, Gopala Anumanchipalli, Trevor Darrell, Boyi Li, Pieter Abbeel, Jitendra Malik, and Alexei A Efros. The sound of simulation: Learning multimodal sim-to-real robot policies with generative audio. *CoRL*, 2025b.

Siyin Wang, Jinlan Fu, Feihong Liu, Xinzhe He, Huangxuan Wu, Junhao Shi, Kexin Huang, Zhaoye Fei, Jingjing Gong, Zuxuan Wu, et al. Roboomni: Proactive robot manipulation in omni-modal context. *ICLR*, 2026.

Weiqi Wang, Limeng Cui, Xin Liu, Sreyashi Nag, Wenju Xu, Chen Luo, Sheikh Muhammad Sarwar, Yang Li, Hansu Gu, Hui Liu, et al. Ecomscriptbench: A multi-task benchmark for e-commerce script planning via step-wise intention-driven product association. In *ACL*, pp. 1–22, 2025c.

Xiaohan Wang, Linchao Zhu, Zhedong Zheng, Mingliang Xu, and Yi Yang. Align and tell: Boosting text-video retrieval with local alignment and fine-grained supervision. *IEEE Transactions on Multimedia*, 25:6079–6089, 2022b.

Zhenyu Wu, Yuheng Zhou, Xiuwei Xu, Ziwei Wang, and Haibin Yan. Momanipvla: Transferring vision-language-action models for general mobile manipulation. In *CVPR*, pp. 1714–1723, 2025.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2. 5-omni technical report. *arXiv*, 2025a.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv*, 2025b.

Zhijie Yan, Shufei Li, Zuoxu Wang, Lixiu Wu, Han Wang, Jun Zhu, Lijiang Chen, and Jihong Liu. Dynamic open-vocabulary 3d scene graphs for long-term language-guided mobile manipulation. *IEEE RAL*, 2025.

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, et al. Homerobot: Open-vocabulary mobile manipulation. *arXiv*, 2023.

Abdelrahman Younes, Daniel Honerkamp, Tim Welschehold, and Abhinav Valada. Catch me if you hear me: Audio-visual navigation in complex unmapped environments with moving sounds. *IEEE RAL*, 8(2):928–935, 2023.

Songsong Yu, Yuxin Chen, Hao Ju, Lianjie Jia, Fuxi Zhang, Shaofei Huang, Yuhan Wu, Rundi Cui, Binghao Ran, Zaibin Zhang, et al. How far are vlms from visual spatial intelligence? a benchmark-driven perspective. *arXiv*, 2025.

Yinfeng Yu, Wenbing Huang, Fuchun Sun, Changan Chen, Yikai Wang, and Xiaohong Liu. Sound adversarial audio-visual navigation. *arXiv*, 2022.

Pingrui Zhang, Xianqiang Gao, Yuhan Wu, Kehui Liu, Dong Wang, Zhigang Wang, Bin Zhao, Yan Ding, and Xuelong Li. Moma-kitchen: A 100k+ benchmark for affordance-grounded last-mile navigation in mobile manipulation. *arXiv*, 2025.

Peiyuan Zhi, Zhiyuan Zhang, Yu Zhao, Muzhi Han, Zeyu Zhang, Zhitian Li, Ziyuan Jiao, Baoxiong Jia, and Siyuan Huang. Closed-loop open-vocabulary mobile manipulation with gpt-4v. In *ICRA*, pp. 4761–4767, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, pp. 2165–2183, 2023.

A APPENDIX

CONTENTS

A.1 Additional Related Work 14
A.2 Prompt for Task Planning 15
A.3 Additional Ablation Study 17
A.4 Additional Qualitative Results 17
A.5 Execution Space and Termination Criteria of Skills 18
A.6 Detailed Numerical Results on STMM-12k 19
A.7 Details of the Low-Level Policy and the Setting-Specific Planner (SSP) 20
A.8 RIRs for Sound Rendering 22
A.9 Limitation 22

A.1 ADDITIONAL RELATED WORK

Mobile Manipulation. Mobile manipulation extends the robot's workspace from confined tabletops to large-scale environments, requiring the synergy of mobility and interaction. Existing methods fall into two primary streams. The first stream focuses on object rearrangement in structured environments. (Szot et al., 2021) and (Shukla et al., 2024) develop high-fidelity simulators to support complex physical training and evaluation. Building on such platforms, (Gu et al., 2023) proposes a hierarchical framework sequencing navigation and interaction skills, while (Kant et al., 2022) integrates commonsense reasoning to infer logical object placements. The second stream targets open-vocabulary manipulation via VLMs. (Zitkovich et al., 2023) transfers web-scale knowledge to robotic control for generalization. To handle unseen objects, (Yenamandra et al., 2023) establishes a rigorous benchmark, and (Liu et al., 2024) achieves zero-shot performance by combining VLMs with navigation primitives. However, these methods remain predominantly instruction-driven. Relying on predefined textual commands limits the ability of robots to automatically process and actively respond to dynamic events in the environment.

Task Planning. Recent LLM advancements greatly broaden the scope of task planning. Specifically, Tool-Planner (Liu et al., 2025b) minimizes error propagation in tool usage by effectively grouping and selecting APIs. In algorithmic planning, (Katz et al., 2024) boosts planning efficiency while maintaining logical rigor through optimized tree-based methods. For web agents, (Wang et al., 2025c) proposes a benchmark evaluating LLMs' ability to translate user intentions into step-by-step product associations. Task planning in robotic manipulation bridges high-level reasoning with low-level execution. Early approaches (Kaelbling & Lozano-Pérez, 2011; Toussaint, 2015; Srivastava et al., 2014) primarily relied on predefined symbolic representations for continuous control. Recent research leverages the extensive world knowledge within LLMs and VLMs for planning and can be broadly categorized into two streams The first stream focuses on task decomposition. Wake *et al*. (Wake et al., 2024) utilize GPT-4V (OpenAI, 2023) to ground spatial-temporal information within the demonstration video and analyze affordances for plan generation. RoboBrain (Ji et al., 2025) unifies planning with affordance perception and trajectory prediction within a single model, enhancing manipulation capabilities. The second stream emphasizes feedback-based refinement. Zhi *et al*. (Zhi et al., 2025) optimize code-based plans using execution feedback, and Feng *et al*. (Feng et al., 2025) improve plans by anticipating future states. However, these works primarily rely on visual and textual inputs, neglecting the importance of sound signals.

Omni-Modal Large Language Models (Omni-MLLMs) Recent advancements in Omni-MLLMs have transitioned from cascaded pipelines to end-to-end architectures, broadly categorized into three paradigms: *Adapted End-to-End Speech LLMs:* Models like Qwen2.5-Omni (Xu et al., 2025a) and Llama-Omni (Fang et al., 2025) align robust text backbones with speech encoders. By bypassing intermediate text translation, they retain critical acoustic nuances (e.g., tone, emotion) and significantly reduce response latency. *Streaming Omni Models:* Architectures such as MiniCPM-o (Cui et al., 2026) and Moshi (Défossez et al., 2024) are designed for full-duplex processing of parallel

streams. They excel in reactive environments by dynamically handling continuous auditory inputs and sudden user interruptions. *Native Omni-Modal Foundation Models:* Qwen3-Omni (Xu et al., 2025b) tokenizes audio, text, and visual data into a unified representation space from the very first layer, maximizing complex cross-modal reasoning without internal translation bottlenecks. Current omni-MLLMs excel at processing human speech. However, omni-MLLMs show limited generalization to natural environmental sounds, presenting a significant challenge for comprehensive task planning in STMM.

A.2 Prompt for Task Planning

To ensure a fair comparison, we utilize a standardized prompt structure across all task planners. The input consists of two components: a system prompt and a user prompt. The system prompt defines the detailed rules and logic for task planning, while the user prompt provides the environmental context, including visual and acoustic observations.

A.2.1 Prompt for Task Planning in the Single-event Setting.

The system prompt for the single-event setting is structured into four components. The first component defines the strictly allowed skill vocabulary. The second establishes the logical mapping between sound and the corresponding skill chains. The third specifies the planning constraints and the mandatory JSON output format. Finally, the fourth component provides a structural template for the response. Complementing this, the user prompt supplies the visual and acoustic observations to ground the planning process.

(I) Prompt for Task Planning in the Single-event Setting

System Prompt: You are an expert Embodied AI Task Planner. Your goal is to generate a strictly ordered list of skills based on the input auditory (sound) and visual (head_rgb) observations.
1. ALLOWED SKILL VOCABULARY: You are strictly limited to ONLY the following skills. Do not invent new words or change spellings: [“nav”, “pick”, “place”, “open_door”, “close_sink”, “open_air_fryer”, “open_dishwasher”, “open_microwave”, “open_refrigerator”, “open_washing_machine”].
2. LOGIC MAPPING (AUDIO → SKILL): Analyze the input audio sound and classify the source object to determine the skill sequence:

1. Category A: Relocation Objects
   (a) If the sound source is: Mechanical Alarm, Phone (ringing tone), Furby (toy), Talkie, Switch, Radio, Music Box, Bluetooth speaker, Earbuds, Watch, or Tablet. REQUIRED PLAN: [“nav”, “pick”, “place”]
2. Category B: State Transition
   (a) If the sound source is: Doorbell. REQUIRED PLAN: [“nav”, “open_door”]
   (b) If the sound source is: Water Running. REQUIRED PLAN: [“nav”, “close_sink”]
   (c) If the sound source is: Air Fryer Beeping. REQUIRED PLAN: [“nav”, “open_air_fryer”]
   (d) If the sound source is: Dishwasher Running. REQUIRED PLAN: [“nav”, “open_dishwasher”]
   (e) If the sound source is: Microwave Beeping. REQUIRED PLAN: [“nav”, “open_microwave”]
   (f) If the sound source is: Refrigerator Alarm. REQUIRED PLAN: [“nav”, “open_refrigerator”]
   (g) If the sound source is: Washing Machine Running. REQUIRED PLAN: [“nav”, “open_washing_machine”]

3. CONSTRAINTS and OUTPUT FORMAT:

1. Never return an empty list.

2. Output MUST be a valid JSON object containing a single key "plan".
3. Do not include markdown formatting (like ``` json), just the raw JSON string.

4. OUTPUT FORMAT: {"plan": ["skill_1", "skill_2", ...]}

User Prompt:

1. IMAGE INPUT
2. AUDIO INPUT
3. "Input observation provided. Based on the audio class and visual context, output the execution plan JSON."

A.2.2 PROMPT FOR TASK PLANNING IN THE MULTI-EVENT SETTING.

For the multi-event setting, we organize the system prompt into five segments to handle the dual-source complexity. The initial segment lists the permissible skill vocabulary. The second details the correspondence between sound categories and their required skill sequences. Crucially, the third segment enforces interaction priority. We explicitly identify the primary sound source, requiring the planner to deduce the secondary source from the mixed audio and generate a sequential execution plan for both. The fourth and fifth segments impose strict output constraints and provide a valid JSON template to ensure parsing reliability. Finally, the user prompt provides the model with multimodal context, including visual frames and sound data.

(II) Prompt for Task Planning in the Multi-event Setting

System Prompt: You are an expert Embodied AI Task Planner. Your goal is to generate a strictly ordered list of skills based on the input auditory and visual observations. There are exactly two sound sources detected.
1. ALLOWED SKILL VOCABULARY (STRICT): You are strictly limited to ONLY the following skills: ["nav", "pick", "place", "open_door", "close_sink", "open_air_fryer", "open_dishwasher", "open_microwave", "open_refrigerator", "open_washing_machine"]. Do not invent new words or change spellings.
2. LOGIC MAPPING (AUDIO → SKILL): Analyze the input audio sound to determine the action sequence:

1. Category A: Relocation Objects
   (a) If the sound source is: Mechanical Alarm, Phone (ringing tone), Furby (toy), Talkie, Switch, Radio, Music Box, Bluetooth speaker, Earbuds, Watch, or Tablet. REQUIRED PLAN: ["nav", "pick", "place"]
2. Category B: State Transition
   (a) If the sound source is: Doorbell. REQUIRED PLAN: ["nav", "open_door"]
   (b) If the sound source is: Water Running. REQUIRED PLAN: ["nav", "close_sink"]
   (c) If the sound source is: Air Fryer Beeping. REQUIRED PLAN: ["nav", "open_air_fryer"]
   (d) If the sound source is: Dishwasher Running. REQUIRED PLAN: ["nav", "open_dishwasher"]
   (e) If the sound source is: Microwave Beeping. REQUIRED PLAN: ["nav", "open_microwave"]
   (f) If the sound source is: Refrigerator Alarm. REQUIRED PLAN: ["nav", "open_refrigerator"]
   (g) If the sound source is: Washing Machine Running. REQUIRED PLAN: ["nav", "open_washing_machine"]

3. OBJECT MAPPING INSTRUCTION (CRITICAL): You are provided with an event-priority specification: One of the sound sources has a high priority. You have to interact with "{obj_1}" first. Your task is to identify the other sound source from the audio and generate the plan for the whole process. You MUST map the objects exactly as follows:

1. Key "first_sound": Generate the plan for "{obj_1}".
2. Key "second_sound": Identify the SECOND sound source from the audio (the one that is NOT "{obj_1}") and generate its plan.
3. Do NOT output plans for any other potential objects unless they are detected as the second sound.

4. CONSTRAINTS and OUTPUT FORMAT:

1. Output MUST be a valid JSON object containing a single root key "plan".
2. The keys inside "plan" MUST be exactly "first_sound" and "second_sound".
3. Do not include markdown formatting (like ``` json), just the raw JSON string.

5. Expected JSON Structure: { "plan": {
"first_sound": ["skill_1", "skill_2", ...], # ← Actions for the object assigned to 'first_sound'
"second_sound": ["skill_1", "skill_2", ...] # ← Actions for the object assigned to 'second_sound'
} }

User Prompt:

1. `Image Input`
2. `Audio Input`
3. "Input observation provided. Based on the audio class and visual context, output the execution plan JSON."

A.3 Additional Ablation Study

We evaluate the impact of different input modalities for the Setting-Specific Planner (SSP) in the multi-event setting (see Table 2). In the multi-event setting, combining head RGB and sound brings more distractions for task planning than sound alone, unlike in the single-event setting.

Table 2: Ablation study on input modalities for Setting-Specific Planner (SSP) in the multi-event setting.

Model	Input Modality		S.R. (↑)
	Head RGB	Sound	
I	✓	✗	6.25
II	✓	✓	68.25
III	✗	✓	**69.50**

A.4 Additional Qualitative Result

We present qualitative visualization results for the four manipulation skills in Figure 7. Figure 7a and Figure 7b illustrate the Pick skill. In the success case (a), the robot grasps the object and correctly returns the arm to the resting position, while in the failure case (b), the arm fails to retract to the standard pose despite holding the object. The Place skill is depicted in Figure 7c and d. In Figure 7c, the object is stably placed on the table, whereas in (d), the object is released with an unstable pose and falls off the surface of the small table. Figure 7e and f visualize the Open Door skill. In the success scenario (e), the arm pulls the door open and resets to the resting position, but in the failure scenario Figure 7f, the arm remains extended after opening the door. Finally, Figure 7g and h show the Close Sink skill, where the height of the blue water column visually correlates with

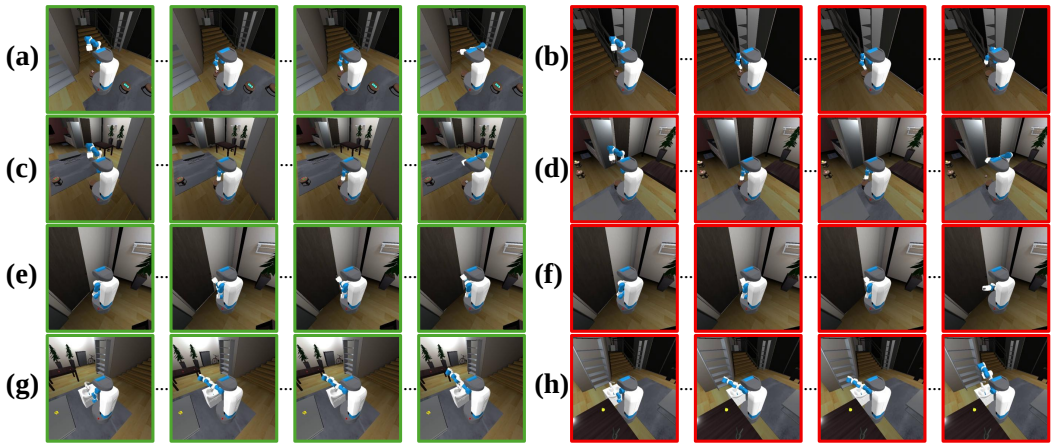


Figure 7: **Additional Qualitative Results of Different Skills.** (a), (b) Success and failure cases of the Pick skill. (c), (d) Success and failure cases of the Place skill. (e), (f) Success and failure cases of the Open Door skill. (g), (h) Success and failure cases of the Close Sink skill.

the faucet handle angle. In Figure 7g, the faucet is successfully closed, while in Figure 7h, the arm accidentally collides with the handle, causing the opening angle to increase rather than decrease. *More visualizations can be seen in the attached videos.* We provide video demos for different settings (use headphones for a better experience).

A.5 Execution Space and Termination Criteria of Skill

Similar to previous work Gu et al. (2023); Shukla et al. (2024); Wang et al. (2025a), the proposed tasks are decomposed into different subtasks. Each subtask is solved by an individual skill. A task is considered completed successfully if and only if all constituent skills are successfully executed in sequence. Skill spaces are shown below.

Navigate. The action space consists of four actions: `MoveForward`, `TurnLeft`, `TurnRight`, and `Stop`. Specifically, `MoveForward` translates the robot by 0.5m along its local forward axis, while rotational actions (`TurnLeft`, `TurnRight`) execute 90° in-place rotations to the left and right, respectively. The `Stop` action terminates the execution of navigation without changing the robot's pose and location. An episode is considered successful if and only if the robot executes `Stop` while positioned within a region of 0.5m radius of the goal.

Pick. The robot operates in an 8-DoF continuous action space. The first seven dimensions control the arm's joint angles, driving the end-effector to the target pose via the kinematic chain Abdelmaksoud et al. (2024), while the final dimension modulates the gripper state. A Pick skill is considered successful with two conditions: (1) Grasping: The gripper action is positive while an object is within a 15cm snapping threshold of the end-effector, triggering a stable grasp. (2) Retraction: The robot effectively holds the object and returns the arm to a predefined resting pose, ensuring a smooth transition to subsequent subtasks.

Place. The robot operates in an 8-DoF action space identical to that of the Pick skill. The success conditions for the Place skill are twofold: (1) Releasement: The gripper opens to release the object, ensuring it settles within 15cm of the target position. (2) Retraction: Upon release, the arm reverts to the default resting pose, preparing for the next subtask.

Open Door. Operating within the 8-DoF action space, the robot's end-effector attaches to the handle of the door and drives it along its own functional trajectory. The skill is considered successful with two conditions: (1) Opening: While connected with the door handle, the robot must drive the door's joint angle to a specified target threshold. (2) Retraction: The robot disengages from the door handle and returns the arm to the resting pose.

Close Sink. Similar to the Open Door skill, the robot utilizes the 8-DoF action space to manipulate the sink's rotary mechanism. The end-effector must grasp the faucet handle and apply torque to

rotate it along its functional axis. Success is determined by two conditions: (1) Closure: While maintaining grasp, the robot must rotate the handle to the zero-degree threshold within a tolerance of 0.2 rad. (2) Retraction: Upon completing the rotation, the robot releases the handle and reverts the arm to the resting pose.

Open Dishwasher. Similar to the Open Door skill, the robot uses the 8-DoF action space to rotate the dishwasher's bottom-hinged door about its hinge axis. The end-effector grasps the outer door panel and drives the door along its downward functional trajectory. Success is determined by two conditions: (1) Opening: While maintaining grasp, the robot must rotate the door to $0.483\,\mathrm{rad}$ within a tolerance of $0.05\,\mathrm{rad}$. (2) Retraction: Upon completing the rotation, the robot releases the door and reverts the arm to the resting pose.

Open Microwave. Operating within the 8-DoF action space, the robot's end-effector grasps the microwave's side-hinged door and rotates it about the hinge. The skill is considered successful with two conditions: (1) Opening: While connected with the door, the robot must drive the joint angle to $0.507\,\mathrm{rad}$ within a tolerance of $0.15\,\mathrm{rad}$. (2) Retraction: The robot disengages from the door and returns the arm to the resting pose.

Open Air-Fryer. Unlike the hinged-door skills, this skill actuates a prismatic drawer in the same 8-DoF action space. The end-effector grasps the drawer front and translates it along the slide rail. The skill is considered successful with two conditions: (1) Opening: While maintaining grasp, the robot must pull the drawer to the specified target of $0.073\,\mathrm{m}$ within a tolerance of $0.025\,\mathrm{m}$. (2) Retraction: The robot releases the drawer and returns the arm to the resting pose.

Open Washing Machine. The robot utilizes the 8-DoF action space to open the washing machine's front-loading circular door. The end-effector grasps the outer door panel and applies torque about the hinge. Success is determined by two conditions: (1) Opening: While maintaining grasp, the robot must rotate the door to $0.507\,\mathrm{rad}$ within a tolerance of $0.05\,\mathrm{rad}$. (2) Retraction: Upon completing the rotation, the robot releases the door and reverts the arm to the resting pose.

Open Refrigerator. Operating within the 8-DoF action space, the robot's end-effector grasps the refrigerator's right-hand side-hinged door and drives it along its functional trajectory. The skill is considered successful with two conditions: (1) Opening: While connected with the door, the robot must drive the joint angle to $0.507\,\mathrm{rad}$ within a tolerance of $0.05\,\mathrm{rad}$. (2) Retraction: The robot disengages from the door and returns the arm to the resting pose.

A.6 Detailed Numerical Results on the STMM-12K

A.6.1 Evaluation Metrics

To rigorously evaluate the robustness of policy models in Sound-Triggered Mobile Manipulation (STMM), we introduce the completion success rate (see Figure 5). A complete task episode in STMM decomposes into an initial high-level task planning phase followed by a sequence of K low-level skills executed by different policy models, denoted as $S = \{S_1, S_2, \ldots, S_K\}$. The evaluation strictly follows a cascaded paradigm, where a failure at any stage terminates the episode immediately. Let N denote the total number of evaluation episodes. The evaluation formulation is defined as follows:

- **Task Planning Success Rate (SR_{plan}):** The execution sequence initiates with the task planning phase ($k = 0$). The planning success rate is defined as:

$$SR_{\text{plan}} = \frac{N_{\text{plan}}}{N}, \tag{4}$$

  where N_{plan} is the number of episodes in which the task planner correctly identifies the requisite skill chain.

- **Success Rate of Individual Skill Completion:** To isolate the policy's inherent capability for each skill without the influence of cascading errors, we define the success rate of individual skill completion. This evaluates the execution of skill S_k in isolation, decoupled from the task planner and preceding skills. Let N_{S_k} denote the total number of independent trials initialized specifically for skill S_k, and $N_{S_k}^{\text{success}}$ denote the number of successful

Table 3: **Numerical Results on the STMM-12k.** We report the success rate (%) of task planning (SR_{plan}), individual skill completion (Individual), and progressive completion success rate of different skills across different settings. The higher the better. *(a)* **Evaluation of object relocation**. *(b)* **Evaluation of state transition**. *(c)* **Evaluation of multi-event setting**. Success rates are reported in the format "First Sound Source / Second Sound Source". * denotes the target position that is provided for the navigate policy model.

(a)

Method	SR_{plan}	Navigate	Pick	Place	CSR
Individual	N/A	81.02	96.59	91.97	N/A
Oracle	100.00	81.02	95.50	92.77	71.78
Oracle*	100.00	95.13	96.93	93.40	86.13
Qwen2.5-Omni-3B	1.46	83.33	100.00	80.00	0.97
Qwen2.5-Omni-7B	2.68	90.91	100.00	90.00	2.19
MiniCPM-o-4.5	13.38	80.00	97.73	88.37	9.25
Qwen3-Omni-30B	88.56	81.32	95.27	92.55	63.50
SSP	99.03	80.84	95.74	92.70	**71.05**
Qwen2.5-Omni-3B*	1.46	83.33	100.00	100.00	1.22
Qwen2.5-Omni-7B*	2.68	90.91	100.00	100.00	2.43
MiniCPM-o-4.5*	13.38	96.36	96.23	90.20	11.19
Qwen3-Omni-30B*	88.56	95.05	96.53	92.81	75.43
SSP*	99.03	95.09	96.90	93.33	**85.16**

(b)

Method	SR_{plan}	Navigate	Interaction	CSR
Individual	N/A	44.20	94.20	N/A
Oracle	100.00	44.20	65.25	28.84
Oracle*	100.00	49.49	61.38	30.38
Qwen2.5-Omni-3B	12.29	62.50	53.33	4.10
Qwen2.5-Omni-7B	31.91	56.15	52.38	9.39
MiniCPM-o-4.5	41.47	48.97	59.66	12.12
Qwen3-Omni-30B	39.42	48.05	64.86	12.29
SSP	99.49	44.25	65.12	**28.67**
Qwen2.5-Omni-3B*	12.29	69.44	46.00	3.92
Qwen2.5-Omni-7B*	31.91	58.82	53.64	10.07
MiniCPM-o-4.5*	41.47	53.50	58.46	12.97
Qwen3-Omni-30B*	39.42	53.68	56.45	11.95
SSP*	99.49	49.57	61.25	**30.20**

(c)

Method	SR_{plan}	Navigate	Pick	Place	Interaction	CSR
Individual	N/A	73.75 / 64.50	65.82 / 44.14	54.75 / 38.28	89.29 / 84.72	N/A
Oracle	100.00	69.25 / 39.16	73.58 / 57.50	75.69 / 91.30	93.55 / 96.00	41.50 / 11.25
Oracle*	100.00	73.75 / 63.32	82.14 / 52.44	76.81 / 86.05	93.02 / 88.64	49.75 / 19.00
Qwen2.5-Omni-3B	26.25	80.95 / 36.73	71.60 / 57.14	77.59 / 75.00	100.00 / 100.00	12.25 / 2.50
Qwen2.5-Omni-7B	44.50	75.28 / 40.28	65.65 / 48.15	80.23 / 84.62	100.00 / 100.00	18.00 / 3.25
MiniCPM-o-4.5	50.00	71.00 / 40.96	72.73 / 58.62	76.04 / 88.24	100.00 / 100.00	20.75 / 5.00
Qwen3-Omni-30B	19.25	62.34 / 50.00	60.00 / 75.00	87.50 / 100.00	87.50 / 100.00	7.00 / 3.00
SSP	69.50	68.35 / 40.19	71.76 / 58.33	72.95 / 92.86	90.00 / 94.74	**26.75 / 7.75**
Qwen2.5-Omni-3B*	26.25	79.05 / 53.70	80.77 / 41.67	77.78 / 60.00	100.00 / 80.00	13.50 / 2.50
Qwen2.5-Omni-7B*	44.50	73.60 / 60.47	78.74 / 48.00	82.00 / 79.17	100.00 / 100.00	21.50 / 5.25
MiniCPM-o-4.5*	50.00	74.50 / 68.32	81.95 / 50.00	77.98 / 86.21	100.00 / 90.91	25.25 / 8.75
Qwen3-Omni-30B*	19.25	72.73 / 72.97	66.67 / 46.15	85.71 / 100.00	92.86 / 78.57	9.25 / 4.25
SSP*	69.50	73.38 / 63.43	82.08 / 52.83	74.65 / 89.29	90.32 / 87.50	**33.50 / 13.25**

executions. The formulation is:

$$\text{Individual} = \frac{N_{S_k}^{\text{success}}}{N_{S_k}}. \tag{5}$$

This metric provides a fundamental evaluation for different policy models.

- **Progressive Skill Completion Success Rate:** The Progressive Skill Completion Success Rate at skill S_k is the fraction of episodes that successfully complete S_k among those that have successfully completed all preceding skills. It effectively captures the cascading errors inherent in long-horizon tasks and is formally defined as the joint probability across the chain of skills:

$$PSCSR = SR_{\text{plan}} \prod_{i=1}^{K} P(S_i \mid S_{1:i-1}, \text{plan}). \tag{6}$$

A.6.2 Result

We provide detailed numerical results on the STMM-12k for quantitative comparison. In addition to the numbers shown in Figure 5 of the main paper, we add the success rate of individual skill completion (see Individual in Table 3) and Progressive Skill Completion Success Rate for each skill.

A.7 Details of low-level policy and Setting-Specific Planner

A.7.1 Architecture

We provide the architecture of the low-level policies and the Setting-Specific Planner (SSP) in Figure 8. For the navigation model in the single-event setting, multi-modal observations are processed

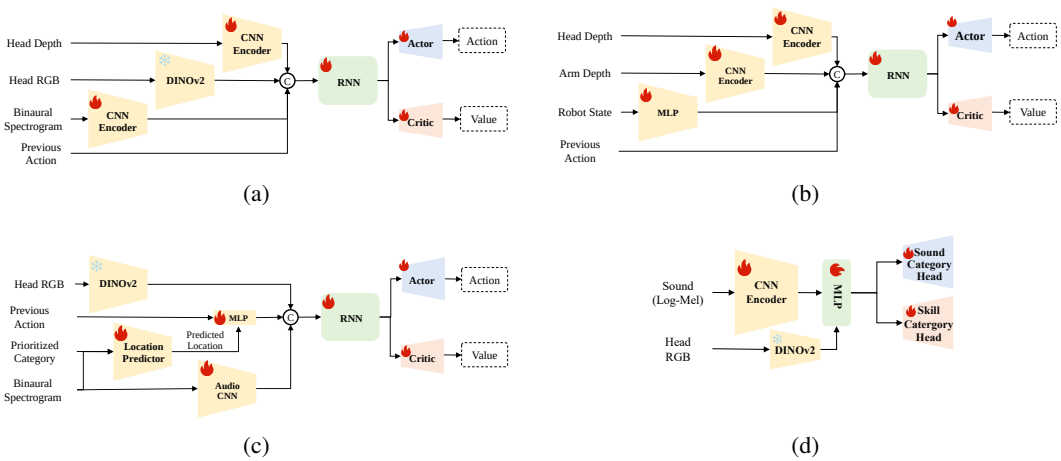


Figure 8: **Architecture of low-level policies and Setting-Specific Planner (SSP)**. (a) Navigation policy at the single-event setting. (b) Low-level policy at the state transition task. (c) Navigation policy at the multi-event setting. (d) SSP at the single-event setting.

via the corresponding encoder (see Figure 8a). The policies for interaction skills, such as Open Door, Close Sink, and Open Washing Machine, share a unified network architecture (see Figure 8b). For the navigation model in the multi-event setting, an additional location predictor is introduced (see Figure 8c). All low-level policies are trained via PPO, with the internal location predictor (see Figure 8c) trained concurrently via supervised learning. The architecture of the Setting-Specific Planner (SSP) is shown in Figure 8d, which is trained in a supervised manner. The SSP processes single-channel mel spectrum and head RGB for the single-event setting, and priority-conditioned features for the multi-event setting.

A.7.2 Reward and Loss Function

All policies are trained via PPO. Reward functions are detailed below.

Reward for Navigation Policy. We train the navigation policy using a dense progress reward, with additional penalties for time and collisions. At step t, the reward is defined as

$$r_t = -0.002 + (d_{t-1} - d_t) - \min(0.001\Delta F_t, 0.2) + 5.0\mathbb{I}_{\text{succ},t}, \tag{7}$$

where -0.002 is a slack penalty applied at each step. d_t denotes the shortest collision-safe geodesic distance from the current position to the nearest valid navigation goal at timestep t, such that progress toward the goal yields a positive reward and movement away from it incurs a penalty. $\Delta F_t = max((F_t - F_{t-1}), 0)$ denotes the increase in robot collision force, with the corresponding penalty capped at 0.2 per step. $\mathbb{I}_{\text{succ},t}$ is the success indicator, which is activated only when the policy stops within the goal region.

Reward for Low-level Policy. We train the low-level policy for the state-transition task using a phase-aware dense reward. The task consists of four phases: approaching, state transition, releasing, and returning the gripper to its resting pose. The reward at step t is defined as

$$\begin{aligned} r_t = & -0.01 + s_{\text{approaching}} \left(d_{t-1}^{\text{handle}} - d_t^{\text{handle}}\right) + \mathbb{I}_{\text{first_grasp},t} \\ & + s_{\text{transiting}} \left(d_{t-1}^{\text{joint}} - d_t^{\text{joint}}\right) + 3\,\mathbb{I}_{\text{first_goal},t} + 2\,\mathbb{I}_{\text{valid_release},t} \\ & + s_{\text{reseting}}\, \text{round}_{10^{-3}} \left(d_{t-1}^{\text{rest}} - d_t^{\text{rest}}\right) \\ & - \mathbb{I}_{\text{wrong_grasp},t} - 2\,\mathbb{I}_{\text{invalid_release},t} - 0.5\,\mathbb{I}_{\text{invalid_grasp},t} \\ & - 0.02\,\mathbb{I}\left(F_t \geq 1000\right) + 5\,\mathbb{I}_{\text{success},t}. \end{aligned} \tag{8}$$

Here, d_t^{handle} denotes the distance from the gripper to the interactive region of the articulated object, $d_t^{\text{joint}} = |\hat{q} - q_t|$ denotes the error between the target and current joint states, and d_t^{rest} denotes the distance from the gripper to its resting position. The phase-dependent factors $s_{\text{approaching}}$, $s_{\text{transiting}}$, and s_{reseting} scale the corresponding progress rewards.

The indicators $\mathbb{I}_{\text{first_grasp},t}$, $\mathbb{I}_{\text{first_goal},t}$, and $\mathbb{I}_{\text{valid_release},t}$ provide one-time bonuses for the first correct grasp, reaching the target state while maintaining the grasp, and releasing after reaching the target state, respectively. Wrong grasps, invalid releases, and invalid grasps incur penalties and terminate the episode. A collision force exceeding 1000N incurs an additional penalty. Success requires the object to reach its target state, the gripper to release it, and the gripper to return within 0.05m of its resting position.

PPO Loss. Given the task-specific reward r_t, we compute the return target $\hat{R}_t$ using Generalized Advantage Estimation (GAE) and define the advantage as

$$\hat{A}_t = \hat{R}_t - V(x_t), \tag{9}$$

where x_t is the policy input and $V(x_t)$ is the critic prediction.

The PPO loss is defined as

$$\begin{aligned} \mathcal{L}_{\text{PPO}} = & -\mathbb{E}_t \left[\min \left(\rho_t \hat{A}_t,\, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon) \hat{A}_t \right) \right] \\ & + c_v\, \mathbb{E}_t \left[\left(V(x_t) - \hat{R}_t \right)^2 \right] - \beta\, \mathbb{E}_t \left[\mathcal{H}\big(\pi(\cdot \mid x_t)\big) \right], \end{aligned} \tag{10}$$

where

$$\rho_t = \frac{\pi(a_t \mid x_t)}{\pi_{\text{old}}(a_t \mid x_t)}. \tag{11}$$

Here, the first term updates the action policy according to the advantage, the second term trains the critic to predict the reward-induced return, and the entropy term $\mathcal{H}$ encourages exploration. Different policies share the same optimization objective, while their different reward functions lead to different return targets and action updates.

A.8 RIRs for Sound Rendering

RIRs ensure two critical physical factors for sound rendering: (1) Geometry-aware propagation: Through bidirectional path tracing, RIRs capture intricate wave phenomena, including direct paths, specular reflections, and reverberation. This ensures that the rendered sounds remain consistent with both the room's static 3D geometry and the dynamic position and orientation of the robot. (2) Material-dependent scattering: RIRs account for acoustic material properties by mapping semantic annotations to frequency-dependent absorption and transmission coefficients. This enables the simulator to reflect distinct acoustic interactions with diverse surfaces.

A.9 Limitation

This initial exploration also highlights several limitations and directions for future work. Our evaluation focuses on simulation, leaving real-robot validation as an important next step. Additionally, state transitions remain challenging, as do task planning and navigation to the second sound source in multi-event tasks. Incorporating recovery behaviors into the navigation skill may help the robot recover when stuck by obstacles and continue navigating toward the second sound source.