Supplementary Material: Appendix

A. Overview

This appendix provides additional details and analyses to support the findings presented in the main manuscript. We organize this material as follows: B describes the specification of the subsets and annotations used for evaluating egocentric spatio-temporal action understanding on EPIC-KITCHENS and Ego-Exo4D. Section C presents detailed information about the implementation and experimental configurations. Later, Section D showcases additional qualitative results and visual examples. Section E presents ablation studies on description types for audio features, audio extraction window size, audio robustness to background noise and audio integration strategies. Sections F and G discuss the limitations of our approach and broader societal implications. Finally, we conclude with Section H, which provides a complete list of all assets used in this work.

B. Data Preprocessing

B.1. EPIC-KITCHENS subset

We evaluate on scenes from the EPIC-KITCHENS (EK-100) dataset [1, 2] that meet the following criteria: (1) availability of segmentation masks from VISOR [3], (2) released estimated camera poses from EPIC Fields [15], and (3) existing corresponding audio annotations from EPIC-Sounds [6]. These criteria ensure comprehensive multimodal ground truth for our analysis. We select 7 representative scenes that provide diverse kitchen environments, action types, camera motions, and acoustic conditions.

B.1.1. Training and Test Splits

Each video is split into training and test sets. The training set is used for reconstruction and feature distillation, while the test set evaluates generalization to unseen views. To construct a reliable test set based on manually verified segmentation masks (VISOR GroundTruth-SparseAnnotations), we apply a two-stage sampling strategy:

1. **Action coverage**: we sample frames with manually annotated masks from different actions, ensuring at least one frame per action (*within action*).
2. **Background coverage**: When possible, we sample twice as many frames from non-action segments of the video (*outside of action*). In action-dense scenes (>80% action coverage), we proportionally reduce background sampling while maintaining temporal diversity where possible, which ensures balanced training data and fair evaluation across all test frames.

The remaining frames are allocated to the training set. Some frames are later excluded from both splits due to missing camera pose annotations in EPIC Fields [15]. Similarly, actions without corresponding manual segmentation masks in VISOR [3] are removed from the test set.

To support reproducibility and future benchmarking, we release our training and test splits alongside the source code. Detailed dataset statistics are provided in Table 1.

B.1.2. Annotation Refinements

To associate actions with sound descriptions, we use annotations from EPIC-Sounds [6]. An action is paired with a sound description if its timestamp aligns. In cases where multiple sound descriptions align with a single action, we compute text feature embeddings for each and use their average representation during retrieval.

To ensure the reliability of the ground truth in our evaluations, we perform minimal refinements to address objective annotation inconsistencies visible in the video frames. The adjustments were made only when necessary to enable accurate interpretation of the depicted actions.

Refined ambiguous narrations. As shown in Figure 1 (left), we refined ambiguous narrations in several instances. For example, narration ID `P18_01_41`, originally described as "put back in toaster", was refined to "put back bread slices in toaster". Similarly, for `P18_01_36`, the narration "place it on the plate" was modified to "place knife on the plate".

Consolidated action segments. We also consolidated redundant action segments. As shown in Figure 1 (center), segments `P18_02_13` ("wash coffee pot") and `P18_02_14` ("still washing coffee pot") were merged into a single, continuous action to better reflect the uninterrupted activity.

Extended temporal range. In a few cases, we also refined the temporal range of the actions. For instance, in segment `P18_02_23`, shown in Figure 1 (right), the action "spread peanut butter onto toast" was initially annotated to end too early. We extended the segment to include all relevant frames, such as the bottom frame, in which the action is still visibly ongoing.

In total, we refined the annotations of 18 out of 112 actions (approximately 16%), keeping the changes minimal and limited to essential corrections. The 112 action segments (2-33 per scene) average 3.32 s (226.35 frames) each, combining 20 verbs and 33 nouns into 94 unique narrations. The action distribution shows that most narrations (78) occur once, 15 twice, and "open drawer" 4 times as the most frequent. All updated annotations will be released to ensure fair and consistent comparisons in future work.

B.2. Ego-Exo4D subset

To demonstrate our framework's generalizability across different activity domains, we construct an additional evaluation subset from Ego-Exo4D [5], a large-scale multimodal dataset capturing diverse human activities from both egocentric and exocentric viewpoints. We selected two representative egocentric scenes with complementary characteristics:

Table 1. **EK-100 Subset Statistics.** Statistics per scene for the EK-100 subset used in our experiments. **KID**: EK-100 video ID. **Length (s)**: full video duration in seconds. **Frames**: total number of original frames. **w/ Poses**: frames with registered camera poses from EPIC Fields. **w/ GT Masks**: frames annotated with GT VISOR segmentation masks. **Train/Test**: number of frames included in our training and test splits. **Actions**: action segments with at least one frame with ground truth VISOR annotations. **A w/ Sound**: number of these actions that also have corresponding audio annotations from EPIC-Sounds [6].

ID	KID	Length (s)	Frames	w/ Poses	w/ GT Masks	Train	Test	Actions	A w/ Sound
01	P09_07	55	3325	3013	32	2931	48	10	7
02	P32_07	29	1745	1728	6	1710	17	3	3
03	P18_01	166	9992	9832	69	9627	162	33	26
04	P18_02	190	11435	11342	44	11215	90	24	13
05	P04_26	37	2219	2084	4	2073	11	2	2
06	P03_11	53	3164	2654	29	2578	52	10	9
07	P06_11	77	4624	4106	36	4008	65	17	7

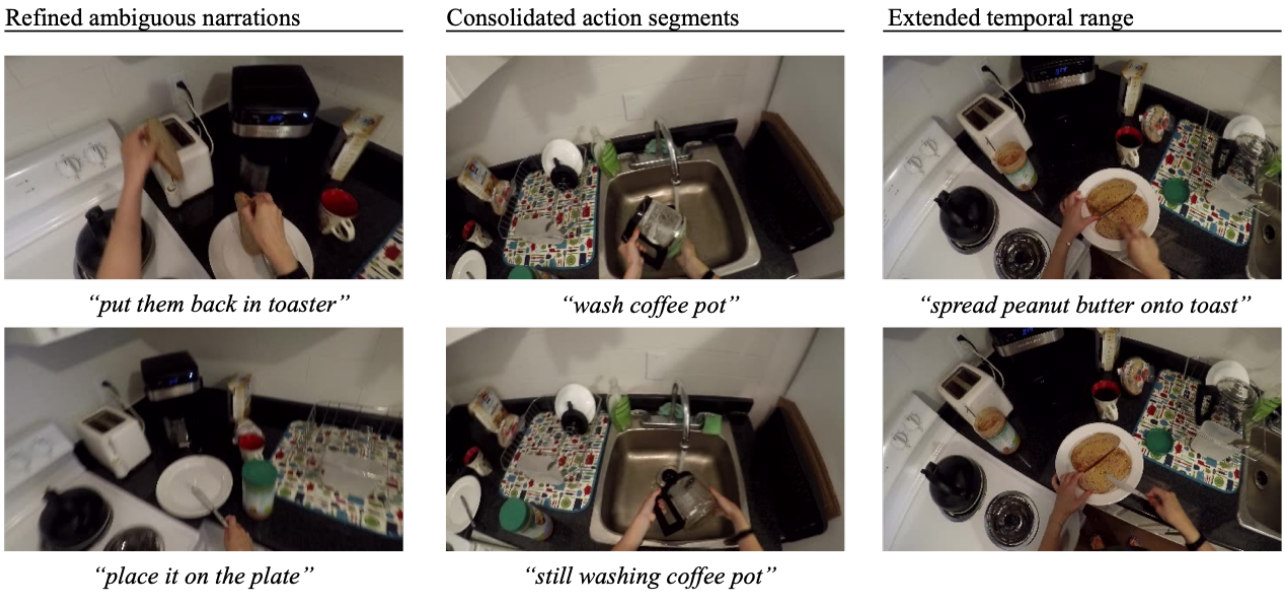

Figure 1. **Annotation Refinements. Left column:** Examples of clarified narrations. Top: `P18_01_41:frame_0000008680` was refined from "put back in toaster" to "put back *bread slices* in toaster". Bottom: `P18_01_36:frame_0000007662` was refined from "place *it* on the plate" to "place *knife* on the plate". **Middle column:** Consolidated action segments. Top frame (`P18_02_13:frame_0000005760`) labeled "wash coffee pot" and bottom frame (`P18_02_14:frame_0000007395`) labeled "still washing coffee pot" were merged into a single continuous action with a single annotation. **Right column:** Extended temporal range. Action segment `P18_02_23` originally included the top frame (`frame_0000010407`) for "spread peanut butter onto toast" but excluded the bottom frame (`frame_0000010740`), where the action was still ongoing. This frame was added to accurately reflect the full action duration.

(1) `unc_basketball_03-31-23_01_7` (48s, 35 original narrations, 6 unique actions) capturing dynamic sports activities in a basketball court, and (2) `fair_bike_06_10` (35s, 9 original narrations, 7 unique actions) documenting technical repair activities in a bike workshop. These scenes complement our EK-100 evaluation by providing different environmental conditions, activity types, and motion dynamics.

B.2.1. Manual Annotation Process

Unlike EK-100, which leveraged existing multi-source annotations, the Ego-Exo4D scenes required comprehensive manual annotation across all modalities to provide suitable ground truth for our evaluation. Our annotation process included:

1. **Action boundaries and narrations.** We refined and consolidated the narrations provided by Ego-Exo4D to match the concise style of EK-100, mainly containing [verb +

object], as the original descriptions were overly detailed for our task requirements. For example, the original Ego-Exo4D provides highly granular descriptions such as "C shoots a jump shot at the hoop with both hands" and "The basketball bounces off the rim," which we consolidated into simpler actions like "shoots basketball at the hoop" and "catches basketball." Additionally, we manually annotated action boundaries as only timestamps were provided in the original annotations.

2. **Segmentation masks.** We combined existing segmentation masks provided by Ego-Exo4D with additional masks generated using SAM2 [13] to ensure comprehensive coverage, as not all frames and relevant objects had annotations in the original dataset. Notably, hand segmentation was not included in the original annotations, requiring us to generate these critical masks for egocentric activity understanding.
3. **Audio annotations.** We manually annotated corresponding sound descriptions to enable multimodal evaluation consistent with our EK-100 analysis.

All annotations were carefully reviewed to maintain consistency with our evaluation framework and ensure high-quality ground truth for reliable assessment. In total, we processed and refined annotations for 44 narrations across both scenes (35 from the basketball scene, 9 from the bike scene), consolidating them into 13 unique actions.

To match the conditions of our EK-100 evaluation, we estimated camera poses for the Ego-Exo4D scenes using the EPIC Fields approach [15], enabling consistent 3D reconstruction and neural rendering across both datasets.

B.2.2. Training and Test Splits

Similar to the EK-100 subset, each video is split into training and test sets following our two-stage sampling strategy. We apply the same approach of ensuring action coverage by sampling frames from different actions, and background coverage by including frames from non-action segments where possible. Detailed dataset statistics are provided in Table 2.

To support reproducibility and future benchmarking, we will release our annotations, training and test splits alongside the source code.

C. Implementation Details

C.1. Mask and Crop Generation

We compute dense feature maps for every third frame and extract object-centric crops fully automatically via SAM [9] and SAM2 [13], requiring no manual annotations or human intervention, building on the implementation of Zrporz [18]. SAM is applied using a 32×32 grid of point prompts. The resulting masks are filtered based on the following criteria: Intersection-over-Union (IoU) with previously selected masks less than 0.7, stability score of at least 0.85, and overlap ratio below 0.7. Each retained mask is assigned a unique masklet ID. These masks are then propagated across subsequent frames using SAM2, ensuring the consistency of masklet IDs. Every 20 propagated frames, we compute the uncovered area ratio. If this ratio increases by more than 1% relative to the baseline established at the previous keyframe, we treat this as a significant scene change. At these change points, SAM is reapplied to the current frame to detect new objects and update the set of active masklets accordingly.

To generate the final crops, we compute the largest bounding box that encloses all masks associated with each masklet instance over a fixed-length video clip. For practical reasons, we define the clip duration as the temporal input length of the downstream VideoLM. For each crop, we center this bounding box on the current position of the object and pad it to form a square. This ensures consistent spatial framing and contextual integrity while aligning with the VideoLM input requirements. Pixels not associated with any masklet receive only RGB supervision during NeRF training, while the feature distillation loss is applied exclusively to masked regions.

C.2. Feature Extractors

We extract features using the following pretrained models:

- CLIP [7]: Visual features extracted using OpenCLIP ViT-B/32.
- SigLIP [16]: Visual features extracted using SigLIP-B/16 with 256 image patches.
- ImageBind [4]: Visual and audio features extracted using ImageBind-H.
- LaViLa [17]: Zero-shot visual features extracted using LaViLa TSF-B.
- EgoVideo [11]: Video features extracted using the EgoVideo model with 4-frame input segments.

C.3. NeRF Training

We extend the three-stream architecture of NeuralDiff [14] by adding a two-layer MLP to each feature stream. The output dimension of each MLP matches the corresponding feature embedding size: 256 for LaViLa, 512 for CLIP and EgoVideo, 768 for SigLIP, and 1024 for ImageBind. We train separate models for each feature type on a per-scene basis. Each experiment takes approximately 10 hours on a single NVIDIA A6000 GPU.

Training is conducted using hierarchical sampling with separate coarse and fine networks. We use a batch size of 1024 and optimize with Adam for 10 epochs. The initial learning rate is set to 5×10^{-4} and decays following a cosine annealing schedule. All feature embeddings are normalized after the rendering stage.

The average rendering time per frame for inference varies by feature type, ranging from approximately 3 minutes for

Table 2. **Ego-Exo4D Subset Statistics.** Statistics per scene for the Ego-Exo4D subset used in our experiments. **Scene ID**: Ego-Exo4D video identifier. **Length (s)**: video duration in seconds. **Frames**: total number of original frames. **Train**: number of frames included in our training split. **Test**: number of frames included in our test split. **Actions**: unique action types with ground truth annotations.

ID	Scene ID	Length (s)	Frames	Train	Test	Actions
01	unc_basketball_03-31-23_01_7	48	1511	1463	44	6
02	fair_bike_06_10	35	1059	1027	32	7

LaViLa to 10 minutes for ImageBind. Rendering is performed with a chunk size of 1024 to limit memory usage, measured on a single NVIDIA A6000 GPU. Since AViON4D is agnostic to the underlying 3D representation, it can be integrated with more efficient rendering techniques, such as Gaussian Splatting [8], to significantly reduce rendering times.

C.4. Audio-Visual Relevance Score Fusion

To account for differences in score distributions and dynamic ranges across models, we use model-specific α values to fuse visual and audio similarity scores. A uniform α would be suboptimal, as the relative contributions of visual and audio features vary significantly between models. These variations influence both the scale and the effectiveness of cross-modal fusion, directly impacting localization performance. We determine the optimal α values via grid search, resulting in the following settings: LaViLa (0.5), CLIP (0.5), SigLIP (0.9), EgoVideo (0.8), and ImageBind (0.8). This adaptive weighting ensures a balanced contribution from both modalities, leading to consistently improved localization performance across all models.

D. Qualitative Results

The sequence in Figure 2 shows a person washing their hands at a kitchen sink (1), followed by washing a plate (2), and finally washing a coffee pot (3, 4) before placing it to dry (5, 6). The corresponding heatmaps below track temporal actions, with brightness indicating similarity scores. The model successfully identifies and localizes coffee pot washing while distinguishing it from other washing activities. Spatial predictions are accurate, with activations consistently focused on the sink area and intensifying during coffee pot washing.

Figure 3 demonstrates the advantage of video-based models (EgoVideo, LaViLa) over image-based models (CLIP, SigLIP, ImageBind) for temporally opposite actions. Video models correctly distinguish "take bread slices out of toaster" from "put back bread slices in toaster", while image-based models confuse these visually identical but temporally opposite actions, despite some (SigLIP, ImageBind) retrieving semantically related content.

Additional qualitative results (see Figure 4) demonstrate that relying solely on visual features can cause inaccurate spatio-temporal action localization, while incorporating audio enhances accuracy by providing action-specific auditory cues for both image and video models. For example, the query "wash spoon" is initially localized by CLIP to a static scene, but when the sound of running water is included, the action is correctly identified. For LaViLa when dealing with the query "take peas", audio helps suppress noise artifacts, enabling precise spatio-temporal localization.

E. Additional Ablations

In this section, we evaluate different components of AViON4D, complementing the main ablation in the manuscript. Unless otherwise specified, for these experiments we use all seven scenes of the EK-100 subset with CLIP, LaViLa, and EgoVideo models.

E.1. Audio Window

We evaluate audio window lengths of 1, 2.5, and 3.5 seconds for extracting audio segments per frame (Table 3). The 1-second window falls below ImageBind's requirement of approximately 2-second audio clips sampled at 16 kHz (32,000 samples), triggering zero-padding and degrading audio feature quality. On the other hand, the 3.5-second window introduces excessive context, potentially overlapping information between unrelated frames, which also negatively affects performance. The 2.5-second window achieves the best trade-off as it is long enough to avoid padding while remaining temporally focused, leading to the highest accuracy across all three models.

E.2. Text Prompts

AViON4D computes a relevancy score by comparing audio embeddings with textual descriptions. These queries can be either the same action descriptions used for visual similarity or dedicated descriptions of the associated sounds. In our setup, we use paired sound descriptions from the EPIC-Sounds Dataset. In this ablation, we compare this approach to using sound descriptions automatically generated by a Large Language Model (LLM) such as GPT-4 [10]. Specifically, we prompt GPT-4 with: "Which is the sound produced by *[action]*". Our results show that sound-focused descriptions lead to the greatest performance improvements, and

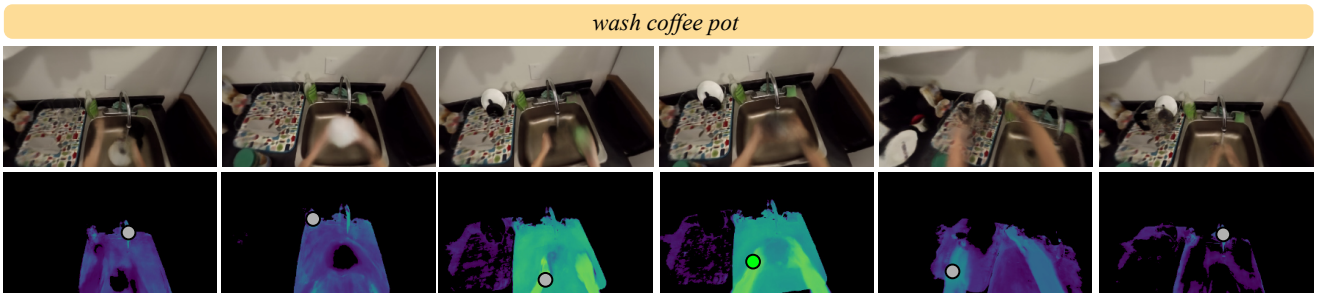

Figure 2. **Spatio-temporal localization for "wash coffee pot".** Top: Rendered video frames showing hand washing (1), plate washing (2), and coffee pot washing (3-4), then placing to dry (5-6). Bottom: Similarity heatmaps show the model correctly distinguishes coffee pot washing from other washing activities, with spatial predictions consistently focused on the sink area.

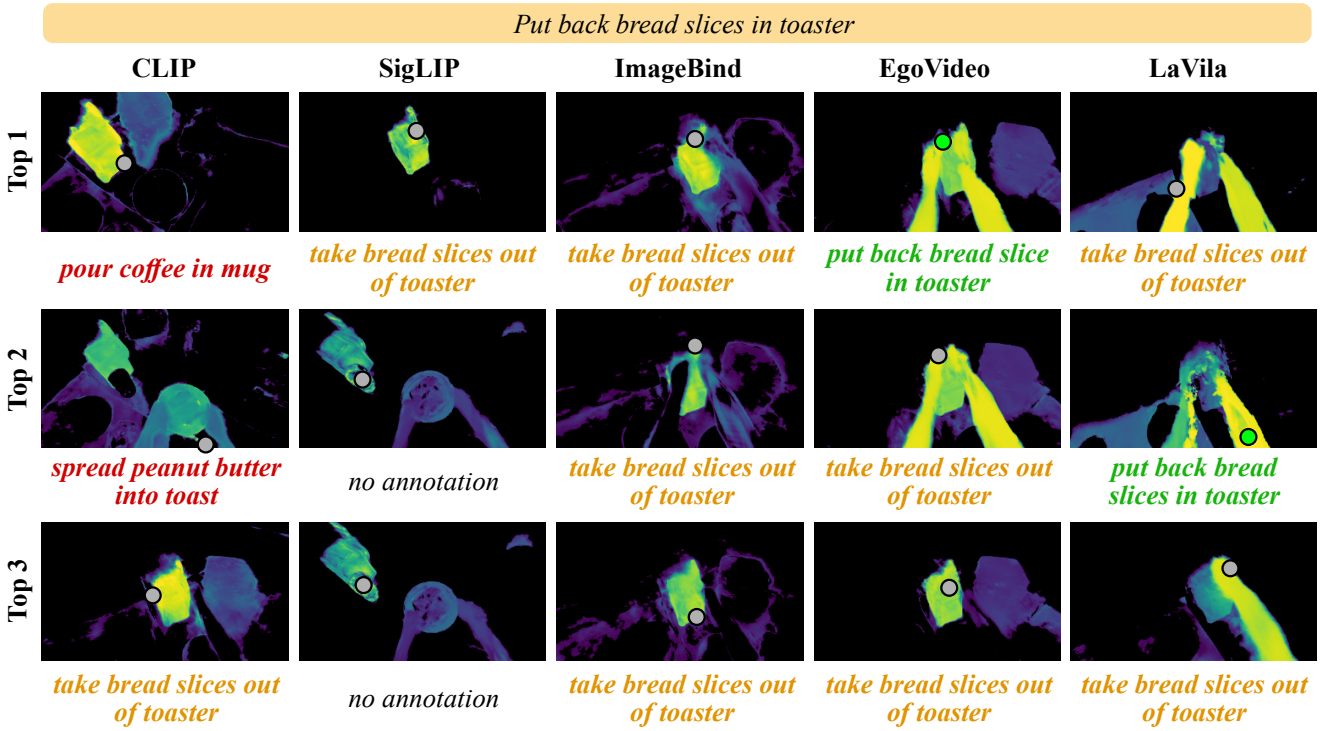

Figure 3. **Comparison of top-3 similarity maps for "put back bread slices in toaster".** Results across CLIP, SigLIP, ImageBind, EgoVideo, and LaViLa visual encoders. Ground truth annotations are shown below each prediction. Video models (EgoVideo, LaViLa) correctly retrieve the action (text in green), whereas image-based models either confuse it with the opposite action ("take bread slices out of toaster" in orange) or mistake it for a different action (text in red).

that LLM-generated descriptions closely approximate the effectiveness of human-annotated ones.

E.3. Audio Robustness to Background Noise

We evaluate the robustness of our audio-visual fusion approach under noisy conditions by adding synthetic white noise to the audio tracks (Table 5). We added white noise with a Signal-to-Noise Ratio (SNR) of 15-20 dB to evaluate performance degradation under realistic background noise conditions. Prior analysis of the original audio revealed challenging baseline conditions: under-leveled recordings ($-$ 31.9 dB RMS), substantial ambient noise and poor frequency balance.

Despite these already degraded audio conditions, adding synthetic noise caused minimal performance degradation, 0% to 1.4% across all encoders. Importantly, audio-visual fusion continues to outperform visual-only baselines even with added noise. For instance, CLIP achieves 19.1% with

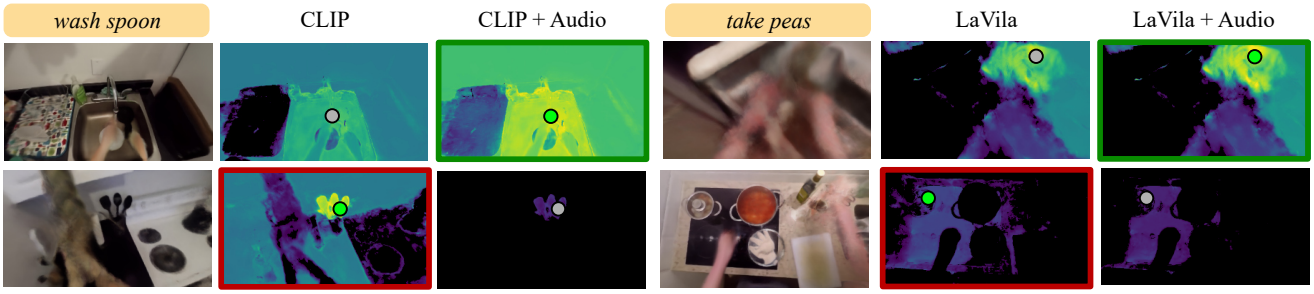

Figure 4. **Audio improves spatio-temporal action localization: additional results.** Left: Query "wash spoon" with CLIP. Without audio (middle), CLIP localizes spoons in a static scene; with audio (right), the action is correctly identified. Right: Query "take peas" with LaViLa. Audio helps suppress noise artifacts (middle vs. right), enabling precise spatio-temporal localization.

Table 3. **Audio Window Ablation.** Impact of window length on per-frame audio encoding. Spatio-temporal mean accuracies (Acc, Acc5) for action localization across CLIP, LaViLa, and EgoVideo models with different audio window sizes (1s, 2.5s, 3.5s). Row with gray background indicates the configuration used in our final method.

	CLIP		LaViLa		EgoVideo	
	Acc	Acc5	Acc	Acc5	Acc	Acc5
1	12.0	22.5	19.5	35.7	13.6	35.7
2.5	19.5	24.0	19.5	29.7	16.6	38.7
3.5	13.6	28.5	18.0	35.7	13.6	35.7

Table 4. **Text Prompt Ablation.** Impact of description type for audio feature relevance for action localization. Baseline: without audio. Action Desc.: using action descriptions. GPT Desc.: GPT-4 generated descriptions. Sound Desc.: sound-specific descriptions. Row with gray background indicates the configuration used in our final method.

	CLIP		LaViLa		EgoVideo	
	Acc	Acc5	Acc	Acc5	Acc	Acc5
Baseline	13.5	25.2	18.0	32.7	13.5	37.2
Action Desc.	6.1	22.6	19.5	27.1	13.6	35.9
GPT Desc.	12.0	22.4	16.4	26.8	13.5	36.9
Sound Desc.	19.5	24.0	19.5	29.7	16.6	38.7

noisy audio versus 13.5% with vision only, highlighting the robustness of our fusion strategy in noisy real-world conditions.

E.4. Audio Integration Strategy

We compare two strategies for incorporating audio: (1) *independent processing*, where audio and visual features are computed separately and combined at query time, and (2) *joint distillation*, where both modalities are distilled into the radiance field during training.

For joint distillation, we extend the NeRF to predict audio embeddings alongside visual features with an additional loss: $\mathcal{L}_a = \text{MSE}\left(\frac{\hat{E}_a}{|\hat{E}_a|_2}, \frac{e_a(t)}{|e_a(t)|_2}\right)$, where $\hat{E}_a$ is the rendered audio embedding from the NeRF and $e_a(t)$ is the target audio feature extracted from the audio encoder at time t. The total training objective becomes $\mathcal{L} = \mathcal{L}_{\text{rgb}} + \mathcal{L}_v + \mathcal{L}_a + \lambda_\sigma \mathcal{L}_\sigma$.

Table 5. **Audio Robustness to Background Noise.** Impact of synthetic white noise (SNR 15–20 dB) on spatio-temporal mean accuracies for action localization. Visual: vision-only baseline, Audio-visual (orig.): original audio conditions, Audio-visual + Noise: with added synthetic noise.

	CLIP		ImageBind		LaViLa	
Condition	Acc	Acc5	Acc	Acc5	Acc	Acc5
Visual	13.5	25.2	15.1	22.4	18.0	32.7
Audio-visual (orig.)	19.5	24.0	16.5	24.4	19.5	29.7
Audio-visual + Noise	19.1	24.0	15.1	24.1	19.5	29.7

Table 6. **Audio Integration Strategy.** Comparison of audio integration approaches for 4D action localization using LaViLa on EK-100. Row with gray background indicates our final method.

	Spatio-temporal		Temporal	
Strategy	Acc	Acc5	Acc	Acc5
Joint distillation	19.2	34.2	28.3	46.3
Independent (ours)	19.5	29.7	28.3	49.5

Table 6 shows comparable performance between both approaches. We adopt independent processing for computational efficiency: joint distillation incurs overhead during both training and inference by supervising and querying audio features at every spatial location, despite audio being temporally-global with no spatial variation. If dense spatial-audio correspondence maps could be constructed, joint distillation could become advantageous.

F. Limitations

While AViON4D demonstrates promising results for egocentric open-vocabulary 4D understanding, it still faces limitations.

Object-Centric Masking and Spatial Precision. We extract object-centric video crops using square bounding boxes around segmentation masks, which inevitably include surrounding contextual regions. This is particularly evident for surface-type masks (*e.g.*, countertops, tables), where objects placed on these surfaces are unintentionally included in the crop. While temporal localization remains accurate, spatial precision is reduced in these cases.

Limitations of Foundation and 3D Models. AViON4D inherits the limitations of pretrained foundation models for both visual and audio feature extraction and 3D reconstruction. Semantic understanding quality is bounded by encoder training data and architectural choices. Reconstruction quality depends on the 3D representation backbone. Although NeuralDiff achieves state-of-the-art results in EK-100 egocentric scenes [15], like all NeRF-based methods, it can still produce artifacts in regions with extreme motion. These dependencies are common to all methods leveraging pretrained foundation models and 3D neural representations.

Per-Scene Optimization Overhead. Our current implementation, building on NeRF-based 3D representations, requires training a separate network for each video (approximately 10 hours per scene) and rendering times of 3-10 minutes per frame on a single NVIDIA A6000 GPU. These computational costs are inherent to NeRF-based methods and limit real-world applications.

All these limitations stem primarily from the current state of foundation models and 3D representations rather than our core contribution. AViON4D is designed to be modular and representation-agnostic, enabling direct integration of alternative 3D representations (e.g., Gaussian Splatting) and improved encoders as they become available. Notably, these limitations do not affect the validity of our experimental findings or diminish the demonstrated benefits of multimodal 4D scene understanding. As these technologies mature, we expect the practical impact of these limitations to diminish substantially.

G. Broader and Societal Impacts

Our multi-modal framework enables fine-grained understanding of actions in egocentric videos, offering promising applications in domains such as assistive technologies, human-robot interaction, and activity monitoring. However, these capabilities also raise critical societal considerations.

Privacy Risks. By localizing and interpreting user behavior in 3D space over time, the system may inadvertently expose sensitive personal information, particularly in private or domestic settings. The inclusion of audio further compounds these risks by capturing ambient sounds, which may reveal conversations or background noise not intended for analysis. Such concerns highlight the need for strong safeguards around data collection, storage, and usage.

Bias Propagation. Our method relies on pre-trained vision and language models, which are known to encode societal biases based on the data they were trained on. These biases may manifest in the interpretation or prioritization of actions, potentially leading to unfair or inaccurate outcomes. Without explicit mitigation strategies, such issues could be perpetuated or amplified in downstream applications.

Responsible Deployment. As egocentric devices and multimodal AI systems become more widespread, their deployment must be guided by principles of transparency, accountability, and fairness. This includes ensuring informed consent, enabling opt-out mechanisms, and actively working to reduce bias and protect user privacy.

H. List of Assets

We provide the implementation sources and model weights for all methods used in this work:

1. OpenCLIP [7, 12] (https://github.com/mlfoundations/open_clip). The open-source implementation of OpenAI's CLIP model is primarily licensed under the MIT License.
2. SigLIP [16] (https://huggingface.co/google/siglip-base-patch16-224). The code is released under an Apache-2.0 license.
3. ImageBind [4] (https://github.com/facebookresearch/ImageBind). The code and model weights are released under the CC-BY-NC 4.0 license.
4. LaViLa [17] (https://github.com/facebookresearch/LaViLa). The code and model weights are released under a MIT License.
5. EgoVideo [11] (https://github.com/OpenGVLab/EgoVideo. The repository by OpenGVLab does not explicitly state a license. The accompanying research paper is published under the CC-BY-NC 4.0 license.
6. AutoSeg-SAM2 [18] (https://github.com/zrporz/AutoSeg-SAM2). The code is released under a MIT License.
7. SAM [9] (https://github.com/facebookresearch/segment-anything). The source code and model weights are released under the Apache License 2.0.
8. SAM2 [13] (https://github.com/facebookresearch/sam2). The source code and pre-trained model weights are released under the Apache License 2.0.

9. NeuralDiff [14] (https://github.com/dichotomies/epic-fields-rendering). The official PyTorch implementation is licensed under the MIT License. The checkpoints are provided alongside the source code covered under the same MIT License.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *ECCV*, pages 720–736, 2018. 1

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *IJCV*, 130(1):33–55, 2022. 1

[3] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. EPIC-KITCHENS VISOR Benchmark: VIdeo Segmentations and Object Relations. In *NeurIPS*, pages 13745–13758, 2022. 1

[4] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One Embedding Space To Bind Them All. In *CVPR*, pages 15180–15190, 2023. 3, 7

[5] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-Exo4D: Understanding Skilled Human Activity from First-and Third-Person Perspectives. In *CVPR*, pages 19383–19400, 2024. 1

[6] Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, and Andrew Zisserman. EPIC-Sounds: A Large-Scale Dataset of Actions that Sound. In *ICASSP*, pages 1–5, 2023. 1, 2

[7] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. 3, 7

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 4

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *ICCV*, pages 4015–4026, 2023. 3, 7

[10] OpenAI. ChatGPT: Language Models are Few-Shot Learners. https://chat.openai.com, 2023. Accessed: 2025-05-15. 4

[11] Baoqi Pei, Guo Chen, Jilan Xu, Yuping He, Yicheng Liu, Kanghua Pan, Yifei Huang, Yali Wang, Tong Lu, Limin Wang, et al. EgoVideo: Exploring Egocentric Foundation Model and Downstream Adaptation. *arXiv preprint arXiv:2406.18070*, 2024. 3, 7

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*, pages 8748–8763, 2021. 7

[13] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment Anything in Images and Videos. In *ICLR*, 2025. 3, 7

[14] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. NeuralDiff: Segmenting 3D Objects that Move in Egocentric Videos. In *3DV*, pages 910–919, 2021. 3, 8

[15] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Laina, Diane Larlus, Dima Damen, and Andrea Vedaldi. EPIC Fields: Marrying 3D Geometry and Video Understanding. *NeurIPS*, 36, 2024. 1, 3, 7

[16] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *ICCV*, pages 11975–11986, 2023. 3, 7

[17] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning Video Representations from Large Language Models. In *CVPR*, pages 6586–6597, 2023. 3, 7

[18] Zrporz. AutoSeg-SAM2, 2024. Automated image segmentation tool based on Segment Anything Model (SAM). 3, 7