

## A Summary of Supplementary Materials

In this supplementary materials, we provide:

1. A video demonstrating the case examples detailed in Figure 4 of the main paper is available at our webpage here. For the best viewing experience, **we recommend watching the video with headphone or a device that supports spatial audio playback.** See Section B for details.
2. Details of benchmark construction pipeline, including data processing, annotations, QA synthesis and quality review, see Section C.
3. Evaluation details of SAVVY-Bench, including open-source AV-LLMs, proprietary AV-LLMs and human evaluations, see Section D.
4. Details of input data to the pipeline, including video input settings, multi-channel audio settings (microphone configurations), as well as camera trajectory, see Section E.
5. Additional implementation details of all stages in SAVVY, see Section F.
6. Efficiency analysis of SAVVY, see Section G.
7. Additional ablation studies of SAVVY-Bench on input modalities, see Blind Testing in Section H.
8. Limitations of SAVVY, see Section I.
9. Broader impacts of the work with safeguards, see Section J.
10. Additional qualitative results which showcase the reasoning process of SAVVY as well as the error types analysis, see Section K.

## B Video Examples

The demo videos contain two case examples—one egocentric direction task and one allocentric distance task—captured in a single video clip featuring two people conversing in an indoor setting. **We recommend watching the video with headphones or a device that supports spatial audio playback.**

These examples correspond to the qualitative results presented in the main paper. In both cases, the queried event is: *confirming they have La Croix drinks*, corresponding to the spoken sentence, “Yeah, let’s see ... grab some La Croix for us,” from a guest (a male wearing a blue shirt) speaking to the camera wearer.

**Egocentric Direction Example.** The question asks for the relative direction of the other person, with options: *front-left*, *front-right*, *back-left*, or *back-right*. In this clip, the other person is not visible at any timestamp during the event, as he is located in the *back-right* quadrant relative to the camera wearer. While the direction must be inferred from spatial audio cues, a human viewer can clearly perceive the sound as coming from the back-right when watching the video with spatial audio. SAVVY correctly predicts this as *back-right*, whereas Gemini-2.5-pro incorrectly classifies it as *front-left*.

**Allocentric Distance Example.** This question asks for the distance between the two-seater dining table and the speech sound source (the male guest in the blue shirt). The table is clearly visible in several frames throughout the video. SAVVY localizes both the table and the sound source using a combination of egocentric tracks via Snapshot Descriptor, text-guided snapshot segmentation and spatial audio cues. SAVVY estimates the distance as *3.49 meters*, which is close to the ground truth of *3.82 meters*. In contrast, Gemini-2.5-pro predicts a significantly incorrect distance of *2.30 meters*.

These examples illustrate SAVVY’s robustness in both directional and quantitative spatial reasoning, especially in challenging, partially observed scenarios.

## C Benchmark Construction

We implement a four-stage pipeline to construct SAVVY-Bench. The stages are **Data Preprocessing**, **Annotation**, **QA Synthesis**, and **Quality Review**. Each stage combines automated tools with human checks to ensure that every Question–Answer (QA) pair is precise.

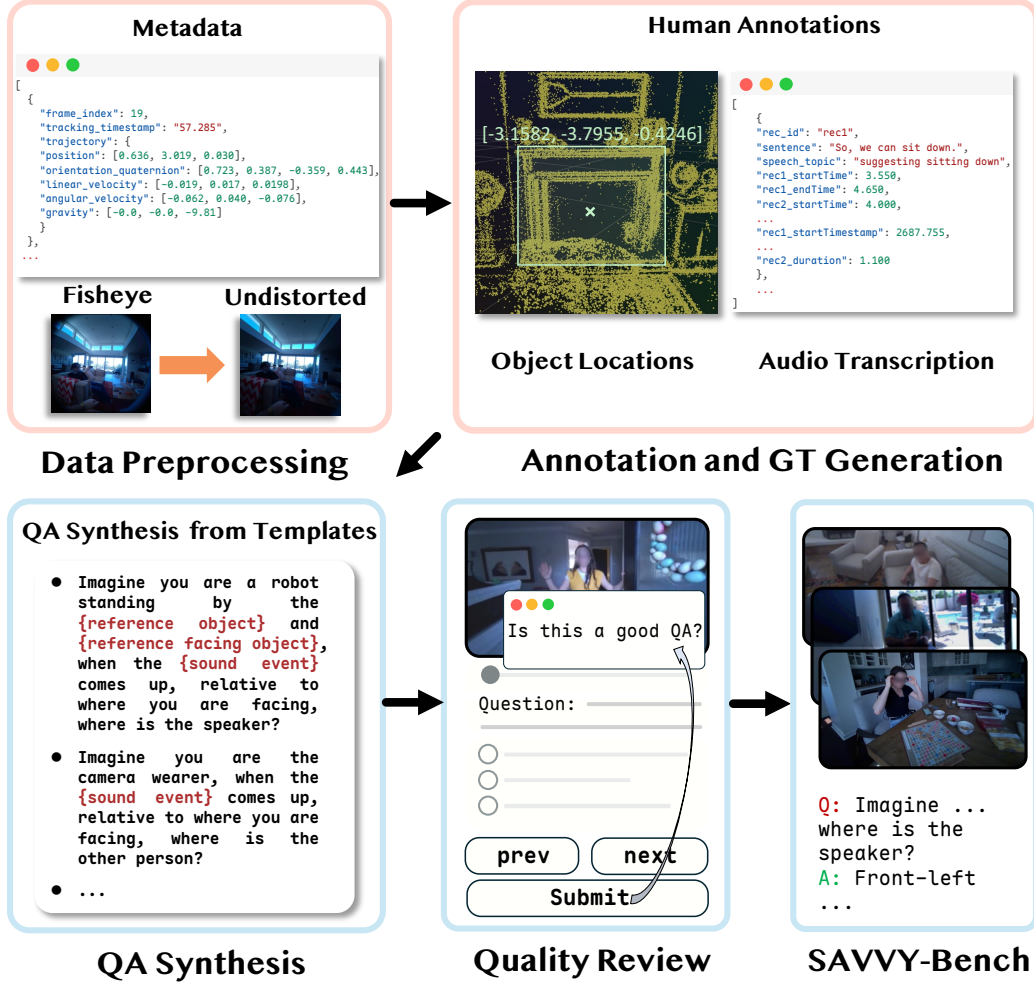


Figure 5: Human-in-the-Loop Dataset Curation and Benchmark Construction Workflow for SAVVY-Bench.

### C.1 Data Preprocessing

We preprocess the video data from the Aria Everyday Activities (AEA) Dataset [60] and integrate raw annotations—such as word-level transcriptions, camera-wearer trajectories, and other sensor signal records—into a unified metadata schema, as illustrated in Figure 5.

For video preprocessing, the original fisheye recordings are undistorted into rectilinear frames to ensure compatibility with AV-LLMs. In scenarios with two wearer-mounted camera streams, the videos are temporally aligned to form a unified timeline. This alignment supports consistent segmentation of speech into sentences and facilitates accurate speech topic extraction.

### C.2 Annotation and Ground Truth Generation

Our annotation focuses primarily on objects and events.

**Static Object Annotation.** Static objects are automatically detected using EFM3D [61] based on a predefined list of object categories (e.g., couch, fireplace). We use Vision-LLM [41] to generate an informative description phrase for each detected object. Annotators then inspect the 3D coordinates and descriptions in a point-cloud viewer, correcting any errors in location, category, or description as needed.

**Sounding Event Annotation.** For each sound event, we annotate the event description or transcription, its start and end times, and the identity and 3D location of the sound source—if the source is tied to a physical object (e.g., running water with a faucet, a thud with a door). Human annotators adjust the event time span and label the source object and its position accordingly. Specifically for speech events, we first cluster raw word-level transcripts into complete sentences. Annotators then label speech events on a sentence-by-sentence basis. A prompted, rule-based agent [41] converts these validated sentences into concise speech topics that describe individual conversational moments. The prompt design used for this process is shown in Figure 6.

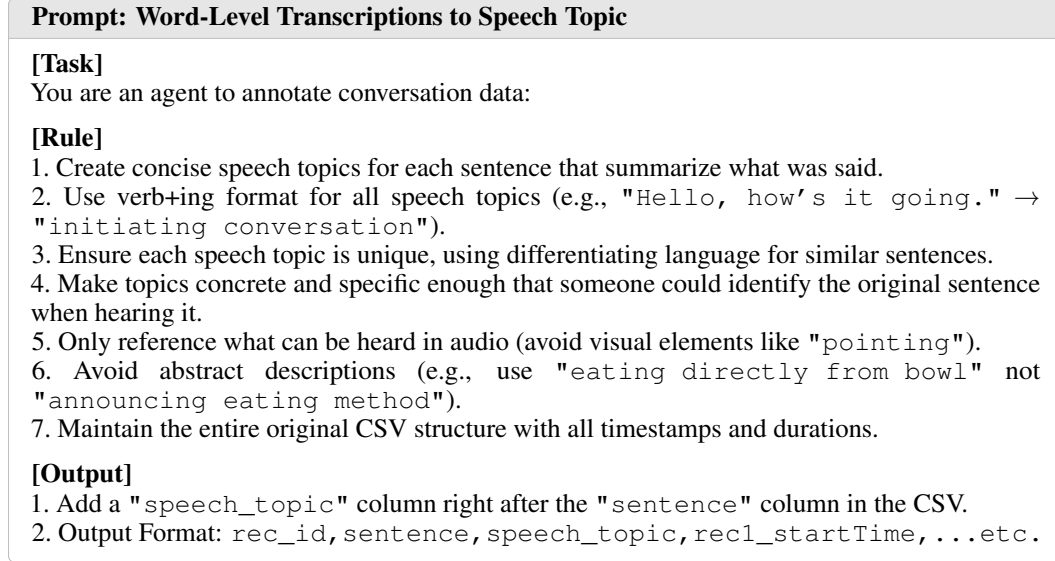


Figure 6: Prompt used to generate speech topics from word-level transcripts.

**Sound Event Annotation System and UI.** To streamline the annotation process and reduce errors, we developed a desktop annotation tool using PyQt5. This system integrates video playback, speech and non-speech event labeling, and timestamp editing in a single interface (see Figure 7). It supports dual-camera views with synchronized playback and saves annotations locally. The tool is self-contained, works offline, and requires no server backend.

**Human Annotation Guideline for Sound Events.** Annotators follow five key principles:

- 1) *Accuracy*: For speech events, correct the original word-level transcription to ensure that every spoken word and audible event is captured exactly as heard. Remove filler words and non-informative tokens, retaining only meaningful content.
- 2) *Completeness*: Label the full audible span of each event, setting start and end times as close as possible to the actual boundaries to avoid clipping or omission.
- 3) *Synchronization*: For speech events involving two participants, maintain the temporal alignment between the recordings from both devices throughout the annotation process.
- 4) *Label Uniformity*: For non-speech sound events, ensure that each description is unique, unambiguous, and consistent across the entire video.
- 5) *Language and Mechanics*: Use standard spelling, punctuation, and capitalization. Maintain consistent formatting across all annotations.

### C.3 QA Synthesis

We use template scripts to generate QA pairs for SAVVY-Bench. These scripts integrate the unified metadata (described in Section C.1) with the new annotations and ground truth data (from Section C.2) using well-defined question schemas, resulting in unambiguous and structured QA pairs.

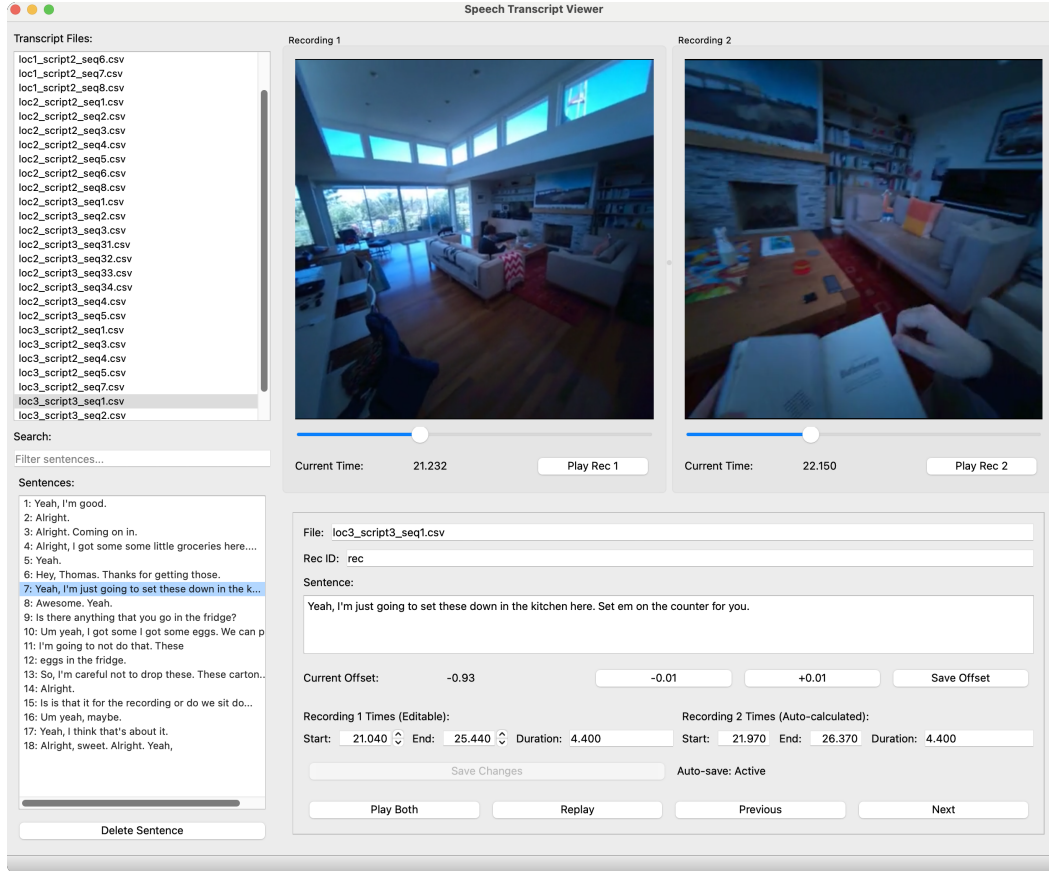


Figure 7: **Interface for sound event annotation.** The tool displays dual-camera videos with synchronized playback and saves annotations locally.

SAVVY-Bench includes six templates covering four task types: egocentric direction, egocentric distance, allocentric direction, and allocentric distance. For both egocentric and allocentric direction tasks, we design two levels of difficulty: a simple template with three options (left, right, back) and a hard template with four options (front-left, front-right, back-left, back-right).

We provide the complete set of templates for all six QA types, each specified for both speech and non-speech sound events as follows:

### Egocentric Direction - Simple

1. Imagine you are the camera wearer, when the {non-speech sound event} sound comes up, relative to where you are facing, where is the sound source: left, right, or back? If the object is generally to your left and facing it requires turning less than 120 degrees left, choose 'left'. If the object is generally to your right and facing it requires turning less than 120 degrees right, choose 'right'. If the object is generally behind you and facing it requires turning 120 degrees or more, choose 'back'.
2. Imagine you are the camera wearer, when the speech topic {speech topic} comes up, relative to where you are facing, where is the other person : left, right, or back? If the object is generally to your left and facing it requires turning less than 120 degrees left, choose 'left'. If the object is generally to your right and facing it requires turning less than 120 degrees right, choose 'right'. If the object is generally behind you and facing it requires turning 120 degrees or more, choose 'back'.

### Egocentric Direction - Hard

1. Imagine you are the camera wearer, when the {non-speech sound event} sound comes up, relative to where you are facing, where is the sound source: front-left, front-right, back-left, or back-right? The directions refer to the quadrants of a Cartesian plane (if you are standing at the origin and facing along the positive y-axis). Consider the center point location of the object as the its location.
2. Imagine you are the camera wearer, when the speech topic {speech topic} comes up, relative to where you are facing, where is the other person: front-left, front-right, back-left, or back-right? The directions refer to the quadrants of a Cartesian plane (if you are standing at the origin and facing along the positive y-axis). Consider the center point location of the object as the its location.

### Egocentric Distance

1. Imagine you are the camera wearer, when the {non-speech sound event} sound comes up, relative to where you are standing, what is the distance between you and the sound source in meters? Consider the center point location of the object as the its location. Calculate the Euclidean distance between the two points in the horizontal plane. Answer in numeric format.
2. Imagine you are the camera wearer, when the speech topic: {speech topic} comes up, relative to where you are standing, what is the distance between you and the other person in meters? Consider the center point location of the object as the its location. Calculate the Euclidean distance between the two points in the horizontal plane. Answer in numeric format.

### Allocentric Direction - Simple

1. Imagine you are a robot standing by the {reference object} white recessed fireplace and facing {facing object}, when the {non-speech sound event} sound comes up, relative to where you are facing, where is the sounding object: left, right, or back? If the object is generally to your left and facing it requires turning less than 120 degrees left, choose 'left'. If the object is generally to your right and facing it requires turning less than 120 degrees right, choose 'right'. If the object is generally behind you and facing it requires turning 120 degrees or more, choose 'back'.
2. Imagine you are a robot standing by the {reference object} and facing the {facing object}, when the speech topic: {speech topic} comes up, relative to where you are facing, where is the speaker: left, right, or back? If the object is generally to your left and facing it requires turning less than 120 degrees left, choose 'left'. If the object is generally to your right and facing it requires turning less than 120 degrees right, choose 'right'. If the object is generally behind you and facing it requires turning 120 degrees or more, choose 'back'.

### Allocentric Direction - Hard

1. Imagine you are a robot standing by the {reference object} and facing the {facing object}, when the {non-speech sound event} sound comes up, relative to where you are facing, where is the sounding object: front-left, front-right, back-left, or back-right? The directions refer to the quadrants of a Cartesian plane (if you are standing at the origin and facing along the positive y-axis). Consider the center point location of the object as the its location.
2. Imagine you are a robot standing by the {reference object} and facing the {facing object}, when the speech topic: {speech topic} comes up, relative to where you are facing, where is the speaker: front-left, front-right, back-left, or back-right? The directions refer to the quadrants of a Cartesian plane (if you are standing at the origin and facing along the positive y-axis). Consider the center point location of the object as the its location.

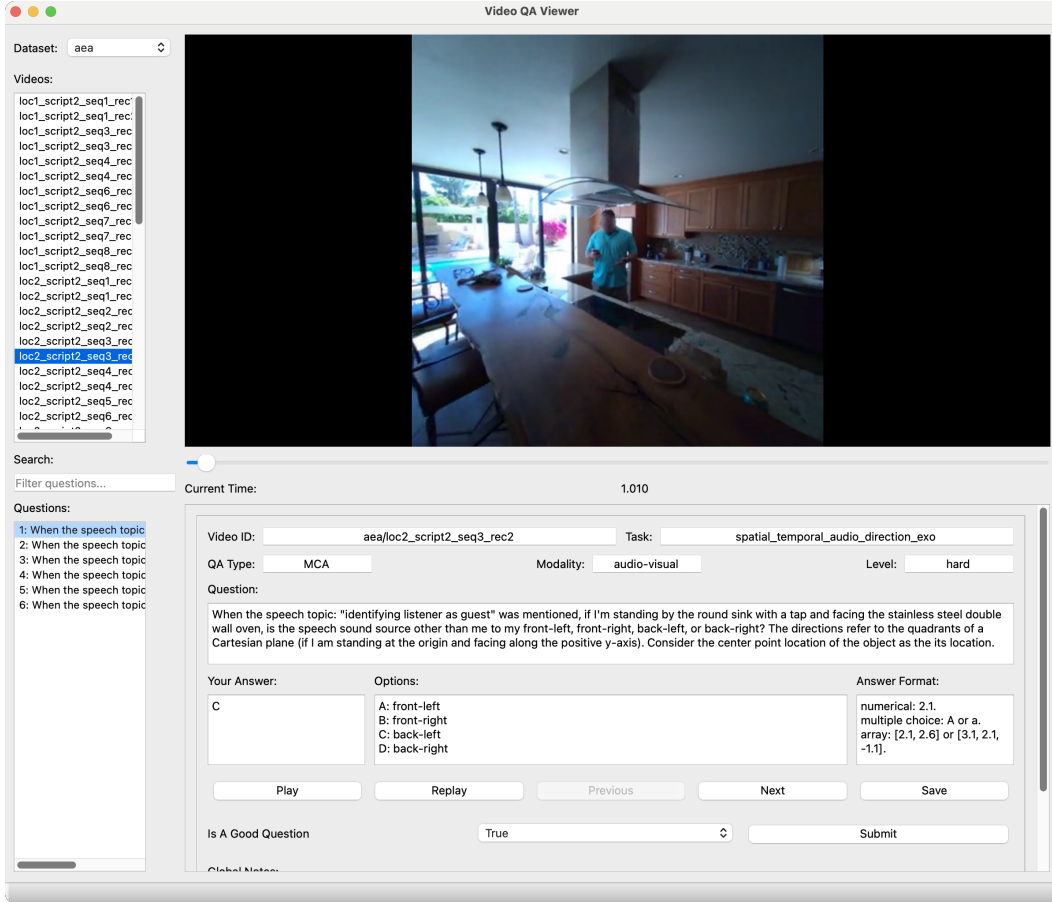


Figure 8: **Review interface for QA pair quality review.** The tool displays each video clip alongside its associated question and predicted answer, allowing reviewers to efficiently assess correctness, clarity, and formatting, and make a decision on whether the QA pair is a good QA that should be retained.

### Allocentric Distance

1. When the {non-speech sound event} sound is happening, what is the distance between the {reference object} and the sounding object in meters? Consider the center point location of the object as the its location. Calculate the Euclidean distance between the two points in the horizontal plane. Answer in numeric format.
2. When the speech topic: {speech topic} is mentioned, what is the distance between the {reference object} and the speech sound source in meters? Consider the center point location of the object as the its location. Calculate the Euclidean distance between the two points in the horizontal plane. Answer in numeric format.

## C.4 Quality Review

We combine automated QA generation with manual review to ensure both scalability and quality. This hybrid pipeline enables efficient creation of large-scale QA pairs while preserving high annotation accuracy. The resulting dataset offers a reliable benchmark for evaluating 3D spatial reasoning in AV-LLMs. In this section, we detail the human quality review process that supports this workflow.

**Review Interface.** To secure the final data quality, we construct a review system with PyQt 5. The system presents each video clip together with its question and answer and offers a simple interface for reviewers to validate or revise the pair efficiently, as illustrated in Figure 8.

**Review Guideline.** Reviewers follow five principles:

- 1) *Correctness*: The stored answer must be fully supported by what is visible and audible in the clip.
- 2) *Clarity*: The question text must be clear and free of ambiguity.
- 3) *Relevance*: A question must refer only to content that is explicitly present in the clip or its metadata. It should not rely on commonsense inference or assumptions beyond what is observable.
- 4) *Consistency*: Answers must respect the predefined format, units, and option labels.
- 5) *Traceability*: Each reviewed QA pair is labeled as accepted or rejected based on whether it qualifies as a “good” question. All edits are logged to support future auditing and reproducibility.

## D SAVVY-Bench Evaluation Details

### D.1 Open-Source AV-LLMs

All experiments are run in inference mode without model training. For open-source AV-LLMs at around 7B scale, we use a single A100 GPU (40GB). For 13B scale AV-LLM, we use a single 80GB VRAM A100 GPU. Evaluation follows the LMMs-Eval module [71]. We use greedy decoding with temperature set to 0, and both top-p and top-k set to 1. Following [71], we sample 32 video frames uniformly across the entire video duration. For audio, we average multiple channels to produce a compressed monaural input, with a sampling rate of 16kHz.

The input for the models is formatted as **[Video Frames]**, **[Audio Content]** and **[Prompt]**

Prompt details:

#### Relative Direction Questions - simple

**[Question]**

Options: A: left B: right C: back.

Answer in single letter or numeric format.

#### Relative Direction Questions - hard

**[Question]**

Options: A: front-left B: front-right C: back-left D: back-right.

Answer in single letter or numeric format.

#### Relative Distance Questions

**[Question]**

Answer in single letter or numeric format.

### D.2 Proprietary Models

For Gemini-2.5-flash and Gemini-2.5-pro, we use Google Cloud Platform’s API. We upload and feed the full video with audio to the model, following API guidelines.

Prompt details:

### Prompt: Proprietary Models on SAVVY-Bench

Given the Video: [Video Frames],  
Question: [Question],  
Options: [Options]

#### [Prompt]

Answer the question.

#### [Format Instructions]

1. Your output **must** be a single, valid JSON object conforming to the schema defined below.
2. **Do NOT** output any thinking steps or reasoning steps.

#### [JSON Schema]

```
{  
  "prediction": "Your final answer (A, B, C, or A, B, C, D, or  
    numeric value). If you can't decide, please output a JSON with  
    the \"prediction\" key's value being null."  
}
```

### D.3 Human Evaluation Guidelines

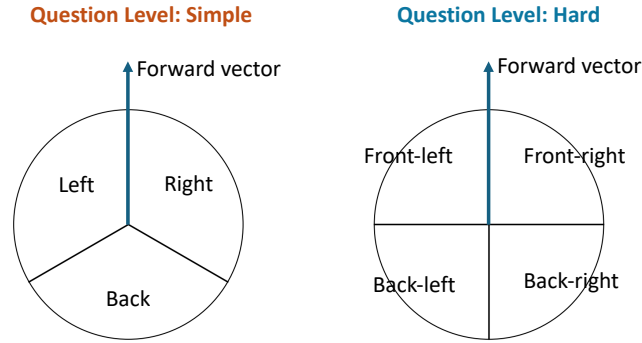


Figure 9: Direction quadrant guide for human evaluation. Egocentric directions are relative to the camera wearer’s facing direction, while allocentric directions use a fixed world frame.

**Evaluation Setup.** We recruited six independent evaluators to participate in the human evaluation. The question set was shuffled and evenly divided among the evaluators. Each evaluator was allowed to pause, replay, or scrub through the video clip as many times as needed before submitting their answer. For direction-based tasks, evaluators followed the quadrant chart shown in Figure 9. For distance-based tasks, the correct response corresponds to the Euclidean distance between the two referenced points projected onto the horizontal plane.

**Evaluation Rules.** Evaluators followed four key rules:

1. *Perspective:* Identify whether the question requires egocentric or allocentric reasoning, and apply the appropriate frame of reference.
2. *Exactness:* Select the most accurate answer supported by visual and audio evidence, avoiding reliance on commonsense inference.
3. *Consistency:* Use the labels and answer formats provided (e.g., A, B, C or numerical values in the specified format).
4. *Independence:* Do not use any external tools such as object trackers or scene maps; rely solely on the provided video clip.



**Ethical Statement.** Participation was voluntary, involved no known physical or psychological risks, and did not collect any personal data beyond the evaluators’ responses.

## E Input Data Details

### E.1 Visual Input Settings

Original fisheye videos from the Aria-Everyday Activities (AEA) dataset [60] were undistorted to a standard rectilinear format for compatibility with common AV-LLM inputs. We also manually aligned the two camera-wearer videos for each conversation, creating a unified timeline to facilitate consistent speech sentence segmentation and speech topic generation. For all open-source AV-LLMs, we evaluated using 32 sampled video frames via uniform sampling.

### E.2 Microphone Configuration

We detail the microphone geometric configuration used in the AEA dataset. The data is collected using Meta’s Aria Glasses, which are equipped with a 7-channel 48kHz microphone array distributed around the frame. Specifically, five microphones are positioned along the front frame, and two are mounted near the rear temple arms. This configuration enables rich spatial audio capture from both forward- and backward-facing directions.

The specific microphone locations (in meters, relative to the center of the glasses) are as follows:

- **Mic 0:** right-front-bottom corner (0.05, −0.04, 0.00)
- **Mic 1:** centered at the bridge of the nose (−0.005, 0.00, 0.00)
- **Mic 2:** left-front-bottom corner (−0.05, −0.04, 0.00)
- **Mic 3:** far-left-up along the front frame (−0.07, 0.00, 0.00)
- **Mic 4:** far-right-up along the front frame (0.07, 0.00, 0.00)
- **Mic 5:** rear left leg (−0.07, 0.00, −0.10)
- **Mic 6:** rear right leg (0.07, 0.00, −0.10)

A visualization of this microphone configuration is available on the Project Aria Hardware Specifications GitHub page.

### E.3 Camera Trajectory

SAVVY uses 6DoF camera trajectories at 1kHz. These trajectories approximate the continuous motion of the egocentric observer and are computed using the foundational visual-inertial odometry (VIO) and simultaneous localization and mapping (SLAM) systems onboard the Project Aria device. In our work, we use the calibrated closed-loop trajectories, represented by 3D position and orientation in quaternion form.

## F SAVVY Details

### F.1 Snapshot Descriptor

As described in the main paper, the Snapshot Descriptor aims to: (1) identify the start and end times of the event; (2) determine whether the question requires an egocentric or allocentric view; (3) identify the *target sounding object*, *reference object*, and *facing object*, along with their text descriptions; and (4) track the egocentric direction and distance of each object at key frames.

To distinguish between views:

- **Egocentric view** refers to the camera wearer’s perspective. In this case, the reference object is the camera, and no facing object is needed. Since the camera trajectory is known, only the target sounding object needs to be identified and tracked.

### Prompt: Open-Source AV-LLMs

#### [Task]

Analyze the given video based on the question: "question". The total video length is duration seconds. Identify the **Sounding Object** (source of sound). Identify the **start\_time** and **end\_time** of the event mentioned in the question. Determine the mode:

- If I'm in the **camera wearer's view** (egocentric), set mode to egocentric.
- If I'm in a **different perspective** rather than the camera's view (allocentric), set mode to allocentric.

#### [Output]

Return a single JSON object with the following structure:

```
{
  "start_time": //start time of the event asked in the question
  "end_time": //end time
  "mode": egocentric/allocentric,
  "sounding_object": {
    "description": "A detailed description of the sounding object (
      source of sound). Include physical characteristics like type,
      color, material, and approximate size/shape.",
    "is_static": true/false // True if the object is generally non-
      moving, false if it typically moves location
  },
  "stand_by_object": {
    "object_name": "Name", //set to camera if requires_allocentric
      is false
    "description": "Description"
  },
  "facing_direction": {
    "object_name": "Name",
    "description": "Description"
  }
}
```

Figure 10: Prompt for Open-Source AV-LLMs on SAVVY-Bench.

- **Allocentric view** requires a perspective other than the camera's. A new coordinate frame is built using the reference object (as the origin) and the facing object (defining the positive y-axis from the reference). In this case, all three objects must be identified and accurately tracked.

Open-source AV-LLMs, typically at the 7B or 13B scale, often struggle to track all objects through prompt guidance. Therefore, we request these AV-LLMs to perform only the first three objectives: identifying the event time span, determining the view mode, and generating accurate object descriptions in correct object categories (target sounding / reference / facing object). For all models, we use greedy decoding with temperature set to 0, and both top-p and top-k set to 1.

Detailed prompts used for both open-source AV-LLMs and proprietary models to generate Snapshot Descriptor are provided in Figure 10 and 11 respectively.

## F.2 Text-Guided Snapshot Segmentation

We uniformly sample 128 frames from each video. For each object, we use its descriptive phrase, extracted from the Snapshot Descriptor, as input to ClipSeg [63] to generate a segmentation mask. Within the segmented region, we sample 10 keypoints and compute the average ClipSeg confidence. A detection is considered valid if the average score exceeds a threshold: 0.5 for dynamic sounding objects and 0.6 for reference and facing objects. We then use the selected keypoints and object descriptions to prompt the SAM model [64], obtaining refined segmentation masks.

### Prompt: Proprietary Models

#### [Task]

Analyze the video at `uploaded_obj` based on the question: `question`.

Identify the **Sounding Object**, the **Reference Object**, and the **Facing Object** (stand by the **Reference Object** and face the **Facing Object**).

Identify the **start\_time** and **end\_time** of the event mentioned in the question.

Determine the mode:

- If I am in the **camera's view** (egocentric), set mode to `egocentric`.
- If I am in a **different perspective** rather than the camera's view (allocentric), set mode to `allocentric`.

Perform **audio-visual tracking** for these objects throughout the *entire duration* of the video.

#### [Tracking Data]

- For each object, provide its estimated position over time.
- Record positions at key moments across the *full video timeline* when the object is clearly visible in the frame.
- Estimate distance in meters from the camera to the object center.
- Estimate direction in degrees (−90 left to 90 right, 0 forward) from the camera.

#### [Output]

Your complete and sole output must be a single JSON object with the following structure:

```
{
  "event": "Brief description of the event from the question",
  "start_time": "minutes:seconds",
  "end_time": "minutes:seconds",
  "mode": "egocentric/allocentric",
  "sounding_object": {
    "description": "A detailed description of the sounding object.
      Include physical characteristics like type, color, material,
      and approximate size/shape.",
    "is_static": true/false, // Set to true if the object is generally
      non-moving (like furniture, walls) and false if it typically
      moves location (like a person, animal, vehicle).
    "key_frames": { /*entire video* key visible frames
      "minutes:seconds": {"distance": "meters", "direction": "degrees"
      }
    }
  },
  "reference_object": { // Stand by Reference Object or camera
    "object_name": "Name",
    "description": "Description",
    "key_frames": { /*entire video* key visible frames
      "minutes:seconds": {"distance": "meters", "direction": "degrees"
      }
    }
  },
  "facing_object": { // Facing the facing_object, empty for camera
    "object_name": "Name",
    "description": "Description",
    "key_frames": { /*entire video* key visible frames
      "minutes:seconds": {"distance": "meters", "direction": "degrees"
      }
    }
  }
}
```

Figure 11: Prompt for Proprietary AV-LLMs (Gemini 2.5 models) on SAVVY-Bench.

To evaluate the robustness of SAVVY with the text-guided segmentation module (*Seg*), we conduct ablation studies on the ClipSeg confidence threshold (*Seg thr*) and the number of sampled frames (*N\_frame*). We report sounding object localization accuracy (*loc\_acc*) and QA accuracy on both egocentric and allocentric tasks from SAVVY-Bench. See the Experiments section of the main paper for detailed metric definitions.

For *Seg thr*, we test values 0.3, 0.5, 0.7, and 0.9, using the average ClipSeg score across keypoints, with all valid detections required to have at least one keypoint above 0.5. Results in Table 7 show stable performance across thresholds 0.3 to 0.7, with less than 3% variation. Lowering the threshold increases object recall, which improves sounding object localization accuracy (*loc\_acc*), as SAVVY’s egotrack-based outlier filtering and aggregation can effectively leverage the additional recalled samples. For QA tasks, a 0.5 threshold yields the highest overall accuracy, while 0.3 improves distance-related QA but reduces directional accuracy.

For *N\_frame*, we evaluate 8, 16, 32, 64, and 128 frames (Table 8). Higher sampling rates lead to more valid detections from *Seg*, boosting sounding object *loc\_acc* by 6.6% from 8 to 128 frames and improving egocentric QA accuracy. However, for allocentric QA, segmentation on static objects may introduce noise. As a result, lower frame counts like 32 or even 8 can perform comparably to 128 frames. These findings suggest a hybrid strategy: use *Seg* for sounding objects and rely more on other egotrack types such as the Snapshot Descriptor for static objects.

| Seg thr | Sound Loc loc_acc | Egocentric QA |          | Allocentric QA |          |
|---------|-------------------|---------------|----------|----------------|----------|
|         |                   | direction     | distance | direction      | distance |
| 0.3     | 79.2              | 83.8          | 64.1     | 43.9           | 41.0     |
| 0.5     | 78.6              | 84.7          | 62.9     | 44.0           | 40.2     |
| 0.7     | 77.1              | 81.4          | 61.2     | 43.4           | 39.9     |
| 0.9     | 69.8              | 77.3          | 59.2     | 43.5           | 40.9     |

Table 7: Ablation results on the average snapshot segmentation confidence threshold (*Seg thr*). We report sounding object localization accuracy (*loc\_acc*) and accuracy on egocentric and allocentric QA tasks. Lower thresholds generally yield higher sounding object recall, improving localization and distance-related QA accuracy with SAVVY, while moderate thresholds provide balanced performance.

| Seg N_frame | Sound Loc loc_acc | Egocentric QA |          | Allocentric QA |          |
|-------------|-------------------|---------------|----------|----------------|----------|
|             |                   | direction     | distance | direction      | distance |
| 128         | 78.6              | 84.7          | 62.9     | 44.0           | 40.2     |
| 64          | 76.7              | 82.7          | 61.6     | 43.0           | 40.2     |
| 32          | 74.8              | 81.9          | 61.1     | 43.7           | 41.4     |
| 16          | 73.8              | 81.9          | 59.8     | 43.2           | 40.5     |
| 8           | 72.0              | 80.1          | 59.4     | 44.7           | 39.9     |

Table 8: Ablation results on the number of sampled frames (*N\_frame*) used in text-guided snapshot segmentation. Increasing the number of frames improves sounding object localization and egocentric QA accuracy. However, allocentric QA performance is less sensitive and can degrade at high frame counts due to noise in static object segmentation.

### F.3 Spatial Audio Cues

We process spatial audio signals at 0.25s per segment, with a sampling rate of 48 kHz. For each segment, we estimate the direction of arrival (DoA) by evaluating candidate angles over the full azimuthal range from  $-180^\circ$  to  $180^\circ$ , sampled at  $1^\circ$  resolution. For each candidate angle, we apply the Generalized Cross-Correlation with Phase Transform (GCC-PHAT) method on each microphone pair to compute time-difference-of-arrival (TDOA) estimates. The angle  $\hat{\phi}$  that maximizes the summed GCC-PHAT responses across all pairs is selected as the most likely direction of the source.

To assess the spatial diffuseness of the sound field for the sound source distance estimation, we compute the Coherent-to-Diffuse Ratio (CDR) from the multi-channel microphone signals. The input

to this process includes the raw microphone waveforms, the sampling frequency  $f_s$ , microphone positions, and the estimated TDOAs for each pair. The analysis is constrained to the 500–2000 Hz frequency band for speech-related audio cues.

We estimate the power spectral densities (PSDs) and cross-spectral densities (CSDs) using Welch’s method, with a segment length of 1536 samples (around 32ms) and 50% overlap. We clip negative values to zero and compute the mean CDR over the selected frequency band. The final CDR is averaged across all microphone pairs and serves as a global indicator of the ratio between coherent (direct-path) and diffuse (reverberant) components in the scene.

#### F.4 Egocentric Track Aggregation

In the second stage of SAVVY, we aggregate three egocentric object tracks—produced by the Snapshot Descriptor, text-guided snapshot segmentation, and spatial cues—into a unified global map. Each per-frame trajectory is transformed into global coordinates, forming a global spatial map for downstream reasoning. The target object forms a time-varying global trajectory  $\{\mathbf{p}_{\text{sound}}(t) \mid t \in \mathcal{T}_q\}$ , while reference and facing objects are treated as static, with global positions  $\mathbf{p}_{\text{ref}}$  and  $\mathbf{p}_{\text{face}}$  computed by averaging their per-frame locations. These together define the **dynamic global map**:

$$\mathcal{M}_q = \{\mathbf{p}_{\text{sound}}(t) \mid t \in \mathcal{T}_q\} \cup \{\mathbf{p}_{\text{ref}}, \mathbf{p}_{\text{face}}\}.$$

We describe the aggregation strategies for static and dynamic objects below.

**Static objects.** Since the Snapshot Descriptor (SD) are better at localizing static objects (reference/-facing) after track aggregation based on our ablation results (see main paper ablations), we prioritize the SD track. If the SD captures the object, we apply DBSCAN clustering (maximum distance of 1 m) on the SD track to determine a stable location. If the SD fails to detect the object, we fall back to the text-guided segmentation-based track (Seg), and apply DBSCAN with the same clustering threshold.

**Dynamic sounding object.** The Seg method is more accurate for tracking sounding objects (see main paper ablations), so we prioritize its trajectory when aggregating dynamic sound source tracks. We log Seg-tracked positions at each timestamp. For timestamps not covered by Seg, we query the SD track and filter outliers based on spatial consistency with the existing Seg trajectory. The resulting track is then extended by spatially fitting a smooth trajectory and removing outliers through the Seg-tracked points.

We then incorporate spatial audio cues to refine this trajectory. Specifically, we define a frustum-based search region for audio tracks around the target direction and distance, spanning a distance range of  $\pm 1$  meter and an angular span of 45 degrees. We sample candidate points at the centers of 10 angular bins and 5 distance bins within this region. If the audio indicates that the object is located behind the camera (i.e., absolute angle  $\theta > 90^\circ$ ), or provides positional information for timestamps not covered by Seg or SD, we refine the track by comparing with audio-based predictions. Inconsistent points are filtered based on spatial agreement with nearby audio-informed estimates, and the trajectory is extended accordingly to produce the final track.

The aggregation process can be summarized as Algorithm 1.

#### Discussion: What roles does the global mapping play in SAVVY?

Camera trajectory serves as the bridge between Stage 1 egocentric tracks and the Stage 2 dynamic global map. It can be obtained using real-time SLAM technologies [62, 60] with devices such as AR glasses or robotic sensors. Given camera pose (location and orientation), egocentric direction  $\theta$  and distance  $r$  can be transformed into global 3D coordinates. This transformation allows tracks from multiple modalities—Snapshot Descriptor (SD), text-guided snapshot segmentation (Seg), and spatial audio cues (Audio)—to be aligned in a shared 3D coordinate system (global mapping). Different modalities may capture object trajectories at different timestamps; by mapping them to a global frame, these partial observations can complement each other. Through outlier filtering and temporal smoothing, we obtain reliable tracks for dynamic objects and stable positions for static ones.

Table 9 compares performance with and without global mapping in terms of sounding object localization accuracy (*loc\_acc*) and egocentric QA accuracy (*direction* and *distance*) on SAVVY-Bench. In the *w/o Global Mapping* setting, we directly take egocentric tracks from SD, Seg, and Audio based on the Snapshot Descriptor’s grounded time span, then vote on direction and take the median angle and

---

**Algorithm 1** Track Aggregation Algorithm for Global Map Construction

---

```
1: Input:  $\mathcal{S}, \mathcal{D}, \mathcal{A}$  (dense segmentation, SD, audio tracks);  $o$  (object type);  $\mathbf{L}(t)$  (camera trajectory);  $\mathcal{T}_q$  (query time range)
2: Define:  $\text{MapToGlobal}(\boldsymbol{\tau}, \mathbf{L}(t)) := \mathbf{L}(t) + \begin{bmatrix} r \cdot \cos(\theta) \\ r \cdot \sin(\theta) \end{bmatrix}$ , where  $\boldsymbol{\tau} = (t, \theta, r)$ 
3: Initialize map  $\mathcal{M}_q \leftarrow \emptyset$ 
4: if  $o$  is static then
5:   for each  $\boldsymbol{\tau} \in \mathcal{D}, \mathcal{S}$  do
6:      $\mathbf{p}(t) \leftarrow \text{MapToGlobal}(\boldsymbol{\tau}, \mathbf{L}(t))$ 
7:     break
8:   end for
9:    $\bar{\mathbf{p}} \leftarrow$  centroid of clustered  $\mathbf{p}(t)$ 
10:   $\mathcal{M}_q \leftarrow \mathcal{M}_q \cup \{\bar{\mathbf{p}}\}$ 
11: else
12:   Initialize trajectory  $\mathbf{p}(t) \leftarrow \emptyset$ 
13:   for each  $t \in \mathcal{T}_q$  do
14:     for each  $\boldsymbol{\tau}$  in  $\{\mathcal{S}, \mathcal{D}, \mathcal{A}\}$  if  $t \in \boldsymbol{\tau}$  do
15:       Filter outliers near  $\mathbf{p}(t')$ 
16:        $\mathbf{p}(t) \leftarrow \text{MapToGlobal}(\boldsymbol{\tau}, \mathbf{L}(t))$ 
17:     end for
18:   end for
19:   Interpolate and smooth  $\mathbf{p}(t)$  over  $\mathcal{T}_q$ 
20:    $\mathcal{M}_q \leftarrow \mathcal{M}_q \cup \{\mathbf{p}(t)\}$ 
21: end if
22: return  $\mathcal{M}_q$ 
```

---

distance at the queried time. Global mapping improves single-modality performance, especially for dense tracks like Seg and Audio, which see localization accuracy (*loc\_acc*) gains of about 10%. SD, being sparse, is less sensitive to global mapping and may perform better without it. For combined modalities, global mapping not only supports self-correction within each modality but also enables cross-modality completion, yielding even greater improvements—up to 11.5% on egocentric distance accuracy and *loc\_acc*. Full SAVVY with all three tracks shows the strongest gains: +11.9% in *loc\_acc*, +14.3% in egocentric distance accuracy, and +4.1% in direction estimation.

| Track Type |       |     | w/ Global Mapping (SAVVY) |           |          | w/o Global Mapping |           |          |
|------------|-------|-----|---------------------------|-----------|----------|--------------------|-----------|----------|
| SD         | Audio | Seg | loc_acc                   | direction | distance | loc_acc            | direction | distance |
| ✓          |       |     | 55.7                      | 68.3      | 47.9     | 56.3               | 71.1      | 52.6     |
|            | ✓     |     | 59.0                      | 73.9      | 48.1     | 49.7               | 75.6      | 40.1     |
|            |       | ✓   | 72.5                      | 81.2      | 52.0     | 62.3               | 75.8      | 43.7     |
| ✓          | ✓     |     | 66.8                      | 74.5      | 54.6     | 55.3               | 73.0      | 43.3     |
| ✓          | ✓     | ✓   | 78.6                      | 84.7      | 62.9     | 66.7               | 80.6      | 48.6     |

Table 9: Ablation study on the impact of global mapping. We evaluate combinations of egocentric track modalities—Snapshot Descriptor (*SD*), Spatial Audio (*Audio*), and Segmentation (*Seg*)—with and without global coordinate transformation. Metrics include sounding object localization accuracy (*loc\_acc*) and egocentric QA accuracy on SAVVY-Bench (*direction* and *distance*). Global mapping consistently enhances performance, particularly when aggregating dense tracks (*Seg* and *Audio*) and integrating multiple modalities.

## G Efficiency Analysis

We report average latency and peak GPU memory for each stage over 200 test samples on a single NVIDIA A100. **1) SLAM:** real-time on Aria glasses. **2) SD:** one AV-LLM forward pass per QA (latency comparable to standard AV-LLM inference). **3) Audio & Global Map:** fewer than 0.1 s per sample on CPU. **4) Seg:**  $\approx 0.52$  s/frame at  $512 \times 384$  per object; peak GPU memory 9.4 GB. Depth estimation costs  $\approx 0.44$  s/frame (up to 6 GB). While *Seg* is the primary bottleneck of the efficiency, Table 8 shows that using 32 frames per video for *Seg* maintains strong accuracy. Besides, removing *Seg* (Table 6 in main paper) substantially reduces runtime while still outperforming LLM-

only baselines, particularly on allocentric questions. Overall, modular design of SAVVY enables accuracy–efficiency trade-offs to meet real-time deployment constraints.

## H Blind Testing

We conduct blind testing to evaluate the contribution of the visual modality in audio-visual spatial reasoning on SAVVY-Bench, using AV-LLM baseline models. Specifically, we compare performance between two settings: *Audio Only* (removing visual frames, using only audio and the text query as input) and *Audio + Visual* (using both modalities). We evaluate on egocentric QA tasks to assess how models infer the direction and distance of sound sources relative to the camera.

We test the top five open-source 7B models and the strongest proprietary model, Gemini-2.5-pro. As shown in Table 10, Gemini demonstrates strong grounding capabilities (67.4% t-mIoU, as reported in the main paper), and its performance shows a clear dependence on visual input. Under *Audio Only*, Gemini’s direction accuracy drops sharply by 32.4%, while distance accuracy decreases by only 2.8%. This aligns with observations from our reasoning process visualizations: Gemini relies heavily on visual input for spatial direction reasoning, whereas distance estimation is less affected—likely due to the role of commonsense priors from audio and language.

Other AV-LLMs exhibit similar trends: direction accuracy degrades more under *Audio Only*, while distance accuracy remains relatively stable or even improves for some AV-LLMs such as MiniCPM-o. However, the performance gap is smaller than with Gemini, likely because these models fail to reliably ground events in time—achieving less than 5% t-mIoU—regardless of the input modality. As a result, even with visual input, their spatial reasoning remains limited.

| Method              | Audio Only |          | Audio + Visual |          |
|---------------------|------------|----------|----------------|----------|
|                     | Direction  | Distance | Direction      | Distance |
| VideoLLaMA2-7B [21] | 39.1       | 40.7     | 46.4           | 42.7     |
| MiniCPM-o 2.6 [25]  | 41.9       | 50.7     | 45.8           | 42.3     |
| EgoGPT [23]         | 39.3       | 37.0     | 40.2           | 57.6     |
| Gemini-2.5-pro      | 42.8       | 56.8     | 75.2           | 59.6     |

Table 10: Blind testing on SAVVY-Bench: comparison between *Audio Only* and *Audio + Visual* input settings. Reported metrics are egocentric QA accuracy for direction and absolute distance. Gemini-2.5-pro shows the largest gap, indicating strong reliance on visual input for accurate direction estimation.

## I Limitations

One limitation of SAVVY is that it currently relies on a strong foundational AV-LLM—specifically Gemini—and inherits its capabilities in temporal grounding and object referral. The pipeline may underperform if the base model lacks these abilities in the initial stage. Additionally, the spatial audio tracking module uses rule-based signal processing: while effective for direction estimation, distance estimation remains challenging, particularly given the wide variance of near- and far-field cases in the current dataset. Future work could improve audio-visual track aggregation by enhancing this module through large-scale training on realistic spatial audio data.

## J Broader Impacts

This work contributes to the development of AV-LLMs capable of fine-grained spatial reasoning in dynamic 3D environments. By introducing a benchmark and training-free pipeline that enables structured spatial understanding across audio and visual modalities, our work opens new avenues for intelligent multi-modal systems in domains such as assistive robotics, AR/VR, human-computer interaction, and audio-visual navigation [72]. These capabilities have the potential to significantly enhance accessibility tools (e.g., guiding visually impaired users through complex spaces), improve AR/VR user experiences, and support more context-aware AI agents in embodied environments.

However, alongside these benefits, the increasing power of AV-LLMs introduces potential risks. Models capable of interpreting spatial relationships from audio-visual input could be misused in surveillance applications, unauthorized tracking, or context inference without user consent. Moreover, as our method builds on these foundation models, it inherits their limitations and biases, which can propagate through the pipeline and affect real-world deployments. There is also the risk that such models may make confident but incorrect spatial inferences in safety-critical settings. To mitigate these concerns, we recommend that future systems incorporating AV-LLMs for spatial reasoning include safeguards such as: (1) explicit transparency about model uncertainty and failure modes; (2) data collection and evaluation guidelines that prioritize privacy and ethical use of human-centered audio-visual data; and (3) usage restrictions for sensitive applications, especially those involving biometric data or real-time environmental monitoring. Furthermore, research into interpretability and robustness of spatial reasoning components will be critical for safe deployment.

## K Additional Qualitative results: Reasoning Error Analysis

In this section, we show additional reasoning examples of Gemini-2.5-pro and conclude four major types of errors in the visualization:

- 1) *Referral Error*: This error occurs when the model fails to correctly identify, locate, or interpret the properties of specific objects, persons, or abstract reference points mentioned in the question. It is particularly common when the referenced object descriptions are complex, rely on relative positioning (e.g., “the armchair further from the wall painting”), or refer to abstract sound events (e.g., “a thud sound”) that are not tied to a clearly visible object and must be inferred from broader video context. The model may select an incorrect referent or misinterpret its attributes, leading to a flawed premise for subsequent spatial reasoning. An example is shown in Figure 12, where the model incorrectly identifies the queried armchair (the facing object) as the one at the arched opening.
- 2) *Temporal Localization Error*: This error occurs when the model fails to accurately identify the correct time span of the queried sound event in the question. As a result, the model analyzes the spatial context at an incorrect point in time, leading to flawed reasoning about object locations or spatial relationships. Figure 13 shows an example where the model confuses the speech event “suggesting trying the coffee” with another semantically similar topic, “complimenting the coffee taste,” leading to an error in egocentric direction prediction.
- 3) *Spatial Relationship Error*: This error occurs when the model misinterprets or misapplies fundamental spatial relationships (e.g., left/right, front/back, in front of/behind, next to, between) between correctly identified entities, even within a correct frame of reference. In Figure 14, the model successfully identifies the correct event time span, detects all relevant objects as well as their locations. However, it fails to interpret the relative direction correctly, placing the object on the right side of the robot’s view instead of the left, resulting in an incorrect prediction of “front-right” rather than the correct “front-left.”
- 4) *Spatial Measurement Error*: This error arises in tasks that require quantitative responses—such as estimating distances or making precise angular judgments (e.g., in Snapshot Descriptor-based tasks). Even when the model correctly identifies the relevant objects and understands their qualitative spatial relationships, it may still make significant errors in geometric reasoning (e.g., applying Pythagorean theorem incorrectly, flawed calculation logic), scale estimation, or numerical calculations. Figure 15 presents an example where the model correctly identifies both the sound source and the queried reference object, and even retrieves a relevant navigation path between them. However, it fails to calculate the distance accurately. This case also reveals a typical reasoning pattern in AV-LLMs for distance estimation: the model anchors the sound source and reference object to static landmarks in the scene, recalls the relevant navigation routes observed in the video, and then estimates the distance according to the routes.



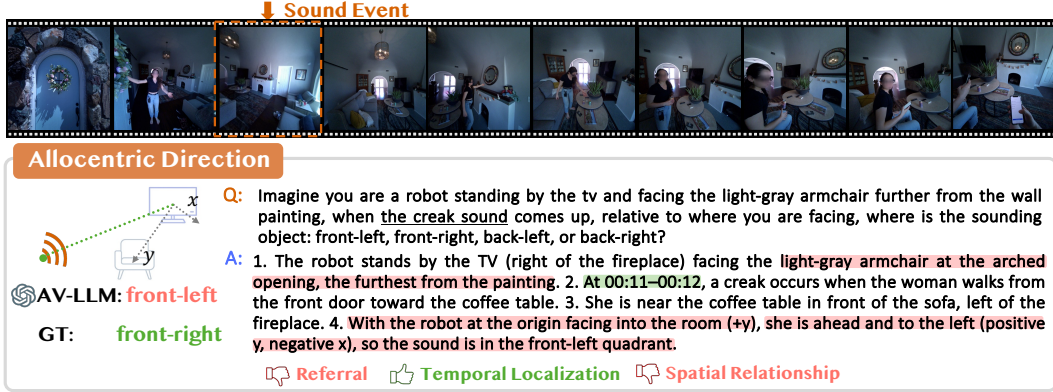


Figure 12: **Example of Referral Error.** The model incorrectly selects the armchair (facing object) at the arched opening, failing to identify the correct object description queried in the question. This misinterpretation further causes error for the spatial relationship.



Figure 13: **Example of Temporal Localization Error.** The model incorrectly grounds the speech event "suggesting trying the coffee," confusing it with a similar topic. This mismatch causes incorrect egocentric direction prediction.

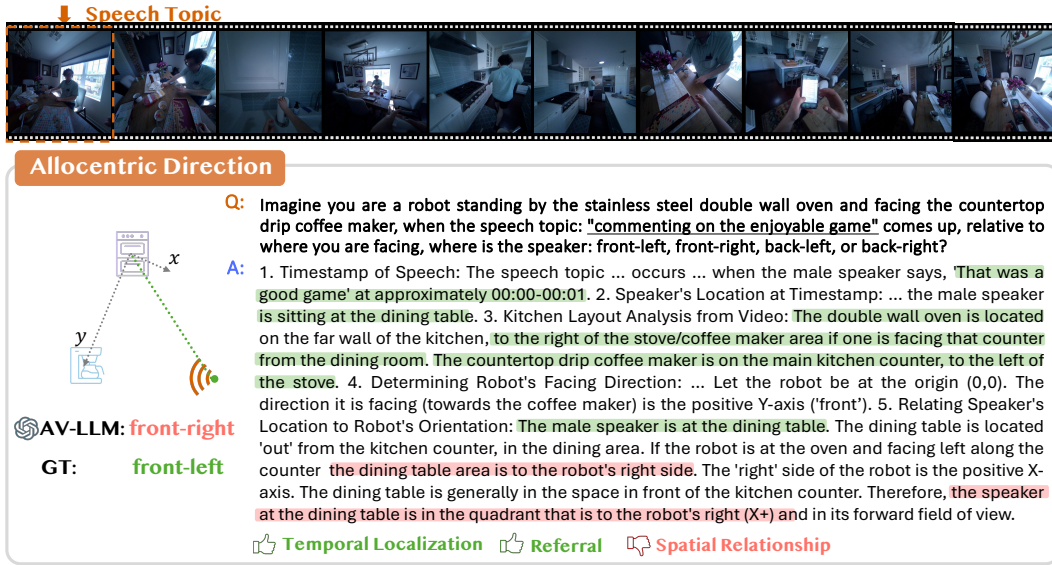


Figure 14: **Example of Spatial Relationship Error.** The model correctly identifies the queried time span, relevant objects, and their locations, but fails to determine the correct relative direction of the sounding object with respect to the robot's view. It incorrectly predicts the object as being on the right side instead of the left, leading to a misclassification of "front-right" instead of the correct "front-left."

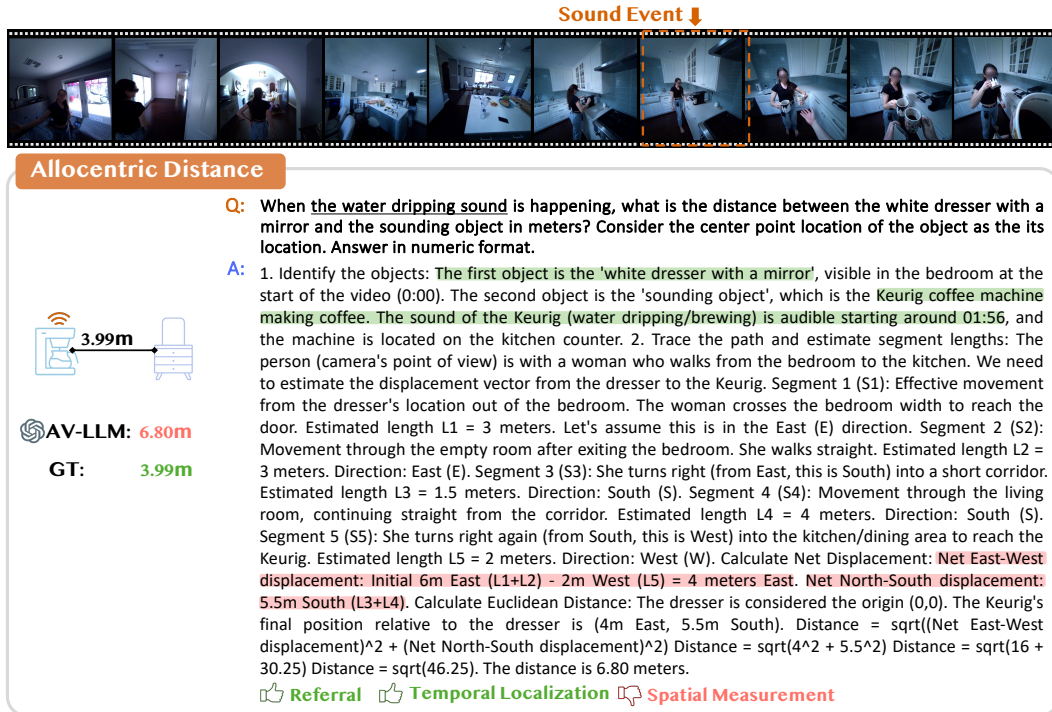


Figure 15: **Example of Spatial Measurement Error.** The model correctly identifies the sound source and reference objects, but fails to compute the distance accurately along with the navigation route from the reference object to the sound source.