ATTEST: Evidence-Grounded Reinforcement Learning for Long-Audio Temporal Understanding

Anonymous authors
Paper under double-blind review

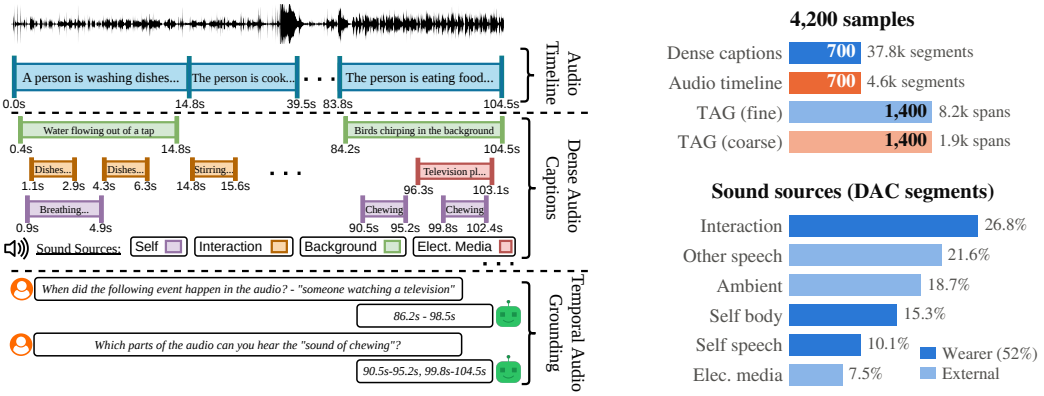


Figure 1: **EgoSCAPE benchmark overview.** *Left:* the three benchmark tasks: audio timeline generation, dense audio captioning (DAC) with sound sources, and temporal audio grounding (TAG). *Right:* samples per task and sound-source distribution.

Abstract

Temporal grounding is fundamental to audio understanding, yet large audio language models (LALMs) remain weak at it on long audio, where hallucinations compound and annotated data is scarce. Existing benchmarks cover only short clips or offer coarse, synthetic annotations. We introduce EgoSCAPE, a manually verified benchmark created from Ego4D with 700 real audios of durations up to 5 minutes, annotated with dense captions, sound sources, audio timelines, and temporal grounding queries. We further propose ATTEST, a GRPO-based reinforcement learning framework that improves temporal grounding without changing the base model's architecture or token vocabulary. ATTEST combines (i) a counterfactual grounding reward penalizing predictions that survive removal of their acoustic evidence, (ii) an attention-density reward encouraging generation to attend to its own claimed time spans, and (iii) token-wise advantages giving segment-level credit for long dense captions. ATTEST outperforms prior temporal grounding methods on existing dense captioning and grounding benchmarks, with fewer hallucinations. Together, EgoSCAPE and ATTEST establish a rigorous evaluation–training recipe that brings temporally grounded audio understanding closer to practical, long-form deployment.

1 Introduction

Temporal grounding is fundamental for building audio models for complex audio understanding and causal reasoning. *Sound event detection* (SED) deals with detecting various sound events densely within an audio along with their onset and offset timestamps (Mesaros et al., 2016; 2017; 2025). *Language-based audio moment retrieval* (also known as *temporal audio grounding* or TAG) deals with identifying where an audio query described by text occurs in an audio clip (Martín-Morató & Mesaros, 2023; Munakata et al., 2025; 2026). With the advent of large audio language models (LALMs) (Chu et al., 2024; Tang et al.,

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.

2024; Ghosh et al., 2025b; 2026; Ding et al., 2025), more recently, there has been an increased interest in *dense audio captioning* (DAC) which involves captioning all the audio events occurring in an audio along with their timestamps (Kumar et al., 2026; Shao et al., 2026; Wang et al., 2026b). Such dense captioning requires both semantic and temporal understanding about the input audio necessary for downstream reasoning in LALMs.

Prior dense audio captioning work (Wang et al., 2026b; Shi et al., 2026) targets short 10–30 second clips from datasets such as AudioSet (Gemmeke et al., 2017; Hershey et al., 2021), AudioTime (Xie et al., 2025), and TACOS (Primus et al., 2025). Some studies generate temporal annotations for longer audio synthetically, but such evaluations may not reflect real-world use (Munakata et al., 2025; Kumar et al., 2026; Sun et al., 2026). LAT-Bench (Shao et al., 2026) covers audio up to 30 minutes but provides coarse event timelines rather than dense captions (see Fig. 1), while CASTELLA (Munakata et al., 2026) spans up to 5 minutes with only coarse temporal grounding annotations.

On the modeling side, existing approaches to temporal understanding with LALMs either modify the base model's input token representation (Shi et al., 2026; Kutsakov et al., 2026) or alter its architecture and output vocabulary (Wang et al., 2026b). These changes consume extra tokens, increasing inference cost and context usage, and frequently cause catastrophic forgetting on tasks where the base model originally excelled (Sun et al., 2026; Kumar et al., 2026; Shi et al., 2026; Kutsakov et al., 2026). Furthermore, large language models (LLMs) and LALMs are prone to hallucination (Huang et al., 2025; Kuan et al., 2024; Cheng et al., 2025), which worsens in long-context settings (Hsieh et al., 2024), a problem further compounded for long-audio temporal grounding by the scarcity of data noted in the above paragraph.

To address these gaps, we introduce **EgoSCAPE**, a long-audio temporal grounding benchmark built on the Ego4D dataset (Grauman et al., 2022), with manually verified dense audio captions, coarse audio timelines, and temporal grounding annotations (see Fig. 1). It comprises 700 audio clips of up to 5 minutes each, totaling 13.15 hours. Beyond temporal captions of acoustic events, we also annotate each sound's source into one of the six categories mentioned in Fig. 1.

To improve long-audio temporal understanding in LALMs, we propose **ATTEST**, a reinforcement learning framework that mitigates hallucinations and promotes encourages predictions grounded in the audio without altering the base model's architecture or token vocabulary. ATTEST introduces three components: (i) a counterfactual reward that ensures predictions to be grounded in the acoustic events of the corresponding temporal region, (ii) an attention density reward that steers the model's attention toward the audio tokens of the time span each prediction claims during generation, and (iii) a token-wise advantage that provides fine-grained supervision for long, dense captions. Experiments on existing dense captioning and temporal grounding benchmarks, as well as EgoSCAPE, show that ATTEST outperforms prior temporal grounding methods by significant margins while producing hallucination-free predictions.

In summary, our main contributions are: (i) **EgoSCAPE**, a new benchmark for dense audio captioning, audio timeline generation, and temporal audio grounding, built from a subset of Ego4D (Grauman et al., 2022) with audio of up to 5 minutes; (ii) **ATTEST**, a novel RL framework that mitigates hallucinations and aligns the model's attention with the time spans it predicts; and (iii) extensive experiments showing that ATTEST outperforms existing baselines on dense audio captioning and temporal audio grounding without regressing the base model's performance on general audio understanding.

2 Related Works

Large audio language models and long audio. LALMs augment text-only LLMs by accepting an audio input through an audio encoder (Radford et al., 2023; Chen et al., 2023) and training on a large audio-text corpus (Tang et al., 2024; Chu et al., 2024; Kong et al., 2024; Ghosh et al., 2025b;a; 2026). Moreover, omni LLMs extend this recipe to accept multiple modalities including audio and text, showing better performance than audio-only

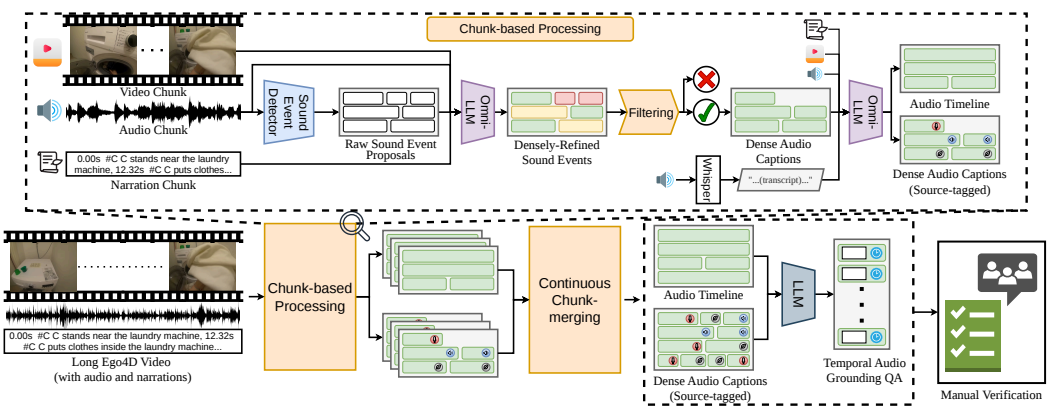


Figure 2: **EgoSCAPE benchmark creation pipeline.** To ensure high-quality annotations, we first chunk the audios that are automatically annotated using an omni-LLM with additional supervision from the video and narrations available for Ego4D. All the generated samples for the benchmark are manually verified and corrected at the end.

models on various benchmarks (Xu et al., 2025a;b). Evaluation of these models has largely been using short audio benchmarks such as MMAR (Ma et al., 2025), MMAU (Sakshi et al., 2025) and MMSU (Wang et al., 2026a). There has been recent focus on long-audio processing where benchmarks such as AudioMarathon (He et al., 2025), BLAB (Ahia et al., 2025), and VoiceGiraffe (Ye et al., 2026) show challenges to long audio processing and AudioFlamingo series of models (Ghosh et al., 2025b;a; 2026) support long audios by training on large-scale synthetic datasets. Our benchmark and proposed RL framework tackle the specific challenge of long-audio temporal grounding and understanding.

Temporal audio grounding. While temporal grounding tasks such as sound event detection have long been prominent (Mesaros et al., 2017; 2025; Munakata et al., 2025; 2026), use of multimodal LLMs for temporal grounding and dense captioning is fairly recent (Ghosh et al., 2024; Wang et al., 2026b; Shao et al., 2026; Shi et al., 2026). TimeAudio (Wang et al., 2026b) introduces temporal markers in the model vocabulary to improve DAC and TAG for short audios. TimePro-RL (Shi et al., 2026) uses audio-side time tokens and RL to improve DAC and TAG for short utterances. SpotSound (Sun et al., 2026) and TAC (Kumar et al., 2026) utilize SFT training over synthetic datasets to improve performance on TAG and DAC respectively. While most of these baselines deal with short audios of around 30 seconds, LAT-Audio (Shao et al., 2026) introduces an agentic framework for audio timeline generation and TAG for audios up to 30 minutes. GigaChat-Audio (Kutsakov et al., 2026) performs TAG on speech up to 2 hours. Compared to these approaches, EgoSCAPE benchmark consists of DAC with sound source, Audio Timeline, and TAG annotations for audios up to 5 minutes. Moreover, the proposed ATTEST framework improves temporal understanding without modifying model vocabulary or consuming additional input context via time-tokens.

3 EgoSCAPE - Egocentric Sound CAPtion of Events

We describe the benchmark creation pipeline for EgoSCAPE in this section as shown in Fig. 2. Compared with other benchmarks of temporal audio grounding (see §B.1), our benchmark is rich in annotations as it contains dense annotation over long audio clips.

3.1 Automatic Benchmark Creation

Traditional SED models (Chen et al., 2023; Shao et al., 2024a) predict accurate temporal boundaries but are restricted to a fixed label set (typically AudioSet (Gemmeke et al., 2017) or ESC-50 (Piczak, 2015)), limiting annotation richness. Omni-LLMs, in contrast, generate open-ended annotations but their temporal boundaries often drift over long multimodal inputs (Shao et al., 2026). To combine their strengths, we split Ego4D videos into 30-second chunks, obtain SED labels over AudioSet classes using a BEATs-based detector

(Chen et al., 2023), and use an omni-LLM to verify and refine them (prompt in Tab. 16), discarding chunks where over 30% of events are marked incorrect. A second omni-LLM pass annotates sound sources and generates a coarse audio timeline (prompts in Tabs. 17 and 18), incorporating Whisper transcripts (Radford et al., 2023) for speech content. Both passes also take the Ego4D video and textual narrations as input to help disambiguate events and sources.

We then merge consecutive chunk annotations into a single annotation per audio file, using an LLM to correctly join events spanning chunk boundaries in the timelines (prompt in Tab. 19). Finally, an LLM generates temporal grounding queries: fine-grained queries about atomic events from the dense captions, and coarse queries about multi-sound acoustic scenes from the timelines (prompts in Tabs. 20 and 21). We use Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025b) as the omni-LLM and Qwen3-235B-A22B-Instruct (Yang et al., 2025) as the LLM throughout the pipeline.

3.2 Human Verification and Benchmark Stats

To ensure high data quality, each generated annotation in EgoSCAPE is verified by three human annotators (details in §B.3), and annotations flagged as incorrect are corrected accordingly. The final benchmark comprises 700 unique audio clips drawn from 121 unique Ego4D videos, totaling approximately 13.15 hours. Further benchmark statistics are provided in §B.2.

4 ATTEST: ATTention and Evidence-grounded Segment Training

We propose **ATTEST**, an RL-based framework that modifies Group Relative Policy Optimization (GRPO) (Shao et al., 2024b) to improve temporal audio grounding over long audios as described in this section. We perform RL only on the dense audio captioning and audio timeline generation tasks and show that the temporal grounding capabilities transfer to other tasks as well in §5.2 & §5.4.

Notation. Let a denote the input audio of duration T_{aud} and x the full input token sequence (audio tokens $\mathcal{T}_{\text{aud}}$ and text prompt tokens $\mathcal{T}_{\text{tex}}$). The ground truth is a set of N_g segments (or triplets) $D = \{(s_i, e_i, c_i)\}_{i=1}^{N_g}$, where (s_i, e_i) are the start/end timestamps of an event described by caption c_i. For a rollout g sampled from $\pi_{\theta_{\text{old}}}(\cdot \mid x)$, the prediction is $\hat{D}^{(g)} = \{(\hat{s}_j^{(g)}, \hat{e}_j^{(g)}, \hat{c}_j^{(g)})\}_{j=1}^{N_p^{(g)}}$, decoded from the fixed output format *"0.1 - 2.2, person speaking, 1.4 - 3.8, birds chirping, ..."*. We write $\mathcal{G}_j^{(g)}$ for the set of generated token positions that spell out the j-th predicted triplet, and $\mathcal{T}(s, e) \subseteq \mathcal{T}_{\text{aud}}$ for the audio tokens whose receptive field overlaps the time interval $[s, e]$. $\text{Sim}(\cdot, \cdot) \in [0, 1]$ is the FENSE caption score (Zhou et al., 2022) and tIoU the temporal intersection-over-union. Throughout, a superscript (g) marks a quantity belonging to rollout g; we suppress it inside expressions that are defined for a single generic rollout.

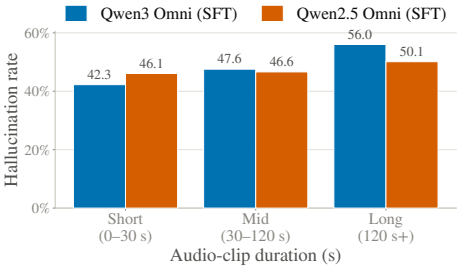


Figure 3: Hallucination for difft. audio durations in EgoSCAPE (details - §C.4).

4.1 Counterfactual Audio-Grounding (CAG)

Our preliminary experiments on EgoSCAPE (Fig. 3) show that the hallucination rate (defined at §C.2.1) rises on longer audio clips. This suggests that the model is increasingly recovering plausible events from linguistic and dataset priors instead of grounding events in the audio. To mitigate this behavior, we propose the following intervention and reward.

Intervention. For each training sample we modify a target fraction p of the clip. We draw K non-overlapping windows $\mathcal{W} = \{w_1, \ldots, w_K\}$, $w = (t_{ms}^w, t_{me}^w)$, uniformly at random

in time. We apply a corruption operator $M(\cdot)$ to obtain a' such that the original audio events in $\mathcal{W}$ are no longer present. We then decode a *single* greedy counterfactual reference $\hat{D}' = \pi_{\theta_{\text{old}}}(a')$, which is shared across the GRPO rollout group, and score each of the G sampled rollouts $\hat{D}^{(g)}$ on the original audio a against it as described below.

Reward. Let $\hat{D}_w^{(g)} = \{\hat{d}_j^{(g)} : [\hat{s}_j^{(g)}, \hat{e}_j^{(g)}] \subseteq w\}$ where $\hat{d}_j^{(g)}$ is the segment triplet $(\hat{s}_j^{(g)}, \hat{e}_j^{(g)}, \hat{c}_j^{(g)})$ and $\subseteq$ denotes containment of the predicted support in the window. Similarly, we can define $\hat{D}'_w$ for the counterfactual reference. For each $j \in \hat{D}_w^{(g)}$ we measure how strongly the prediction *survives* the intervention and penalize it as follows:

$$\chi_j^{(g)} = \underbrace{-\max_{\hat{d}' \in \hat{D}'_w} \left[\text{tIoU}\big(\hat{d}_j^{(g)}, \hat{d}'\big) \cdot \text{Sim}\big(\hat{d}_j^{(g)}, \hat{d}'\big) \right]}_{\text{segment-level reward}} \tag{1}$$

Averaging the per-segment penalties within a window gives the window score $q_w^{(g)}$, and averaging over the K windows gives the response-level reward:

$$q_w^{(g)} = \frac{1}{\max\big(1, |\hat{D}_w^{(g)}|\big)} \sum_{j \in \hat{D}_w^{(g)}} \chi_j^{(g)}, \qquad R_{\text{CAG}}^{(g)} = \frac{1}{K} \sum_{w \in \mathcal{W}} q_w^{(g)}. \tag{2}$$

$\chi_j^{(g)} \in [-1, 0]$ varies across rollouts even though $\hat{D}'$ is fixed, since each rollout agrees with the counterfactual reference to a different degree within each window. We found the silence operator to work best for $M(\cdot)$ as mentioned in §D.4.

4.2 Attention Density Grounding

Because generation is autoregressive and the output format is fixed, the model should attend more strongly to the audio tokens $\mathcal{T}(\hat{s}_j, \hat{e}_j)$ while it emits the triplet for $[\hat{s}_j, \hat{e}_j]$. We verify this empirically in §D.6. Among samples matched in segment count and duration, those with higher FENSE show consistently higher attention density on their claimed spans.

Let $\bar{A}_{t,u}$ be the attention weight from generated token t to input token u. We define the attention density of predicted segment j as the fraction of audio-directed attention mass that falls inside its own temporal support, normalized by the fraction $f_j = (\hat{e}_j - \hat{s}_j)/T_{\text{aud}}$ of the audio clip that the segment occupies, and the corresponding response-level reward as the mean density over the predicted segments:

$$\rho_j = \underbrace{\frac{1}{f_j} \cdot \frac{1}{|\mathcal{G}_j|} \sum_{t \in \mathcal{G}_j} \frac{\sum_{u \in \mathcal{T}(\hat{s}_j, \hat{e}_j)} \bar{A}_{t,u}}{\sum_{u \in \mathcal{T}_{\text{aud}}} \bar{A}_{t,u}}}_{\text{segment-level reward}}, \qquad R_{\text{AD}} = \frac{1}{N_p} \sum_{j=1}^{N_p} \rho_j. \tag{3}$$

To ensure that the model does not hack R_{AD} by predicting short segments, we cap ρ to a maximum of 5. We use the last layer's attention averaged across all heads to compute $\bar{A}_{t,u}$, based on empirical findings in Tab. 14 (§D.4).

Correctness Reward (R_{DAC}). In addition to the above rewards, we also apply a correctness reward for dense captioning which directly utilizes the ground-truth annotations. For each rollout g we solve a Hungarian assignment between $\hat{D}^{(g)}$ and D maximizing

$$\text{Score}(i, j) = \alpha \, \text{tIoU}\big((s_i, e_i), (\hat{s}_j, \hat{e}_j)\big) + (1 - \alpha) \, \text{Sim}(c_i, \hat{c}_j), \tag{4}$$

yielding an assignment $\sigma^{(g)}(j) \in \{0, 1, \ldots, N_g\}$, where $\sigma^{(g)}(j) = 0$ marks an unmatched (spurious) prediction. The segment-level reward is the matched payoff itself,

$$r_j^{(g)} = \begin{cases} \alpha \, \text{tIoU}\big((s_{\sigma^{(g)}(j)}, e_{\sigma^{(g)}(j)}), (\hat{s}_j^{(g)}, \hat{e}_j^{(g)})\big) + (1 - \alpha) \, \text{Sim}\big(c_{\sigma^{(g)}(j)}, \hat{c}_j^{(g)}\big), & \sigma^{(g)}(j) \neq 0, \\ 0, & \sigma^{(g)}(j) = 0. \end{cases} \tag{5}$$

where the final correctness reward is the average of $r_j^{(g)}$ over all the GT segments, i.e. $R_{\text{DAC}}^{(g)} = \frac{1}{N_g} \sum_{j=1}^{N_p^{(g)}} r_j^{(g)}$.

4.3 Token-wise Fine-grained Advantage

Prior work on dense audio captioning (Shao et al., 2026) assigns a single scalar reward to the whole response. The direct analogue here mixes the three response-level rewards of Eqs. (2), (3) and (5) with a format reward $R^{(g)}_{\text{fmt}}$, and normalizes the mixture group-wise as in standard GRPO:

$$R^{(g)}_{\text{seq}} = \lambda_1 R^{(g)}_{\text{CAG}} + \lambda_2 R^{(g)}_{\text{AD}} + \lambda_3 R^{(g)}_{\text{DAC}} + \lambda_4 R^{(g)}_{\text{fmt}}, \qquad A^{(g)}_{\text{seq}} = \frac{R^{(g)}_{\text{seq}} - \text{mean}_g(R_{\text{seq}})}{\text{std}_g(R_{\text{seq}}) + \epsilon}, \tag{6}$$

assigning $\hat{A}^{(g)}_t = A^{(g)}_{\text{seq}}$ to every token. This provides a single coarse, sequence-level reward for the entire rollout. This issue is exacerbated in the case of long audios where the model response for captioning is inherently long leading to poor supervision. We report this variant as an ablation in §5.3.

Anchored per-segment advantages. We propose more fine-grained advantage computation for each of R_{CAG}, R_{AD} and R_{DAC}, instead of a sequence-level advantage in Eq. (6). Each reward is an average of per-segment scores (underbraced in Eqs. (1), (3) and (5)), so we normalize the segment-level rewards directly and let each token inherit the advantage of the segment it spells out. For GRPO advantage calculation, we define different anchors for the rewards as below.

R_{DAC}: GT-anchored. We normalize the segments across the GT-segments they are matched to according to the inverse assignment of $\sigma^{(g)}$ and assign a reward 0 from a rollout that missed that GT-segment. For GT event i, let $\varphi_i(g)$ be the index of the segment in rollout g matched to i, and define the per-rollout score $u^{(g)}_i$ as the segment-level reward $r^{(g)}_{\varphi_i(g)}$ so that $\{u^{(g)}_i\}_{g=1}^{G}$ always contains exactly G entries. The segment advantage is then,

$$A^{\text{DAC},(g)}_i = \frac{u^{(g)}_i - \mu_i}{\varsigma_i + \epsilon}, \qquad \mu_i = \frac{1}{G}\sum_g u^{(g)}_i, \quad \varsigma_i = \text{std}_g\big(u^{(g)}_i\big). \tag{7}$$

This compares each rollout against the other rollouts *on the same event*, so credit is not diluted by how easy or hard the remaining events in the clip happen to be.

R_{CAG}: window-anchored. The counterfactual reference $\hat{D}'$ is decoded once and shared across the group, so the silenced windows $\mathcal{W}$ are fixed and independent of any individual rollout. For window w, we take the window score $q^{(g)}_w$ of Eq. (2) and normalize it within that window,

$$A^{\text{CAG},(g)}_w = \frac{q^{(g)}_w - \mu_w}{\varsigma_w + \epsilon}, \qquad \mu_w = \tfrac{1}{G}\textstyle\sum_g q^{(g)}_w, \quad \varsigma_w = \text{std}_g\big(q^{(g)}_w\big). \tag{8}$$

Localizing the penalty this way means a rollout that hallucinates inside one silenced window is not down-weighted on the segments it predicts correctly elsewhere in the clip.

R_{AD}: group-pooled. Attention density is defined on a rollout's own predicted span and admits no shared anchor across rollouts. We therefore pool the densities over all predicted segments of all G rollouts, $\mathcal{S} = \{(g, j) : 1 \le g \le G,\ 1 \le j \le N^{(g)}_p\}$, and standardize against that pool:

$$A^{\text{AD},(g)}_j = \frac{\rho^{(g)}_j - \mu_{\text{AD}}}{\varsigma_{\text{AD}} + \epsilon}, \quad \mu_{\text{AD}} = \frac{1}{|\mathcal{S}|}\sum_{(g,j)\in\mathcal{S}} \rho^{(g)}_j, \quad \varsigma_{\text{AD}} = \text{std}_{(g,j)\in\mathcal{S}}\big(\rho^{(g)}_j\big). \tag{9}$$

Each segment is thus scored against the density the group achieves on average, so segments that attend to their own claimed time span more sharply than the group norm are reinforced individually rather than through a response-level average.

Token assignment. Every token inside the span $\mathcal{G}^{(g)}_j$ inherits the three segment-level advantages of its segment; format tokens receive only the sequence-level term $A^{(g)}_{\text{seq}}$, now

Table 1: Dense audio captioning results on EgoSCAPE and AudioSet-SL datasets comparing ATTEST to other audio and omni LLM baselines. EvtF1@0.5 and SegF1 are micro. Hal% and SrcF1@0.5 are macro. * denotes sliding-window evaluation described in §C.3.

Model	EgoSCAPE Dense Audio Captioning							AudioSet-SL					
	F@0.3	F@0.5	F@0.7	EvtF1	SegF1	Hal%↓	SrcF1	F@0.3	F@0.5	F@0.7	EvtF1	SegF1	Hal%↓
Proprietary Models													
Gemini 3.1 Pro	0.235	0.213	0.201	0.084	0.276	31.1	0.371	0.207	0.192	0.181	0.079	0.126	22.7
Gemini 3.6 Flash	0.242	0.215	0.200	0.091	0.197	38.9	0.351	0.217	0.200	0.172	0.037	0.066	30.3
GPT 4o Audio	0.226	0.208	0.198	0.059	0.186	48.4	0.359	0.208	0.184	0.167	0.051	0.058	50.9
Captioning Models													
AF-Next Cap.	0.244	0.218	0.201	0.099	0.147	52.7	0.325	0.239	0.207	0.184	0.058	0.087	44.6
Qwen3 Omni Cap.	0.270	0.232	0.206	0.257	0.335	64.4	0.369	0.257	0.220	0.190	0.048	0.057	63.6
Audio and Omni LLMs													
Qwen2 Audio*	0.207	0.199	0.196	0.048	0.127	51.0	0.232	0.207	0.191	0.182	0.054	0.076	43.1
Kimi-Audio*	0.188	0.183	0.177	0.042	0.034	42.9	0.207	0.224	0.198	0.175	0.061	0.107	33.9
TimeAudio*	0.340	0.310	0.271	0.256	0.344	35.2	-	0.426	0.389	0.352	0.353	0.372	36.3
AudioFlamingo 2	0.193	0.193	0.192	0.004	0.001	59.7	-	0.192	0.176	0.167	0.079	0.084	55.5
AF-Next	0.181	0.178	0.176	0.022	0.014	45.9	0.030	0.226	0.202	0.184	0.089	0.114	48.5
Qwen2.5 Omni	0.178	0.177	0.176	0.021	0.035	48.2	0.165	0.220	0.202	0.192	0.124	0.190	20.5
+ SFT	0.412	0.367	0.313	0.320	0.303	47.2	0.564	0.448	0.403	0.362	0.336	0.337	25.8
+ TimePro-RL	0.445	0.410	0.358	0.349	0.348	31.3	0.642	0.475	0.430	0.385	0.347	0.370	20.1
+ ATTEST	**0.461**	**0.427**	**0.372**	**0.371**	**0.355**	**30.5**	**0.656**	**0.489**	**0.442**	**0.399**	**0.354**	0.389	**18.2**
AudioFlamingo 3	0.196	0.194	0.193	0.042	0.010	40.3	-	0.204	0.189	0.177	0.119	0.131	32.4
+ SFT	0.376	0.344	0.298	0.258	0.248	44.4	0.587	0.453	0.406	0.362	0.345	0.396	36.3
+ TimePro-RL	0.421	0.388	0.328	0.335	0.329	34.0	0.620	0.489	0.440	0.384	0.369	0.385	29.2
+ ATTEST	**0.437**	**0.400**	**0.342**	**0.346**	**0.345**	**31.8**	**0.641**	**0.503**	**0.451**	**0.398**	**0.380**	**0.400**	**25.3**

obtained from Eq. (6) with $\lambda_1 = \lambda_2 = \lambda_3 = 0$, i.e. by group-normalizing $R^{(g)}_{\text{fmt}}$ alone:

$$\hat{A}^{(g)}_t \;=\; A^{(g)}_{\text{seq}} \;+\; \sum_{j=1}^{N_p^{(g)}} \mathbb{1}\big[t \in \mathcal{G}^{(g)}_j\big] \Big(\eta_1 A^{\text{CAG},(g)}_{w(j)} \;+\; \eta_2 A^{\text{AD},(g)}_{j} \;+\; \eta_3 A^{\text{DAC},(g)}_{\sigma^{(g)}(j)} \Big), \tag{10}$$

where $w(j) \in \mathcal{W}$ is the silenced window containing segment j. The weights η_1, η_2, η_3 are hyperparameters to balance the three signals. Finally, ATTEST optimizes the clipped GRPO objective with these token-wise advantages:

$$\mathcal{J}(\theta) = \mathbb{E}\Big[\tfrac{1}{G} \sum_{g=1}^{G} \tfrac{1}{|o^{(g)}|} \sum_{t=1}^{|o^{(g)}|} \min\big(\varrho^{(g)}_t \hat{A}^{(g)}_t,\; \text{clip}(\varrho^{(g)}_t, 1-\varepsilon_{\text{clip}}, 1+\varepsilon_{\text{clip}}) \hat{A}^{(g)}_t \big) \Big] \;-\; \kappa\, \mathbb{D}_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}]\,, \tag{11}$$

where $\varrho^{(g)}_t = \pi_\theta(o^{(g)}_t \mid x, o^{(g)}_{<t}) / \pi_{\theta_{\text{old}}}(o^{(g)}_t \mid x, o^{(g)}_{<t})$.

5 Experiments

Setup. We use Qwen2.5 Omni (Xu et al., 2025a) and AudioFlamingo 3 (Ghosh et al., 2025a) as base models (with Qwen3 Omni (Xu et al., 2025b) in the Appendix). RL training uses the AudioSet-SL training set (Hershey et al., 2021), supplemented with long audio generated via the pipeline in §3 from Ego4D videos disjoint from EgoSCAPE. We set $(\eta_1, \eta_2, \eta_3) = (1, 0.2, 1)$ and $\lambda_4 = 1$; the sequence-level ablation reuses these weights as $(\lambda_1, \lambda_2, \lambda_3)$. We apply GRPO with a rollout size of 8 and an effective batch size of 16, using LoRA finetuning at a learning rate of 5×10^{-6} (details in §C.1).

Evaluation. We evaluate on temporal grounding and general audio understanding benchmarks. For dense audio captioning, we report results on the AudioSet-SL test set (Hershey et al., 2021) following TimeAudio (Wang et al., 2026b), alongside EgoSCAPE. For temporal audio grounding, we report zero-shot results on Clotho-Moment (Munakata et al., 2025), UnAV-100 (Geng et al., 2023) subset from Munakata et al. (2025) and CASTELLA (Munakata et al., 2026). For general audio understanding, we use MMAR (Ma et al., 2025) & LongAudioBench (Ghosh et al., 2025b).

Metrics. For dense captioning and audio timeline generation, we report FENSE (Zhou et al., 2022) scores at IoU thresholds of 0.3, 0.5, and 0.7 following Shao et al. (2026), along with Event-F1 and Segment-F1 (Mesaros et al., 2016). Following TAC (Kumar et al., 2026), we use FLAM (Wu et al., 2025) to measure hallucination rates over predicted segments unmatched to any ground truth: a segment is deemed hallucinated if its *confidence*, the maximum FLAM similarity between its audio frames and its caption, falls below 0.15.

Table 2: Zero-shot temporal audio grounding results on Clotho-Moment, UnAV-100 and CASTELLA datasets comparing ATTEST to other audio and omni LLM baselines.

Model	Clotho-Moment			UnAV-100			CASTELLA		
	R@0.5	R@0.7	mIoU	R@0.5	R@0.7	mIoU	R@0.5	R@0.7	mIoU
Proprietary Models									
Gemini 3.1 Pro	0.385	0.296	0.344	0.340	0.240	0.294	0.080	0.041	0.081
Gemini 3.6 Flash	0.581	0.431	0.536	0.570	0.410	0.524	0.233	0.124	0.223
GPT 4o Audio	0.106	0.031	0.185	0.200	0.060	0.216	0.042	0.018	0.062
Finetuned Temporal Grounding Models									
AM-DETR	0.882	0.827	0.809	0.580	0.410	0.428	0.094	0.062	0.103
SpotSound-A	0.908	0.853	0.868	0.780	0.620	0.725	0.180	0.115	0.181
UVCOM	0.885	0.832	0.820	0.590	0.360	0.518	0.302	0.191	0.317
Audio and Omni LLMs									
Qwen2 Audio*	0.043	0.005	0.131	0.080	0.040	0.181	0.053	0.020	0.097
Kimi-Audio*	0.088	0.025	0.164	0.180	0.030	0.265	0.080	0.022	0.146
AudioFlamingo 2	0.009	0.003	0.023	0.040	0.010	0.056	0.011	0.005	0.016
AF-Next	0.004	0.002	0.019	0.030	0.030	0.069	0.008	0.002	0.013
Qwen2.5 Omni	0.409	0.141	0.426	0.470	0.280	0.458	0.092	0.048	0.116
+ SFT	0.875	0.821	0.830	0.790	0.600	0.693	0.224	0.123	0.269
+ TimePro-RL	0.895	0.844	0.859	0.800	0.620	0.711	0.252	0.143	0.295
+ ATTEST	**0.909**	**0.859**	**0.876**	**0.820**	**0.650**	**0.729**	**0.297**	**0.186**	**0.312**
AudioFlamingo 3	0.142	0.063	0.163	0.120	0.030	0.147	0.008	0.004	0.013
+ SFT	0.850	0.754	0.801	0.780	0.620	0.708	0.147	0.095	0.170
+ TimePro-RL	0.892	0.799	0.836	0.810	0.650	0.744	0.168	0.110	0.189
+ ATTEST	**0.903**	**0.819**	**0.854**	**0.850**	**0.660**	**0.753**	**0.202**	**0.123**	**0.266**

For EgoSCAPE, we additionally report sound source detection F1. For temporal audio grounding, we report recall at IoU thresholds of 0.5 and 0.7 and mean IoU (more in §C.2).

Baselines. We compare against Gemini 3.1 Pro (Google DeepMind, 2026a), Gemini 3.6 Flash (Google DeepMind, 2026b), GPT 4o Audio (OpenAI et al., 2024), Qwen2 Audio (Chu et al., 2024), Kimi-Audio (Ding et al., 2025), TimeAudio (Wang et al., 2026b), AudioFlamingo 2, 3 & Next (Ghosh et al., 2025b;a; 2026), Qwen2.5 Omni (Xu et al., 2025a), and Qwen3 Omni (Xu et al., 2025b). As Qwen2 Audio, Kimi-Audio, and TimeAudio cannot process 5-minute audio, we use a sliding-window approach. We also compare with TimePro-RL (Shi et al., 2026) framework adapted for our setup (details in §C.3).

5.1 Audio Captioning

Tab. 1 (extended in §D.1 Tab. 7) reports DAC results on EgoSCAPE and AudioSet-SL (Wang et al., 2026b). On both datasets, ATTEST achieves the best DAC performance, substantially improving over its base models while also markedly reducing hallucinations. Dedicated captioning models outperform most off-the-shelf audio and omni LLMs but hallucinate the most, whereas proprietary models hallucinate far less, with Gemini 3.1 Pro achieving the lowest hallucination rate among baselines on EgoSCAPE. Audio timeline generation results (§D.1 Tab. 8) follow similar trends.

5.2 Temporal Audio Grounding

Tab. 2 (extended in §D.2 Tab. 9) reports zero-shot TAG results on Clotho-Moment, UnAV-100, and CASTELLA. ATTEST substantially improves over its base models and outperforms all LLM-based baselines. Despite never seeing these benchmarks' training sets, it matches or surpasses dedicated grounding methods fine-tuned on them (Munakata et al., 2025; 2026; Sun et al., 2026), demonstrating strong generalization. Results on EgoSCAPE TAG samples (§D.2 Tab. 10) show similar trends.

5.3 Ablations

Role of individual rewards. Tab. 3 ablates the components of ATTEST (§4), reporting FENSE at tIoU 0.5 and hallucination rates on all EgoSCAPE clips and on long clips (>120s). Removing R_{AD} or R_{CAG} substantially increases hallucinations and lowers FENSE. Ablating

Table 3: Ablation study to show the effect of different techniques and rewards used in ATTEST framework. TA: Token-wise fine-grained advantage.

Model	TA	R_{CAG}	R_{AD}	R_{DAC}	EgoSCAPE Dense Captioning				Temporal Grounding	
					Overall		Long (>120s)		Clotho-Moment	CASTELLA
					F@0.5	Hal%↓	F@0.5	Hal%↓	R@0.5	R@0.5
Qwen2.5 Omni	-	-	-	-	0.177	48.2	0.169	55.2	0.409	0.092
+ SFT	-	-	-	-	0.367	47.2	0.321	50.1	0.875	0.224
+ Vanilla GRPO	✗	✗	✗	✓	0.395	38.2	0.379	41.2	0.879	0.236
+ ATTEST	✓	✓	✓	✓	**0.427**	**30.5**	**0.414**	**34.2**	**0.909**	**0.297**
w/o R_{CAG}	✓	✗	✓	✓	0.405	37.2	0.382	38.4	0.898	0.252
w/o R_{AD}	✓	✓	✗	✓	0.409	33.7	0.385	35.8	0.896	0.260
w/o R_{DAC}	✓	✓	✓	✗	0.382	34.1	0.375	36.5	0.881	0.241
w/o Token-adv.	✗	✓	✓	✓	0.411	32.1	0.398	34.9	0.901	0.268

R_{DAC}, which jointly optimizes text similarity and tIoU, causes the largest FENSE drop, while replacing token-wise with sequence advantage degrades performance across all metrics.

Hyper-parameter and reward design. Effects of the reward weights (η_1, η_2, η_3), the corruption operator in R_{CAG}, and the attention layer in R_{AD} are analyzed in §D.4.

5.4 General audio benchmarks

Table 4: LongAudioBench and MMAR results.

Model	MMAR		LongAudioBench	
	All	Temporal	All	Temporal
Qwen2.5 Omni	63.2	56.0	61.5	60.1
+ TimePro-RL	61.0	55.4	58.3	59.5
+ ATTEST	**64.4**	**58.2**	**62.8**	**62.2**
+ ATTEST w/ markers	62.2	56.9	60.5	61.2
AudioFlamingo 3	60.4	53.2	57.7	53.9
+ TimePro-RL	58.3	51.0	55.2	52.5
+ ATTEST	**62.1**	**57.9**	**59.6**	**57.2**
+ ATTEST w/ markers	59.2	52.8	56.9	53.5

Tab. 4 (extended in Tab. 11) reports results on MMAR and LongAudioBench. As neither benchmark tags temporal samples explicitly, we use an LLM to identify a temporal subset and report it separately. ATTEST improves overall performance, with larger gains on the temporal subset, indicating that its temporal improvements transfer to general audio understanding. In contrast, TimePro-RL degrades performance under identical training data and base models, and so does an ATTEST variant with TimePro-RL-style temporal markers, suggesting that temporal markers hurt original audio understanding capabilities of different models.

5.5 Analysis

Robustness to hallucination. To test robustness to audio intervention, we randomly replace segments of EgoSCAPE audio with AudioSet-SL clips and compute FENSE@0.5 on the replaced segments (Fig. 4). ATTEST-tuned models accurately predict sound events and boundaries even for such out-of-context segments, whereas SFT-tuned models degrade significantly. Similar analysis on silence intervention is present in §D.5.

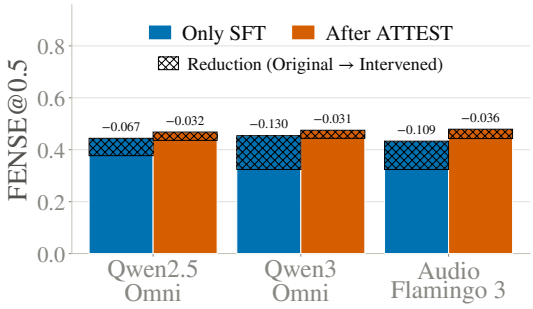


Figure 4: Perf. drop on intervened clips.

Attention analysis and importance of real data. In §D.6, we show how attention patterns of the model differ after training with ATTEST. Additionally, we show the importance of real long data over synthetic data augmentation in §D.7.

6 Conclusion

We introduced EgoSCAPE, a long-audio benchmark for evaluating temporal grounding and captioning in LALMs, and ATTEST, an RL framework for improving their temporal grounding. ATTEST grounds responses in audio evidence via a counterfactual reward, steers attention toward the audio tokens each prediction describes, and provides fine-grained credit assignment through token-wise advantages. We believe this work marks a significant step toward stronger temporal grounding in LALMs, enabling better audio understanding and reasoning in general scenarios.

AI Use Statement

In this work, we used generative AI tools for generating synthetic data sets, support qualitative and thematic data analysis, and assisting with cleaning and reformatting the dataset. We have not used generative AI tools for developing theoretical models or conceptual frameworks, implementing methods, proposing or refining hypotheses, designing or providing feedback on research methodology or experiment, or for interpreting the results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims and assisting in writing of proofs are not applicable to this work.

Additionally, we have used generative AI tools for summarizing and analysing existing literature. The generated literature survey was manually verified for the correctness of citations and relevance to this work. We have also used AI tools for checking the correctness of grammar and language of this research paper.

Ethics Statement

EgoSCAPE is built on Ego4D (Grauman et al., 2022), which was collected with informed consent from camera wearers and under the ethics review of its contributing institutions. We use it under the Ego4D license and release only our annotations with clip identifiers, not the underlying audio or video. Our annotations can not identify individual speakers. We acknowledge that improved temporal audio grounding could be misused for audio surveillance; our models are trained for sound-event captioning and grounding rather than speaker identification, and their outputs should not be used in high-stakes settings without human review.

Reproducibility Statement

We use open-weight models to construct EgoSCAPE so that the benchmark creation pipeline remains reproducible. All experimental details are provided in §5 and §C. Moreover, we demonstrate the efficacy of the proposed ATTEST method on open-source and open-weight models to ensure ease of access. All prompts used throughout this work are provided in §E. The training code and the EgoSCAPE benchmark annotations will be made publicly available upon acceptance.

References

Orevaoghene Ahia, Martijn Bartelds, Kabir Ahuja, Hila Gonen, Valentin Hofmann, Siddhant Arora, Shuyue Stella Li, Vishal Puttagunta, Mofetoluwa Adeyemi, Charishma Buchireddy, et al. Blab: Brutally long audio bench. *arXiv preprint arXiv:2505.03054*, 2025.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. BEATs: Audio pre-training with acoustic tokenizers. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5178–5193. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/chen23ag.html.

Xize Cheng, Dongjie Fu, Chenyuhao Wen, Shannon Yu, Zehan Wang, Shengpeng Ji, Siddhant Arora, Tao Jin, Shinji Watanabe, and Zhou Zhao. AHa-bench: Benchmarking audio hallucinations in large audio-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=vCej5sO61x.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

J L Fleiss. Measuring nominal scale agreement among many raters. *Psychol. Bull.*, 76(5): 378–382, November 1971.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780. IEEE Press, 2017. doi: 10.1109/ICASSP.2017.7952261. URL https://doi.org/10.1109/ICASSP.2017.7952261.

Tiantian Geng, Teng Wang, Jinming Duan, Runmin Cong, and Feng Zheng. Dense-localizing audio-visual events in untrimmed videos: A large-scale benchmark and baseline. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22942–22951. IEEE, 2023.

Sreyan Ghosh, Ashish Seth, Sonal Kumar, Utkarsh Tyagi, Chandra Kiran Reddy Evuru, Ramaneswaran S, S Sakshi, Oriol Nieto, Ramani Duraiswami, and Dinesh Manocha. Compa: Addressing the gap in compositional reasoning in audio-language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=86NGO8qeWs.

Sreyan Ghosh, Arushi Goel, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, et al. Audio flamingo 3: Advancing audio intelligence with fully open large audio language models. *Advances in Neural Information Processing Systems*, 38:41819–41886, 2025a.

Sreyan Ghosh, Zhifeng Kong, Sonal Kumar, S Sakshi, Jaehyeon Kim, Wei Ping, Rafael Valle, Dinesh Manocha, and Bryan Catanzaro. Audio flamingo 2: An audio-language model with long-audio understanding and expert reasoning abilities. In *International Conference on Machine Learning*, pp. 19358–19405. PMLR, 2025b.

Sreyan Ghosh, Arushi Goel, Kaousheik Jayakumar, Lasha Koroshinadze, Nishit Anand, Zhifeng Kong, Siddharth Gururani, Sang-gil Lee, Jaehyeon Kim, Aya Aljafari, et al. Audio flamingo next: Next-generation open audio-language models for speech, sound, and music. *arXiv preprint arXiv:2604.10905*, 2026.

Google DeepMind. Gemini 3.1 pro: Best for complex tasks and bringing creative concepts to life. Technical report, Google, February 2026a. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Google DeepMind. Gemini 3.6 flash model card. Technical report, Google, July 2026b. URL https://deepmind.google/models/model-cards/gemini-3-6-flash/.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18995–19012, 2022.

Peize He, Zichen Wen, Yubo Wang, Yuxuan Wang, Xiaoqian Liu, Jiajie Huang, Zehui Lei, Zhuangcheng Gu, Xiangqi Jin, Jiabing Yang, et al. Audiomarathon: A comprehensive benchmark for long-context audio understanding and efficiency in audio llms. *arXiv preprint arXiv:2510.07293*, 2025.

Shawn Hershey, Daniel P W Ellis, Eduardo Fonseca, Aren Jansen, Caroline Liu, R Channing Moore, and Manoj Plakal. The benefit of temporally-strong labels in audio event classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 366–370, 2021. doi: 10.1109/ICASSP39728.2021.9414579.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities. In *International Conference on Machine Learning*, pp. 25125–25148. PMLR, 2024.

Chun-Yi Kuan, Wei-Ping Huang, and Hung-Yi Lee. Understanding sounds, missing the questions: The challenge of object hallucination in large audio-language models. In *Interspeech 2024*, pp. 4144–4148, ISCA, September 2024. ISCA.

Sonal Kumar, Prem Seetharaman, Ke Chen, Oriol Nieto, Jiaqi Su, Zhepei Wang, Rithesh Kumar, Dinesh Manocha, Nicholas J Bryan, Zeyu Jin, et al. Tac: Timestamped audio captioning. *arXiv preprint arXiv:2602.15766*, 2026.

Aleksandr Kutsakov, Mariia Sadovina, Georgii Gospodinov, Alexandr Maximenko, Oleg Kutuzov, Pavel Bogomolov, and Fyodor Minkin. Gigachat audio: Time-aware large audio language model. *arXiv preprint arXiv:2607.10387*, 2026.

Ziyang Ma, Yinghao Ma, Yanqiao Zhu, Chen Yang, Yi-Wen Chao, Ruiyang Xu, Wenxi Chen, Yuanzhe Chen, Zhuo Chen, Jian Cong, et al. Mmar: A challenging benchmark for deep reasoning in speech, audio, music, and their mix. *Advances in Neural Information Processing Systems*, 38, 2025.

Irene Martín-Morató and Annamaria Mesaros. Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:902–914, 2023.

Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162, 2016.

Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. Dcase 2017 challenge setup: Tasks, datasets and baseline system. In *DCASE 2017-workshop on detection and classification of acoustic scenes and events*, 2017.

Annamaria Mesaros, Romain Serizel, Toni Heittola, Tuomas Virtanen, and Mark D Plumbley. A decade of dcase: Achievements, practices, evaluations and future challenges. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Hokuto Munakata, Taichi Nishimura, Shota Nakada, and Tatsuya Komatsu. Language-based audio moment retrieval. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Hokuto Munakata, Takehiro Imamura, Taichi Nishimura, and Tatsuya Komatsu. Castella: Long audio dataset with captions and temporal boundaries. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 15352–15356. IEEE, 2026.

OpenAI et al. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018. ACM Press, 2015. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806390. URL http://dl.acm.org/citation.cfm?doid=2733373.2806390.

Paul Primus, Florian Schmid, and Gerhard Widmer. Tacos: Temporally-aligned audio captions for language-audio pretraining. In *2025 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5. IEEE, 2025.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Sakshi Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. Mmau: A massive multi-task audio understanding and reasoning benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 84929–84964, 2025.

Mingchen Shao, Hang Su, Wenjie Tian, Bingshen Mu, Zhennan Lin, Lichun Fan, Zhenbo Luo, Jian Luan, and Lei Xie. Listening with time: Precise temporal awareness for long-form audio understanding. *arXiv preprint arXiv:2604.22245*, 2026.

Nian Shao, Xian Li, and Xiaofei Li. Fine-tune the pretrained atst model for sound event detection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 911–915, 2024a. doi: 10.1109/ICASSP48485.2024.10446159.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Yanfeng Shi, Pengfei Cai, Jun Liu, Qing Gu, Nan Jiang, Lirong Dai, Ian McLoughlin, and Yan Song. Towards fine-grained temporal perception: Post-training large audio-language models with audio-side time prompt. *arXiv preprint arXiv:2604.13715*, 2026.

Luoyi Sun, Xiao Zhou, Zeqian Li, Ya Zhang, Yanfeng Wang, and Weidi Xie. Spotsound: Enhancing large audio-language models with fine-grained temporal grounding. *arXiv preprint arXiv:2604.13023*, 2026.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 16607–16629, 2024.

Dingdong Wang, Junan Li, Jincenzi Wu, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. Mmsu: A massive multi-task spoken language understanding and reasoning benchmark. In *International Conference on Learning Representations*, volume 2026, pp. 31374–31410, 2026a.

Hualei Wang, Yiming Li, Shuo Ma, Hong Liu, and Xiangdong Wang. Listening between the frames: Bridging temporal gaps in large audio-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26233–26241, 2026b.

Yusong Wu, Christos Tsirigotis, Ke Chen, Cheng-Zhi Anna Huang, Aaron Courville, Oriol Nieto, Prem Seetharaman, and Justin Salamon. FLAM: Frame-wise language-audio modeling. In *Forty-second International Conference on Machine Learning (ICML)*, 2025. URL `https://openreview.net/forum?id=7fQohcFrxG`.

Zeyu Xie, Xuenan Xu, Zhizheng Wu, and Mengyue Wu. Audiotime: A temporally-aligned audio-text benchmark dataset. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10889879.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025a. URL https://arxiv.org/abs/2503.20215.

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 preprint arXiv:2509.17765*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jashin Ye, Dongxiao Wang, Yixuan Ye, Sashuai Zhou, Weihuang Lin, Mingyang Han, Kunpeng Wang, Zeyu Yuan, Boyu Li, Haoxiang Shi, et al. Voicegiraffe: A benchmark for extreme long-context audio-language understanding. *arXiv preprint arXiv:2605.27976*, 2026.

Zelin Zhou, Zhiling Zhang, Xuenan Xu, Zeyu Xie, Mengyue Wu, and Kenny Q Zhu. Can audio captions be evaluated with image caption metrics? In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 981–985. IEEE, 2022.

Appendix

Table of Contents

- Limitations And Future Work ..... A
- EgoSCAPE Benchmark Details ..... B
  - Comparison with existing benchmarks B.1
  - Benchmark statistics B.2
  - Human verification details B.3
- Experimental Details C
  - Training details C.1
  - Evaluation metrics C.2
  - Baseline implementation details C.3
  - Figure 3 details C.4
- Extended Results ..... D
  - Audio captioning D.1
  - Temporal audio grounding D.2
  - General audio benchmarks D.3
  - Hyper-parameter and reward design D.4
  - Robustness to hallucination D.5
  - Attention behavior D.6
  - Importance of real data D.7
- Prompts ..... E

A Limitations And Future Work

Our work represents a step forward in improving the temporal grounding and understanding of audio models. Although the EgoSCAPE benchmark evaluates how well current models ground events in long audio, there remains a need for benchmarks that probe deeper temporal reasoning, where models must use grounding information to solve multi-hop questions over long recordings. Furthermore, while we have so far evaluated the ATTEST framework on improving grounding in audio models, the same techniques could be extended to vision and omni-modal models to strengthen their temporal capabilities.

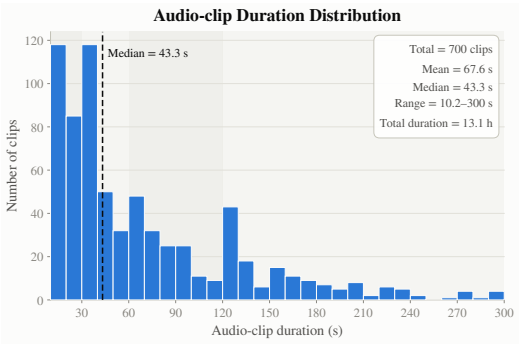
B EgoSCAPE Benchmark Details

B.1 Comparison with existing benchmarks

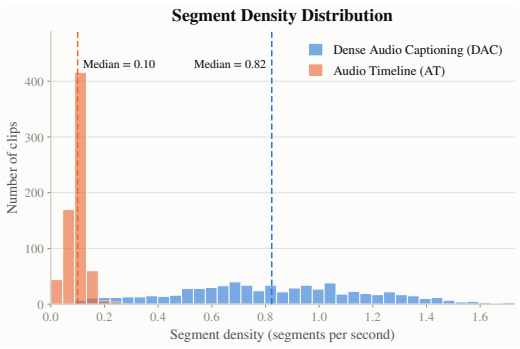
Table 5: Comparison of EgoSCAPE with existing temporal audio grounding benchmarks.

Dataset	Type	# unique audios	# samples	Avg. Dur.	Max. Dur.	Dense Captions	Source Annot.	Audio Timeline	Temporal Grounding
FTAR-test	Real	1224	1738	9.9	10.0	✓	✗	✗	✓
LAT-Bench	Real	247	2722	600.0	1790.2	✗	✗	✓	✓
Clotho-Moment	Synthetic	7569	7569	60.0	60.0	✗	✗	✗	✓
SpotSound-Bench	Synthetic	300	300	52.9	60.0	✗	✗	✗	✓
CASTELLA	Real	640	1302	206.0	300.0	✗	✗	✗	✓
EgoSCAPE	**Real**	**700**	**4200**	**67.6**	**300.0**	✓	✓	✓	✓

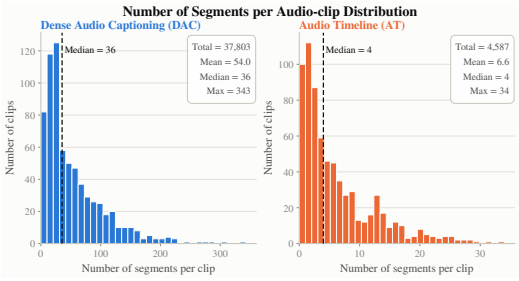
Tab. 5 shows the comparison of our benchmark with existing benchmarks which are not rich in annotations especially for dense annotations over long audio clips (Wang et al., 2026b; Shao et al., 2026; Sun et al., 2026; Munakata et al., 2025; 2026).



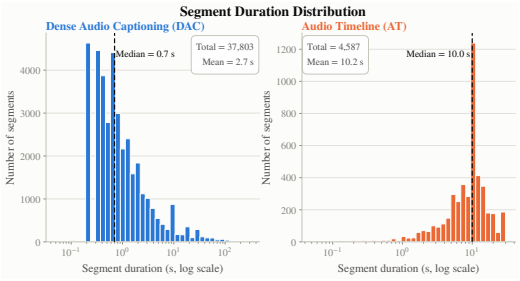
(a) Audio-clip duration distribution.



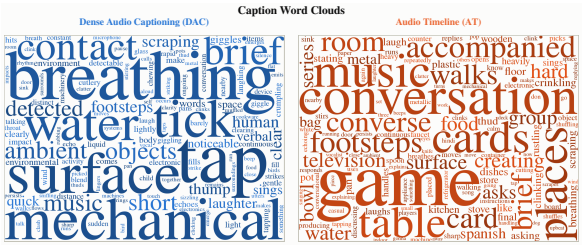
(b) Segment density distribution.



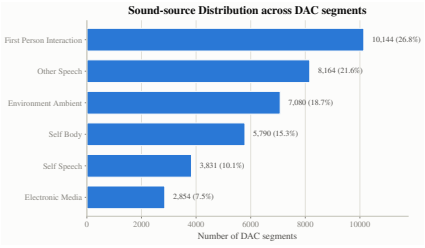
(c) Number of segments per audio-clip.



(d) Segment duration distribution.



(e) Word-clouds for DAC and AT tasks.



(f) DAC sound source distribution.

Figure 5: Statistics for the EgoSCAPE benchmark tasks.

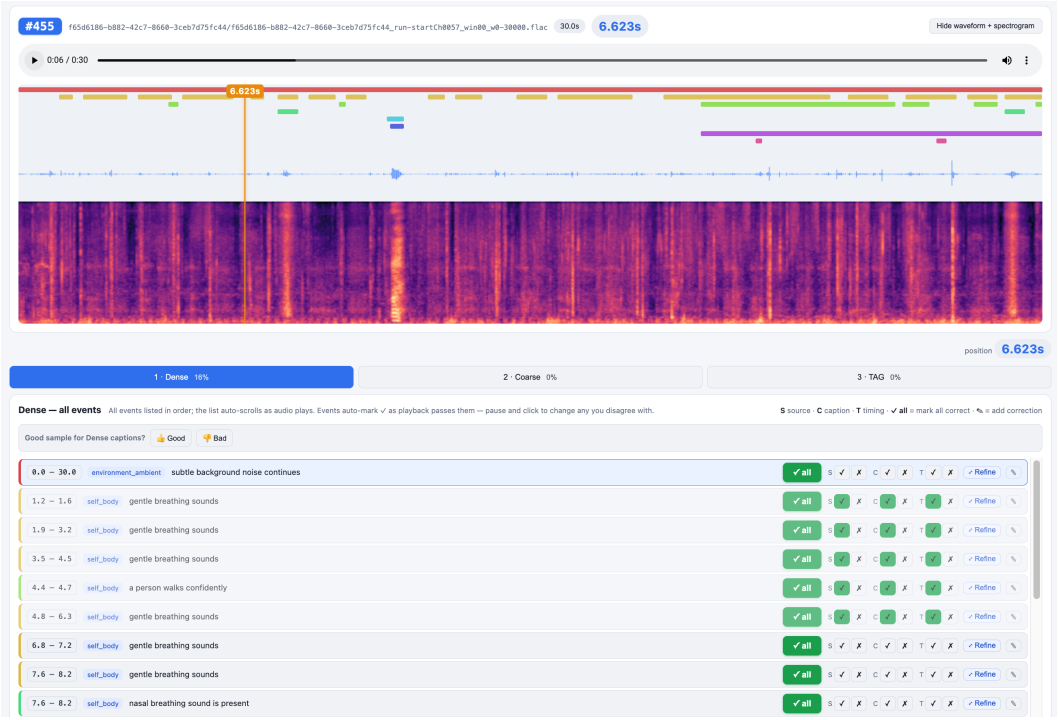


Figure 6: Dashboard used for performing human verification.

Table 6: Inter-annotator agreement for the human verification of EgoSCAPE benchmark. The agreement is computed over the fine-grained entities for each task: each time-stamped segment in dense captioning and audio timeline generation, and each query in temporal audio grounding.

Task	# audios	# entities	# annotators	Raw Agree.	Fleiss' κ
Dense Audio Captioning	700	37803	3	90.7%	0.789
Audio Timeline Generation	700	4587	3	88.6%	0.764
Temporal Audio Grounding	700	2800	3	91.0%	0.797

B.2 Benchmark Statistics

Fig. 5 shows the detailed stats for the EgoSCAPE benchmark. Our benchmark consists of 700 audio clips totaling 13.15 hours, with individual clips spanning durations of up to 5 minutes with a mean duration of around 67 seconds. The dense audio captions consist of a total of 37,803 segments with annotations about the detailed audio event present on it, with around 54 segments annotated per audio on average. Dense caption segments are about 2.7 seconds long on average and have a density of about 0.82 segments per second. On the other hand, the audio timeline consists of coarser time segments with a total of 4,587 segments and about 6.6 segments annotated per audio clip. These segments are 10 seconds long on average. Fig. 5e shows the word clouds for the dense captioning and audio timeline annotations. Finally, the distribution of the sound source annotations present in the dense captions is available in Fig. 5f.

B.3 Human Verification Details

Fig. 6 shows the dashboard used for human verification. For each audio clip, annotators reviewed the annotations generated for the different tasks and marked whether the caption, the timing, and the associated sound sources of each audio segment were correct. If an annotator judged an annotation to be incorrect, they could also suggest a refined version. Annotators could play the audio files but had no access to the corresponding videos. To

ensure accurate temporal annotations, the interface displayed both the audio waveform and its spectrogram.

Tab. 6 reports the inter-annotator agreement for the human verification of the EgoSCAPE benchmark. Agreement is computed over the entities in each task, where an entity is a single time-stamped caption for dense captioning and timeline generation, and a single query for temporal grounding. An entity is considered incorrectly annotated by the automatic pipeline if an annotator marks its caption/query, timing, or sound source as wrong. We compute Fleiss' kappa (κ) (Fleiss, 1971) over the annotators' judgments of whether each entity's annotations are correct.

C Experimental Details

C.1 Training details

We adapt GRPO with the rewards and techniques described in §4. We train only the linear modules of the base model using LoRA (Hu et al., 2022) with rank $r = 8$ and $\alpha = 32$. Prior to RL, we perform a cold-start SFT stage with a learning rate of 5×10^{-5} on dense captioning, audio timeline, and temporal audio grounding data constructed from a separate set of Ego4D audio clips using the pipeline described in §3. Additionally, following TimeAudio (Wang et al., 2026b), we convert AudioSet-SL (Gemmeke et al., 2017; Hershey et al., 2021) into dense captioning and temporal audio grounding formats and include it in the training mix. In total, we use 222k samples for DAC, 70k samples for AT, and 292k samples for temporal audio grounding. For RL fine-tuning, we use 5k samples each for the DAC and AT tasks. We set the GRPO group size to 8 and the KL regularization coefficient β to 0.001. The LoRA adapter is fine-tuned with GRPO using a learning rate of 5×10^{-6} and an effective batch size of 16. All experiments are conducted on 64 NVIDIA A100 (80GB) GPUs, and training is performed using the DeepSpeed[1] ZeRO-3 configuration.

To prevent the model from hacking R_{AD} by predicting short segments, we cap ρ at a maximum value of 5. Moreover, to encourage precise predictions, the format reward also penalizes outputs whose number of predicted segments differs from the number of ground-truth segments. In §4, a segment's A^{DAC} term is 0 if it is unmatched ($\sigma^{(g)}(j) = 0$), and its A^{CAG} term is 0 if it lies outside all silenced windows. Finally, α is set to 0.5 in Eq. (4).

C.2 Evaluation metrics

C.2.1 Captioning metrics

FENSE@τ. We report the FENSE scores (Zhou et al., 2022) at the thresholds of 0.3, 0.5 and 0.7 similar to (Shao et al., 2026) for the captioning tasks (DAC and AT). As defined in §4, the ground-truth has N_g triplets $D = \{(s_i, e_i, c_i)\}_{i=1}^{N_g}$ where (s_i, e_i) are the start/end timestamps of an acoustic event described by caption c_i. Similarly, the model prediction is represented as $\hat{D} = \{(\hat{s}_j, \hat{e}_j, \hat{c}_j)\}_{j=1}^{N_p}$. For each GT segment i, we identify the predicted segment j^* that has the highest IoU above τ with the GT segment, i.e.

$$FENSE_i@\tau = \begin{cases} FENSE(c_i, \hat{c}_{j^*}) & \text{if tIoU}(d_i, \hat{d}_{j^*}) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $j^* = \arg\max_j \text{tIoU}(d_i, \hat{d}_j)$, and d_i & $\hat{d}_j$ represent the triplets (s_i, e_i, c_i) & $(\hat{s}_j, \hat{e}_j, \hat{c}_j)$ respectively. The final metric is the average over all the GT triplets, i.e.

$$FENSE@\tau = \frac{1}{N_g} \sum_{i=1}^{N_g} FENSE_i@\tau \tag{13}$$

SED F1 metrics. For the DAC task we additionally report two sound-event-detection style F1 scores adapted from Mesaros et al. (2016): a frame-based **Seg-F1** and an event-based **Evt-F1** with an onset collar of 0.5 s. Since our captions are free-form rather than drawn from a fixed class vocabulary, we induce per-clip event classes by clustering the captions of D with single-linkage connected components over the SBERT (paraphrase-TinyBERT-L6-v2) (Reimers & Gurevych, 2019) cosine-similarity graph thresholded at $\tau_{\text{sim}} = 0.65$. Let $\kappa(\cdot)$ denote the resulting class assignment, so that two captions are of the same class iff they land in the same cluster. For both metrics we report the micro-averaged F1 obtained by pooling the counts across all clips of the test set.

Hallucination Rate. Finally we report an additional hallucination rate similar to TAC (Kumar et al., 2026) for the DAC task. We report the hallucination rate (Hal%) as the

[1] `https://www.deepspeed.ai/`

percentage of predicted triplets or segments that are not matched to any of the GT segments at tIoU of 0.5 and have a *confidence* of less than 0.15. *Confidence* is defined as the maximum audio-text FLAM (Wu et al., 2025) similarity score across the time-region of a predicted segment/triplet.

C.2.2 Temporal Audio Grounding Metrics

Recall@τ. We compute the tIoU between the predicted time-ranges and the ground-truth time ranges and report the recall at the thresholds of 0.5 and 0.7 tIoU.

mIoU. We report the mean tIoU scores across all the samples of the dataset as mIoU.

C.3 Baseline implementation details

C.3.1 LLM based models

All models are asked to emit timestamps in seconds at a 100 ms (one-decimal) precision. For captioning tasks the instruction requests a flat list of `start - end, description` segments and for temporal audio grounding (TAG) it requests only the `start - end` ranges where the query occurs. The EgoSCAPE dense task additionally requests a source tag drawn from a taxonomy (Fig. 5f) prepended to each segment. Below we describe only what is *specific* to each baseline.

Proprietary models (Gemini). The only model-specific post-processing is timestamp normalization: Gemini tends to emit clock-style (`m:ss`) timestamps, which we convert to decimal seconds before scoring so that the shared parser within our evaluation pipeline reads them correctly.

Open audio and omni LLMs. These models split into two groups by how they handle long audio. *Native long-audio* models are run with a single forward pass and rely on their own internal sliding-window audio encoders: AudioFlamingo 2 (up to 5 min), AudioFlamingo 3 (up to ∼600 s), and AF-Next (∼1800 s). *Chunk-and-merge* models are those whose audio front-end caps at 30 s (Qwen2-Audio and Kimi-Audio). For these we split the clip into non-overlapping 30 s windows, run inference on each window independently, offset every predicted timestamp by its window start, and merge the per-window predictions into a single clip-level prediction (unioning overlapping ranges for TAG and concatenating segments for DAC) before scoring, so the unchanged scorer sees one prediction per clip.

Captioning models. The captioner baselines (AF-Next Captioner and Qwen3-Omni Captioner) are run in two stages because they are trained to produce rich free-form captions rather than a rigid task format. First, the audio model is prompted to produce a timestamped free-form caption of the clip. Second, a text-only LLM (Qwen3-30B-Instruct) reformats that caption into the exact DAC/timeline/TAG format, using the corresponding task instruction as the formatting specification. Because Qwen3 Omni Captioner captions best on short inputs, it is also chunked into 30 s windows and the reformatting stage is applied per window with a deterministic offset added to each window's timestamps before concatenation, so all timeline arithmetic is exact.

Specialized temporal models. For longer clips processed through TimeAudio, we apply the same windowed mechanism as above with 10 s non-overlapping windows; each window's timestamps are shifted by its start and stitched into one prediction. TimeAudio results obtained this way are marked with $*$ in the main tables. LAT-Audio is run with its tool-augmented reasoning loop, in which the model iteratively issues audio-crop calls to zoom into sub-segments before answering, and thus handles long audio natively without fixed chunking. SpotSound-A is used off-the-shelf as it is an AudioFlamingo-3-based grounding model with timestamp-interleaved audio tokens that reads the query and emits a single grounded interval. AM-DETR is the baseline from Munakata et al. (2025) and UVCOM is the baseline from Munakata et al. (2026).

C.3.2 RL-based baselines

We also adapt TimePro-RL (Shi et al., 2026) for our setup by using identical training data and models. We include temporal markers every 0.04 seconds similar to their approach causing the context usage to increase by 7500 tokens for a 300 seconds audio. Additionally, as suggested by TimePro-RL, we include the main reward to be the task metric and auxiliary reward as R_{DAC} as we have defined.

C.4 Figure 3 details

For Fig. 3, we plot the hallucination rate for the SFT trained variants of Qwen3 Omni and Qwen2.5 Omni, over different durations of audios from EgoSCAPE benchmark. We can see in the figure that the hallucination rate clearly rises with the duration of the audio for both the SFT trained models.

D Extended Results

D.1 Audio captioning

Table 7: Dense audio captioning results (expanded from Tab. 1) on EgoSCAPE and AudioSet-SL datasets comparing ATTEST to other audio and omni LLM baselines. EvtF1@0.5 and SegF1 are micro. Hal% and SrcF1@0.5 are macro.

Model	EgoSCAPE Dense Audio Captioning							AudioSet-SL					
	F@0.3	F@0.5	F@0.7	EvtF1	SegF1	Hal%↓	SrcF1	F@0.3	F@0.5	F@0.7	EvtF1	SegF1	Hal%↓
Proprietary Models													
Gemini 3.1 Pro	0.235	0.213	0.201	0.084	0.276	31.1	0.371	0.207	0.192	0.181	0.079	0.126	22.7
Gemini 3.5 Flash	0.226	0.208	0.198	0.066	0.180	38.5	0.327	0.208	0.192	0.176	0.043	0.074	23.7
Gemini 3.6 Flash	0.242	0.215	0.200	0.091	0.197	38.9	0.351	0.217	0.200	0.172	0.037	0.066	30.3
Gemini 3.1 Flash Lite	0.228	0.211	0.199	0.064	0.149	34.7	0.400	0.202	0.185	0.171	0.032	0.064	25.1
GPT RealTime 2.1	0.207	0.199	0.195	0.026	0.113	42.2	0.387	0.193	0.176	0.165	0.026	0.040	54.5
GPT RealTime 2.0	0.209	0.200	0.196	0.025	0.114	45.6	0.387	0.208	0.186	0.170	0.038	0.045	52.9
GPT 4o Audio	0.226	0.208	0.198	0.059	0.186	48.4	0.359	0.208	0.184	0.167	0.051	0.058	50.9
Captioning Models													
AF-Next Captioner	0.244	0.218	0.201	0.099	0.147	52.7	0.325	0.239	0.207	0.184	0.058	0.087	44.6
Qwen3 Omni Captioner	0.270	0.232	0.206	0.257	0.335	64.4	0.369	0.257	0.220	0.190	0.048	0.057	63.6
Audio and Omni LLMs													
Qwen2 Audio*	0.207	0.199	0.196	0.048	0.127	51.0	0.232	0.207	0.191	0.182	0.054	0.076	43.1
Kimi-Audio*	0.188	0.183	0.177	0.042	0.034	42.9	0.207	0.224	0.198	0.175	0.061	0.107	33.9
TimeAudio*	0.340	0.310	0.271	0.256	0.344	35.2	-	0.426	0.389	0.352	0.353	0.372	36.3
AudioFlamingo 2	0.193	0.193	0.192	0.004	0.001	59.7	-	0.192	0.176	0.167	0.079	0.084	55.5
AF-Next	0.181	0.178	0.176	0.022	0.014	45.9	0.030	0.226	0.202	0.184	0.089	0.114	48.5
Qwen2.5 Omni	0.178	0.177	0.176	0.021	0.035	48.2	0.165	0.220	0.202	0.192	0.124	0.190	20.5
+ SFT	0.412	0.367	0.313	0.320	0.303	47.2	0.564	0.448	0.403	0.362	0.336	0.337	25.8
+ TimePro-RL	0.445	0.410	0.358	0.349	0.348	31.3	0.642	0.475	0.430	0.385	0.347	0.370	20.1
+ ATTEST (Ours)	**0.461**	**0.427**	**0.372**	**0.371**	**0.355**	**30.5**	**0.656**	**0.489**	**0.442**	**0.399**	**0.354**	0.389	**18.2**
Qwen3 Omni	0.174	0.176	0.176	0.015	0.038	45.4	0.231	0.265	0.231	0.208	0.138	0.175	29.2
+ SFT	0.419	0.366	0.318	0.301	0.325	47.8	0.547	0.438	0.396	0.368	0.371	0.340	34.5
+ TimePro-RL	0.451	0.415	0.359	0.382	0.337	32.5	0.658	0.484	0.438	0.389	0.399	0.411	28.0
+ ATTEST (Ours)	**0.463**	**0.428**	**0.373**	**0.400**	**0.353**	**30.7**	**0.665**	**0.498**	**0.450**	**0.404**	**0.410**	**0.425**	**26.9**
AudioFlamingo 3	0.196	0.194	0.193	0.042	0.010	40.3	-	0.204	0.189	0.177	0.119	0.131	32.4
+ SFT	0.376	0.344	0.298	0.258	0.248	44.4	0.587	0.453	0.406	0.362	0.345	0.396	36.3
+ TimePro-RL	0.421	0.388	0.328	0.335	0.329	34.0	0.620	0.489	0.440	0.384	0.369	0.385	29.2
+ ATTEST (Ours)	**0.437**	**0.400**	**0.342**	**0.346**	**0.345**	**31.8**	**0.641**	**0.503**	**0.451**	**0.398**	**0.380**	**0.400**	**25.3**

Dense audio captioning. Tab. 7 is an expanded version of the DAC results present in §5.1 including ATTEST framework results on Qwen3 Omni. Additionally, we report results on more proprietary baseline models here including Gemini 3.5 Flash, Gemini 3.1 Flash Lite, GPT RealTime 2.1 and GPT RealTime 2.0. These additional results further strengthen the effectiveness of the proposed approach for dense captioning.

Audio timeline generation. Tab. 8 similarly reports the performance comparison between the baselines and proposed approach for the audio timeline generation task of EgoSCAPE benchmark. The ATTEST framework significantly improves the performance of the base models for this task as well outperforming all the baselines. While the performance of our base models and finetuned models are better on timeline generation compared to dense captioning, the performance of some baselines such as Gemini 3.1 Pro and AudioFlamingo models is similar.

D.2 Temporal audio grounding

Zero-shot results. Tab. 9 is an expanded version of the TAG results present in §5.2 including the ATTEST framework results on Qwen3 Omni. Similar to Tab. 7, we include the results from additional proprietary models on different benchmarks.

EgoSCAPE. For the proposed EgoSCAPE benchmark, we report the temporal grounding results in Tab. 10 over coarse and fine-grained audio queries (§3.1). While the baselines are particularly poor at grounding fine-grained audio events, base models tuned with the proposed approach are equally good at handling coarse and fine-grained queries.

D.3 General audio benchmarks

Tab. 11 expands the results reported on general benchmarks in §5.4. We include results of different post-training techniques including ATTEST on Qwen3 Omni.

Table 8: Audio timeline generation results on the EgoSCAPE benchmark.

Model	F@0.3	F@0.5	F@0.7
Proprietary Models			
Gemini 3.1 Pro	0.234	0.197	0.172
Gemini 3.5 Flash	0.213	0.185	0.166
Gemini 3.6 Flash	0.218	0.183	0.164
Gemini 3.1 Flash Lite	0.245	0.205	0.174
GPT RealTime 2.1	0.209	0.184	0.167
GPT RealTime 2.0	0.217	0.190	0.166
GPT 4o Audio	0.239	0.199	0.166
Captioning Models			
AF-Next Captioner	0.241	0.199	0.169
Qwen3 Omni Captioner	0.341	0.258	0.185
Audio and Omni LLMs			
Qwen2 Audio	0.153	0.150	0.149
Kimi-Audio	0.186	0.170	0.161
AudioFlamingo 2	0.164	0.156	0.154
AF-Next	0.183	0.176	0.160
LAT-Audio	0.220	0.184	0.168
Qwen2.5 Omni	0.211	0.189	0.170
+ SFT	0.466	0.401	0.316
+ TimePro-RL	0.489	0.427	0.329
+ ATTEST (Ours)	**0.503**	**0.436**	**0.340**
Qwen3 Omni	0.304	0.255	0.208
+ SFT	0.469	0.403	0.313
+ TimePro-RL	0.495	0.429	0.332
+ ATTEST (Ours)	**0.509**	**0.442**	**0.345**
AudioFlamingo 3	0.187	0.168	0.159
+ SFT	0.458	0.396	0.312
+ TimePro-RL	0.489	0.425	0.329
+ ATTEST (Ours)	**0.505**	**0.438**	**0.341**

Table 9: Zero-shot temporal audio grounding results on Clotho-Moment, UnAV-100 and CASTELLA datasets comparing ATTEST to other audio and omni LLM baselines.

Model	Clotho-Moment			UnAV-100			CASTELLA		
	R@0.5	R@0.7	mIoU	R@0.5	R@0.7	mIoU	R@0.5	R@0.7	mIoU
Proprietary Models									
Gemini 3.1 Pro	0.385	0.296	0.344	0.340	0.240	0.294	0.080	0.041	0.081
Gemini 3.5 Flash	0.189	0.079	0.223	0.430	0.220	0.373	0.082	0.039	0.098
Gemini 3.6 Flash	0.581	0.431	0.536	0.570	0.410	0.524	0.233	0.124	0.223
Gemini 3.1 Flash Lite	0.235	0.134	0.302	0.540	0.340	0.487	0.042	0.021	0.050
GPT RealTime 2.1	0.021	0.007	0.056	0.090	0.030	0.135	0.029	0.013	0.049
GPT RealTime 2.0	0.042	0.011	0.085	0.100	0.070	0.166	0.036	0.012	0.053
GPT 4o Audio	0.106	0.031	0.185	0.200	0.060	0.216	0.042	0.018	0.062
Finetuned Temporal Grounding Models									
AM-DETR	0.882	0.827	0.809	0.580	0.410	0.428	-	-	-
SpotSound-A	0.908	0.853	0.868	0.780	0.620	0.725	0.180	0.115	0.181
UVCOM	0.885	0.832	0.820	0.590	0.360	0.518	0.317	0.203	-
Audio and Omni LLMs									
Qwen2 Audio*	0.043	0.005	0.131	0.080	0.040	0.181	0.053	0.020	0.097
Kimi-Audio*	0.088	0.025	0.164	0.180	0.030	0.265	0.080	0.022	0.146
AudioFlamingo 2	0.009	0.003	0.023	0.040	0.010	0.056	0.011	0.005	0.016
AF-Next	0.004	0.002	0.019	0.030	0.030	0.069	0.008	0.002	0.013
Qwen2.5 Omni	0.409	0.141	0.426	0.470	0.280	0.458	0.092	0.048	0.116
+ SFT	0.875	0.821	0.830	0.790	0.600	0.693	0.224	0.123	0.269
+ TimePro-RL	0.895	0.844	0.859	0.800	0.620	0.711	0.252	0.143	0.295
+ ATTEST (Ours)	**0.909**	**0.859**	**0.876**	**0.820**	**0.650**	**0.729**	**0.297**	**0.186**	**0.312**
Qwen3 Omni	0.495	0.424	0.528	0.760	0.590	0.681	0.190	0.109	0.187
+ SFT	0.872	0.828	0.846	0.790	0.610	0.717	0.207	0.136	0.229
+ TimePro-RL	0.890	0.834	0.860	0.810	0.620	0.731	0.219	0.139	0.245
+ ATTEST (Ours)	**0.904**	**0.841**	**0.871**	**0.830**	**0.640**	**0.745**	**0.264**	**0.155**	**0.298**
AudioFlamingo 3	0.142	0.063	0.163	0.120	0.030	0.147	0.008	0.004	0.013
+ SFT	0.850	0.754	0.801	0.780	0.620	0.708	0.147	0.095	0.170
+ TimePro-RL	0.892	0.799	0.836	0.810	0.650	0.744	0.168	0.110	0.189
+ ATTEST (Ours)	**0.903**	**0.819**	**0.854**	**0.850**	**0.660**	**0.753**	**0.202**	**0.123**	**0.266**

Table 10: Temporal audio grounding results on EgoSCAPE benchmark across coarse and fine-grained audio-related queries.

Model	Coarse			Fine		
	R@0.5	R@0.7	mIoU	R@0.5	R@0.7	mIoU
Proprietary Models						
Gemini 3.1 Pro	0.173	0.110	0.191	0.060	0.042	0.077
Gemini 3.5 Flash	0.159	0.077	0.192	0.048	0.028	0.065
Gemini 3.6 Flash	0.197	0.106	0.236	0.077	0.046	0.094
Gemini 3.1 Flash Lite	0.184	0.117	0.217	0.074	0.048	0.090
GPT RealTime 2.1	0.072	0.033	0.112	0.023	0.007	0.033
GPT RealTime 2.0	0.103	0.045	0.131	0.032	0.017	0.045
GPT 4o Audio	0.077	0.032	0.135	0.036	0.011	0.056
Audio and Omni LLMs						
Qwen2 Audio*	0.092	0.045	0.126	0.027	0.011	0.039
Kimi-Audio*	0.123	0.049	0.168	0.072	0.019	0.098
SpotSound-A	0.279	0.155	0.283	0.172	0.106	0.183
AudioFlamingo 2	0.053	0.011	0.069	0.025	0.010	0.041
AF-Next	0.028	0.015	0.044	0.025	0.013	0.034
Qwen2.5 Omni	0.202	0.105	0.215	0.069	0.043	0.084
+ SFT	0.444	0.307	0.436	0.506	0.366	0.467
+ TimePro-RL	0.492	0.339	0.489	0.510	0.375	0.474
+ ATTEST (Ours)	**0.518**	**0.358**	**0.505**	**0.529**	**0.393**	**0.481**
Qwen3 Omni	0.243	0.169	0.280	0.119	0.070	0.132
+ SFT	0.443	0.302	0.431	0.458	0.346	0.426
+ TimePro-RL	0.521	0.326	0.505	0.498	0.361	0.456
+ ATTEST (Ours)	**0.539**	**0.343**	**0.521**	**0.514**	**0.378**	**0.479**
AudioFlamingo 3	0.151	0.071	0.167	0.073	0.033	0.091
+ SFT	0.393	0.279	0.403	0.398	0.285	0.367
+ TimePro-RL	0.538	0.355	0.539	0.460	0.328	0.428
+ ATTEST (Ours)	**0.574**	**0.376**	**0.556**	**0.475**	**0.334**	**0.447**

Table 11: Results of the proposed approach on LongAudioBench and MMAR.

Model	MMAR		LongAudioBench	
	All	Temporal	All	Temporal
Qwen2.5 Omni	63.2	56.0	61.5	60.1
+ TimePro-RL	61.0	55.4	58.3	59.5
+ ATTEST (Ours)	**64.4**	**58.2**	**62.8**	**62.2**
+ ATTEST w/ markers	62.2	56.9	60.5	61.2
Qwen3 Omni	73.4	67.1	65.9	64.8
+ TimePro-RL	71.0	65.8	62.3	62.7
+ ATTEST (Ours)	**74.1**	**69.2**	**66.5**	**65.9**
+ ATTEST w/ markers	72.8	66.5	64.1	64.0
AudioFlamingo 3	60.4	53.2	57.7	53.9
+ TimePro-RL	58.3	51.0	55.2	52.5
+ ATTEST (Ours)	**62.1**	**57.9**	**59.6**	**57.2**
+ ATTEST w/ markers	59.2	52.8	56.9	53.5

D.4 Hyper-parameter and reward design

This subsection extends the ablations and hyper-parameter analysis from §5.3.

Reward weight tuning. Tab. 12 reports the effect of varying the advantage weights η_1, η_2, and η_3 one at a time around our default setting $(\eta_1, \eta_2, \eta_3) = (1, 0.2, 1)$. Performance on EgoSCAPE dense captioning and Clotho-Moment is stable across the sweep: F@0.5 stays within 0.410–0.427 and Clotho-Moment R@0.5 within 0.891–0.909. CASTELLA is more sensitive, with R@0.5 ranging from 0.250 to 0.297. Every configuration in the sweep still outperforms vanilla GRPO (Tab. 3), so ATTEST does not depend on precise tuning of these weights. The trends also match the role of each reward. Lowering η_1 (R_{CAG}) increases hallucinations the most (35.5% at $\eta_1 = 0.25$). Raising η_2 (R_{AD}) to 0.5 gives the lowest hallucination rate (29.5%) but reduces F@0.5, indicating a trade-off between hallucination rate and caption quality. Reducing η_3 (R_{DAC}) lowers both captioning and grounding performance.

Counterfactual Audio Grounding Reward. For R_{CAG}, we report different variations in the corruption operator $M(\cdot)$ as well as the fraction of the audio clip where this corruption is applied to in Tab. 13. In addition to silence, we tried other corruption operators such

Table 12: Effect of different reward weights on the performance of ATTEST framework.

$\eta_1(R_{\text{CAG}})$	$\eta_2(R_{\text{AD}})$	$\eta_3(R_{\text{DAC}})$	EgoSCAPE Dense Captioning				Temporal Grounding	
			Overall		Long (>120s)		Clotho-Moment	CASTELLA
			F@0.5	Hal%	F@0.5	Hal%	R@0.5	R@0.5
1	**0.2**	**1**	**0.427**	**30.5**	**0.414**	**34.2**	**0.909**	**0.297**
0.25	0.2	1	0.415	35.5	0.389	37.3	0.901	0.256
0.5	0.2	1	0.424	32.3	0.406	35.1	0.905	0.285
2	0.2	1	0.418	30.8	0.401	34.5	0.906	0.267
1	0.05	1	0.414	32.2	0.395	35.3	0.902	0.261
1	0.1	1	0.425	31.2	0.410	34.6	0.903	0.279
1	0.5	1	0.410	29.5	0.389	33.8	0.891	0.250
1	0.2	0.25	0.411	31.0	0.391	35.0	0.893	0.251
1	0.2	0.5	0.425	31.6	0.399	34.9	0.901	0.281
1	0.2	2	0.426	32.6	0.410	36.8	0.909	0.291

Table 13: Effect of varying corruption operator $M(\cdot)$ and fraction p used in R_{CAG}.

Corruption Operator (M)	Corruption Fraction (p)	EgoSCAPE Dense Captioning				Temporal Grounding	
		Overall		Long (>120s)		Clotho-Moment	CASTELLA
		F@0.5	Hal%	F@0.5	Hal%	R@0.5	R@0.5
silence	**0.3**	**0.427**	**30.5**	**0.414**	**34.2**	**0.909**	**0.297**
pink noise	0.3	0.419	31.2	0.404	35.6	0.902	0.265
time shuffle	0.3	0.421	32.1	0.402	36.0	0.895	0.268
audio swap	0.3	0.411	31.0	0.398	35.2	0.897	0.259
low-pass	0.3	0.409	31.4	0.399	35.1	0.894	0.255
silence	0.1	0.421	32.8	0.403	35.9	0.900	0.269
silence	0.2	0.424	31.5	0.408	35.2	0.908	0.278
silence	0.4	0.422	30.7	0.411	34.0	0.903	0.288
silence	0.5	0.409	31.9	0.400	35.3	0.898	0.271

as pink noise (RMS-matched to the original content), time-shuffle, audio swap as well as using a low-pass filter. We show that the silence operator works the best in reducing model hallucinations and improving performance, as it is the only signal that converts the audio to something that truly does not contain any acoustic events. Moreover, changing the corruption fraction to high values of 0.5 reduces the model performance significantly due to over-penalization.

Attention Density Reward. We show the effect of using different layers to drive R_{AD} in Tab. 14. We report the performance on using the first, middle, and last layers of the base Qwen2.5 Omni model. Moreover, we also tried a combination of the first, middle and last layers as well as using every 3 layers for R_{AD}. Finally, we also experimented by using the layer which has the peak difference between the attention densities of the best and worst performing samples based on FENSE scores on EgoSCAPE dense captioning task. We observe that using a single layer, specifically the last, the middle layer or the peak gap layer leads to similar performance improvements as using multiple layers. This shows that the proposed approach can be run on different choice of layers and using a single layer to keep compute minimal. We use the last layer for all our experiments wherever not specified. Also note that we use average attention over all the attention heads of that layer for all the experiments.

D.5 Robustness to hallucination

We perform additional analysis to show the robustness of ATTEST-tuned models to different types of interventions. In Fig. 7, we include silence of random duration between 20-30% of the audio duration into the EgoSCAPE dense captioning samples and re-

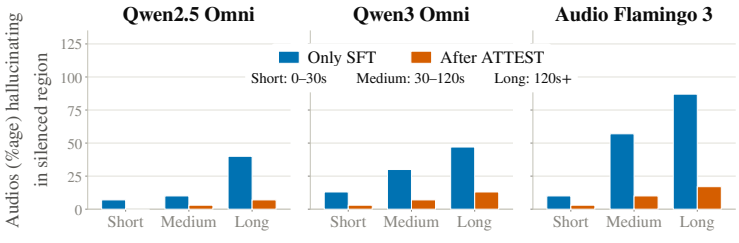


Figure 7: Percentage of audios in EgoSCAPE where at least one dense captioning segment was predicted in a silenced region.

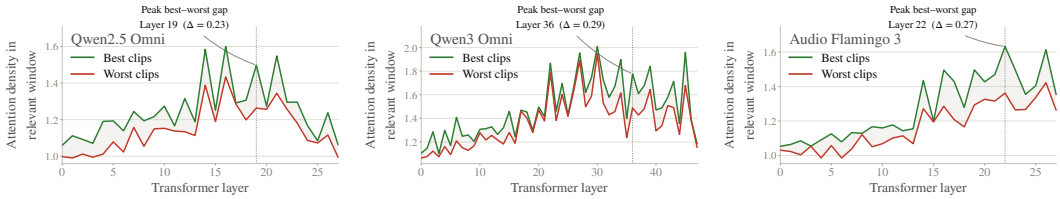


Figure 8: Attention density over the relevant audio tokens across the layers for the best performing and the worst performing samples in EgoSCAPE after performing SFT on different models.

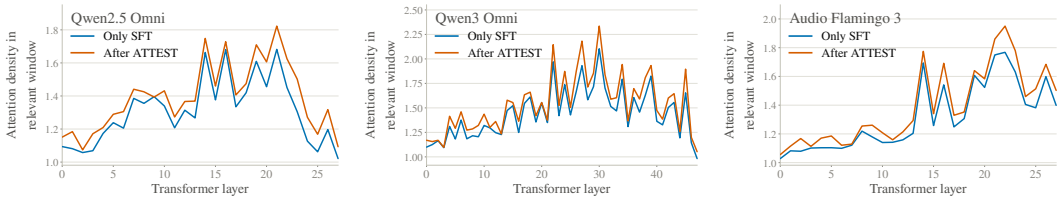


Figure 9: Attention density across the layers over the relevant audio tokens before (only SFT) and after tuning the model with the ATTEST framework averaged over all the samples of EgoSCAPE dataset.

port the fraction of audios in which at least one of the predicted segments had its center in the silenced region, indicating hallucination. We can see that only performing SFT, makes the model prone to predicting hallucinated events in the silenced regions, whereas ATTEST significantly reduces the hallucinations.

D.6 Attention behavior

Fig. 8 shows how the attention pattern differs across the best and the worst scoring samples according to FENSE scores on the EgoSCAPE dense captioning task. We ensure that the number of segments and the duration of samples in the best and worst performing samples are the same to ensure samples are of similar difficulty. We can observe that across the layers, the samples that attend to relevant audio tokens while producing sound event descriptions have better FENSE scores. This observation is the motivation to develop the attention density reward. Moreover, to show the efficacy of R_{AD} on correcting model's attention patterns, we show the attention density of different models across layers before and after applying ATTEST in Fig. 9. We can see that the attention density increases on relevant tokens across all the layers despite the supervision provided only on the last layer.

Table 14: Effect of using different layers in R_{AD}. Peak gap represents using the layer where the SFT checkpoint has the most attention density gap between the best and the worst FENSE scoring samples as mentioned in §D.6.

Layers used for R_{AD}	**EgoSCAPE Dense Captioning**				**Temporal Grounding**	
	Overall		**Long (>120s)**		**Clotho-Moment**	**CASTELLA**
	F@0.5	**Hal%**	**F@0.5**	**Hal%**	**R@0.5**	**R@0.5**
Last	**0.427**	**30.5**	**0.414**	**34.2**	**0.909**	**0.297**
First	0.419	31.6	0.408	34.9	0.905	0.289
Middle	0.426	30.4	0.409	34.5	0.905	0.281
First + Middle + Last	0.427	30.8	0.412	34.4	0.909	0.293
Every 3	0.424	30.1	0.410	33.8	0.899	0.287
Peak Gap	0.426	30.5	0.411	34.1	0.907	0.293

D.7 Importance of real data

We can create long-audios and corresponding temporal annotations for such long audios by joining short audios from a dataset (e.g. AudioSet-SL) which already has temporal annotations. To show that such a combination does not capture the real data distribution, we report the results of training the Qwen2.5 Omni model using real and/or synthetic data using the ATTEST framework. The synthetic data is constructed by joining multiple AudioSet-SL audios together randomly and creating temporal annotations for them by off-setting the original timestamps in AudioSet-SL. For reporting performance, we use a source-stripped version of EgoSCAPE which has the dense captions without the sound sources for fair comparison with AudioSet-SL. We can clearly observe that the model's performance on EgoSCAPE on using synthetic data only is significantly worse than using Real data. Moreover, we can observe that the model's performance on EgoSCAPE is better compared to synthetically joined AudioSet-SL because real data has continuous audio and continuous context makes the dense captioning task easier.

Table 15: Dense captioning results on using Real vs Synthetic long audio training data on EgoSCAPE as well as AudioSet-SL (by joining multiple clips together).

Training Data	EgoSCAPE (>120s)		AudioSet-SL (Joined-Long)	
	F@0.5	Hal%	F@0.5	Hal%
Real	**0.425**	**33.2**	0.398	36.2
Synthetic	0.361	43.8	**0.404**	**34.6**
Real + Synthetic	0.416	34.9	0.399	35.3

E Prompts

Table 16: Prompt used to verify and refine the sound events detected by the sound event detector model as described in §3.

Task. You are annotating a short audio clip (up to 30 seconds) from an egocentric recording. For each candidate sound event detected in the clip, decide keep / drop / relabel by listening to the audio and watching the video.

Inputs. You are given:
1. a short video clip (typically up to 30 seconds) with synchronized audio — watch and listen to it;
2. a list of candidate sound events, each with an idx, a class name, and onset/offset times in seconds (relative to the clip start at 0.0);
3. sometimes a NARRATION section (see below).

Guidelines.
Decisions — for each candidate choose one:
• **keep** — the event is present in the audio at roughly the indicated time, and its class label is correct. You can optionally refine the sound event class to make it more accurate.
• **drop** — the event is not actually present.
• **relabel** — a sound is audible at this time but the class name is wrong; the visual frames at the same timestamp support a different class. Use relabel whenever the class is wrong but a sound is present — it is usually a better answer than drop in that case.
How to decide:
• The candidate list comes from a high-recall audio model. Most events are real, but some are false positives, wrong-class, or duplicates.
• Listen to the audio at [onset, offset] before deciding. Faint, brief, distant, or background sounds count as present — do NOT drop just because a sound is subtle, as long as you can hear something matching the class.
• Drop when: (a) the audio at [onset, offset] is genuinely silent or contains no sound matching the class; (b) the candidate is an obvious duplicate of another candidate of the same class with near-identical timing AND the audio clearly contains only one such event.
• Relabel when: the audio has a clear sound at [onset, offset], but it's not what the class says — and the video frames disambiguate the true class.
• Keep when: you can hear a sound matching the class, even if faint or partially obscured.
Rules:
• Do not add new events that were not in the candidate list.
• Do not change onset or offset.
• Every candidate (every idx in the input) must appear in decisions exactly once.
• If two candidates of the same class overlap heavily and both look correct, keep both.

Using the narration. A NARRATION section may list what people DID and SAID during the clip (window-relative seconds), split into the WEARER (#C = the FIRST PERSON / camera wearer) and OTHER people (#O/#X, sometimes named). Use it as a STRONG HINT:
• #C or #O actions that make sound (handling objects, tools, walking, pouring) support KEEPING a co-occurring candidate of the matching class.
• When a candidate's class conflicts with what the narration says is happening at that time, prefer RELABEL (if the video supports the true class) or DROP.
• Narration is sparse and may miss events or be slightly off in time; OVERRIDE it when what you actually hear/see clearly conflicts. When no narration is provided, judge from audio+video alone.

Output format. Return STRICT JSON, in a single `json` fenced block, with this exact schema:
`{"decisions": [{"idx": <int>, "action": "keep", "refined_class": "<refined class label>", "reason": "<short reason>"}, {"idx": <int>, "action": "drop", "reason": "<short reason>"}, {"idx": <int>, "action": "relabel", "new_class": "<short class label>", "reason": "<short reason>"}]}`
Reasons should be one short sentence, no more than 120 characters. No prose outside the JSON block.

continued on next page

User-turn template (the clip's video+audio is attached):
`Clip: {chunk_label} ({duration_sec:.2f} s, video + audio)`
Candidate events (sorted by onset), as a `json` block — one entry per SED candidate:
`[{"idx": <int>, "class": "<class name>", "onset": <float>, "offset": <float>}, ...]`
NARRATION (what people did/said; window-relative seconds; #C = the wearer/first person, others are #O/#X):
WEARER / first person (#C) — actions & speech:
`{t:.1f} {text}`
OTHER people (#O/#X, may be named) — actions & speech:
`{t:.1f} [#{tag}] {text}`
`Return the JSON block only.`

Table 17: Prompt used to generate the audio timeline events using the dense audio caption sound events as described in §3.

Task. You are annotating a short audio clip (up to 30 seconds) from an egocentric recording. Write a SMALL NUMBER of coarse, scene-level captions with timestamps that summarize what is happening in the soundscape; together they should cover almost the entire clip except stretches of genuine silence. You HEAR the audio and can SEE the video, but the video is only for disambiguating what you hear.

Inputs. You are given:
1. the audio clip (video available as a disambiguation aid) — watch and listen to it;
2. a list of verified sound events with timestamps (from an SED model, model-audited; sorted by onset);
3. any transcribed speech (with lang and, when non-English, a text_en translation);
4. sometimes a NARRATION section (see below).

Guidelines.
Scene grouping is the primary task:
• Treat overlapping or interleaved fine-grained events as ONE layered scene. Name the dominant sound source; mention background/co-occurring sounds inside the SAME caption.
• Repeated instances of the same sound within a short window (repeated clinks, taps, footsteps) belong in ONE caption describing the pattern, NOT one caption per instance.
• A new caption is only warranted when the scene MEANINGFULLY changes: the dominant sound source changes, a new sonic environment begins, or a distinct isolated event punctuates the ambience.
Worked example (what "scene grouping" means): given events on 0–10 s (coughs at 1.1–2.3 and 4.1–4.7; dishwasher running 0.0–5.6; dish clanks at 0.8–1.2 and 4.5–5.0), WRONG is one caption per event; CORRECT is a single caption: "0.0 - 5.6, Dishes clank inside a running dishwasher while a man coughs briefly in the foreground."

continued on next page

Audio is the subject; video is only a disambiguation aid (strict):
• EVERY caption MUST describe a sound that is actually AUDIBLE. If something is visible but produces NO audible sound, DO NOT mention it at all.
• The video is allowed only to decide the SOURCE of a sound you actually hear (e.g. a metallic clang + a visible dishwasher rack → "dishes clank"). You may NOT add silent actions.
• Examples of what NOT to write: "The person applies yellow wood glue from a bottle" (visual, no distinctive sound); "A woman gestures toward the camera" (silent); "The person walks toward the kitchen" (unless footsteps are audible).
Speech content must be captured (when English or when translated):
• If English speech is audible (or the transcript's lang is en), include a short verbatim quote (<= 15 words, in double quotes) OR a one-sentence summary, inside the caption covering that timespan. Do NOT just write "a person speaks".
• If the speech is non-English, name the language and, if text_en is provided, include a short summary of the translated content; otherwise describe manner and prosody.
• The speech description belongs INSIDE the same layered caption, not as its own separate caption (unless speech is the ONLY thing happening for a long stretch).
Timestamps must be audio-anchored, not grid-snapped:
• Anchor caption boundaries to the ACTUAL audible transitions; the verified event list gives the dense onset/offset timestamps — USE THEM.
• Do NOT snap boundaries to multiples of 5 seconds. Prefer real audio-anchored values like 3.2, 11.7, 24.6. Timestamps are float seconds relative to clip start (0.0), one decimal.
Content style and coverage:
• Describe what the sound ACTUALLY SOUNDS LIKE (pitch, texture, rhythm, loudness, duration, timbre, spatial impression) — not just the activity. Distinguish foreground from background inside a single caption. Aim for 15–30 words per caption.
• The union of your caption spans should cover the audible portion of the clip; consecutive spans should be contiguous when scenes flow into each other; overlapping spans only when two independent layers cannot be merged.

Using the narration. A NARRATION section may list what people DID and SAID (window-relative seconds), split into the WEARER (#C = the FIRST PERSON) and OTHER people (#O/#X, sometimes named). Use it as a STRONG HINT for WHO is doing/saying what:
• #C actions (chopping, pouring, washing, typing) and #O actions help you name the correct sound SOURCE inside a caption; use the names to keep distinct people apart.
• When #C or an #O person is narrated talking around a speech segment, attribute the speech accordingly and capture its content in the caption covering that span.
• Narration describes VISIBLE actions/speech, NOT sounds — mention a narrated action only if it is actually AUDIBLE. OVERRIDE it when what you hear/see conflicts; when none is provided, judge from audio+video alone.

Output format. STRICT JSON in a single `json` fenced block: `{"captions": [{"start": 0.0, "end": 8.0, "text": "..."}]}`. No prose outside the JSON block.

User-turn template (video attached as disambiguation aid):
`Clip: {clip_label} ({duration_sec:.2f} s, audio with video available as disambiguation aid)`
Verified sound events (from an SED model, model-audited; sorted by onset), as a `json` block:
`[{"class": "<name>", "onset": <float>, "offset": <float>}, ...]`
Speech transcription (included only when speech is present), as a `json` block:
`[{"start": <float>, "end": <float>, "lang": "<lang>", "text_en": "<english>"}, ...]`
NARRATION (what people did/said; window-relative seconds; #C = the wearer/first person, others are #O/#X):
WEARER / first person (#C) — actions & speech:
`{t:.1f} {text}`
OTHER people (#O/#X, may be named) — actions & speech:
`{t:.1f} [#{tag}] {text}`
`Return the JSON block only.`

Table 18: Prompt used to tag different acoustic events in dense audio captioning to their sound sources as mentioned in §3. The injected {`taxonomy`} defines seven source keys: `self_body`, `self_speech`, `first_person_interaction`, `environment_ambient`, `electronic_media`, `other_speech`, `uncertain`.

Task. You classify the SOURCE of sounds in an egocentric (head-worn camera) video clip. The person wearing the camera is the 'wearer' / FIRST PERSON — you never see their face; you see the scene from their eyes and their hands. For EACH segment, watch what happens during its time window and listen, then assign its PRIMARY source class from the taxonomy (you decide how many apply — usually exactly one).

Inputs. You are given:
1. the CLIP itself (video with its audio) — watch and listen to it;
2. a numbered list of SOUND SEGMENTS already detected in the clip (id, [start - end] seconds, and a short description);
3. sometimes a NARRATION section (see below).

Guidelines.
Taxonomy (injected):
{`taxonomy`}
Decide with this procedure for every segment:
1. Is the sound SPEECH / a VOICE?
– Yes, and it is the WEARER'S voice (usually the LOUDEST, closest, most 'present' voice, with NO person visible in frame whose mouth moves in sync) → self_speech.
– Yes, and it is ANOTHER co-present person → other_speech.
– Yes, but it comes from a screen/speaker/recording → electronic_media.
2. Not speech? Then:
– the wearer's own BODY (breathing, footsteps, coughing, chewing) → self_body.
– the wearer ACTING on objects/environment (handling, tools, tap, typing, pouring, washing) → first_person_interaction.
– a device or media → electronic_media.
– anything else → environment_ambient.
The same description can recur at several segments; judge each on its own window. Use 'uncertain' only when you truly cannot decide.

Using the narration. A NARRATION section may list what people DID and SAID during the clip (window-relative seconds), split into the WEARER (#C = the FIRST PERSON / camera wearer — that is YOU/the viewpoint) and OTHER people (#O/#X, sometimes named like 'woman H'). Narration describes VISIBLE actions/speech, NOT sounds — use it as a STRONG HINT for WHO is doing/saying what:
• When the WEARER (#C) is narrated talking around a speech segment → that speech is self_speech. When an OTHER person is narrated talking → other_speech (use the names to keep distinct people apart).
• #C actions (kneading, picking, chopping, pouring, washing) support first_person_interaction for co-occurring object sounds; the wearer walking/breathing → self_body.
• #O actions (another person handling things, walking) support environment_ambient for their non-speech sounds; another person's speech → other_speech.
• Narration is sparse and may miss events or be slightly off in time; OVERRIDE it when what you actually hear/see clearly conflicts. When no narration is provided, judge by the audio/video alone.

Output format. STRICT JSON only, in one `json` fenced block, with one object for EVERY segment id given (same ids, no more, no fewer):

```
{{
"segments": [
{{
"seg_id": "<the exact id>",
"source_primary": ["<taxonomy key(s)>"],
"source_labels": ["<same as primary, plus any extra overlapping source>"],
"confidence": <0.0-1.0>,
"rationale": "<one short clause; cite the narrated actor if you used narration>"
}}
]
}}
```

continued on next page

User-turn template (the clip's video+audio is attached):
`CLIP DURATION: {duration_sec:.1f} seconds (the video with audio is attached).`
SOUND SEGMENTS ({n} total) — classify the source of each:
`{seg_id} [{start:.1f} - {end:.1f}] {description}`
... one line per segment ...
NARRATION (what people did/said; window-relative seconds; #C = the wearer/first person, others are #O/#X):
WEARER / first person (#C) — actions & speech:
`{t:.1f} {text}`
OTHER people (#O/#X, may be named) — actions & speech:
`{t:.1f} [#{tag}] {text}`
`Watch and listen to the attached clip, then return the JSON with one object per seg_id above. Return only the JSON block.`

Table 19: Prompt used to rewrite the audio timeline when merging multiple individually generated chunk audio timelines.

Task. You are cleaning up a list of audio captions that has been assembled from one or more 30-second source clips into a single ordered timeline. The listener will read your output start-to-finish as a single audio sample. Remove or rephrase any wording that assumes a different frame of reference (the original 30 s clip's start/end, or a per-clip event count).

Inputs. You are given:
1. The ordered caption list of one sample (text only; no audio, no source-chunk metadata), each caption with start, end, and text.

Guidelines.
Strict rules:
1. Do NOT change timestamps. Every caption's start and end must be identical to the input, to at least three decimal places. If you feel you need to change a timestamp, LEAVE THE CAPTION UNCHANGED.
2. Do NOT change the acoustic description (sound source, character, texture, layering) beyond removing stale positional wording.
3. Do NOT change speech content: preserve verbatim quotes (in double quotes) and one-sentence summaries of what was said.
4. Do NOT invent new information, sounds, or speakers. If a phrase is genuinely descriptive (e.g. "the tapping resumes after a brief pause"), keep it.
5. MERGE two captions if their context is seemingly the same and update their start and end times. *What to rewrite — remove or rephrase:*
• Position-relative to a chunk boundary: "towards the end", "at the very end", "just before the clip ends", "at the very beginning", "at the start of the clip".
• Cardinal/ordinal counters that referenced the source clip's event count: "for the first time", "for the second time", "the third instance", "for the last time", "once again", "yet again", "again" (when it counts events not sounds), "previously in this clip", "earlier in the clip", "later in the clip".
• Ordinal digits or short forms that count within the clip: "1st", "2nd", "3rd", "the third X".
Rephrase strategy: replace with a neutral equivalent when possible ("for the first time" → drop or "a"; "towards the end" → drop or a specific timestamp already carried in the timestamp field). If the phrase is only cosmetic, drop it and keep the sentence natural. If removing it would break the sentence, do the minimum rewrite needed to keep the acoustic meaning. If a caption has NONE of these patterns, return it byte-for-byte identical.

Output format. STRICT JSON in a single `json` fenced block, one item per input caption in the same order, with fields start, end, text (unchanged start/end): {`"captions": [{"start": 0.0, "end": 12.4, "text": "..."}]`}. No prose outside the JSON block. No trailing commentary.

continued on next page

User-turn template (text-only):

```
Sample id: {sample_id}
Total sample duration: {duration:.2f} s
Number of captions: {n}
```

Captions (ordered, timeline-local seconds), as a `json` block:

```
[{"start": <float>, "end": <float>, "text": "<caption text>"}, ...]
Rewrite each caption per the rules above. Preserve timestamps EXACTLY. Return the
JSON block only.
```

Table 20: Prompt used to generate fine-grained audio queries for temporal grounding using dense audio captions. We use $K = 2$.

Task. You are annotating temporal audio grounding (TAG) data. Given the audio and a text annotation of sound events, produce {K} temporal audio grounding QAs that locate specific sounds in time, or abstain if the annotation clearly mismatches the audio.

Inputs. You are given:
1. the audio recording — listen carefully;
2. a text annotation of DETECTED EVENTS (label → list of intervals) produced by an earlier automated model, which may be imperfect.

Guidelines.
Procedure: (1) LISTEN to the audio; (2) READ the annotation; (3) VERIFY whether it plausibly describes what you hear; (4) if clearly wrong or mismatched, ABSTAIN by returning an empty JSON array []; (5) otherwise produce {K} QAs, cross-checking each interval against the audio.
Merge recurrences across labels: the annotation may split the SAME underlying sound across multiple event classes (e.g. "Rub" and "Scrape"; "Speech" and "Male speech"). Gather intervals from EVERY label that describes the SAME acoustic phenomenon.
Do NOT invent events: every query MUST correspond to at least one event class actually listed in the DETECTED EVENTS block. Do not add sounds absent from the annotation, even if audible.
Avoid near-full-duration intervals: any single interval whose length exceeds 80% of {duration} is FORBIDDEN unless the source annotation itself lists an interval that long AND no shorter grouping is possible. Prefer tight, specific intervals aligned with actual bursts you hear.
Requirements for each QA:
• query: a short natural-language phrase (3–12 words) describing a specific perceivable sound the way a human listener would ("scratchy sanding sound on wood", "a man speaking", "sharp metallic clink"). Not a class name.
• intervals: list of [start, end] pairs (seconds, 1-decimal) covering EVERY interval where that sound occurs, as you HEAR them. Timestamps within [0.0, {duration}]. No interval may exceed 80% of {duration}.
• n_occurrences_hint: "multi" if intervals has 2+ entries, "single" if exactly 1.
• verified: true — on every emitted QA, to confirm you listened and cross-checked.
Verification, balance, abstain: seek to each proposed interval and confirm the sound is audible there; drop/trim accordingly. Aim for roughly half the queries with 2+ intervals and half with exactly 1. Return [] if the audio is silent/unintelligible while the annotation describes clear events, if the annotation describes events (e.g. English speech) not heard at all, or if annotation and audio disagree on more than roughly half of the described events.

Output format. Return ONLY a JSON array of {K} objects (or [] to abstain). No prose, no markdown fences. Example: `[{"query": "a scratchy sanding sound on wood", "intervals": [[0.8, 4.7],[12.1, 18.3]], "n_occurrences_hint": "multi", "verified": true}, {"query": "faint background rustling", "intervals": [[0.0, 30.0]], "n_occurrences_hint": "single", "verified": true}]`.

continued on next page

User-turn template (audio attached; appended below the instructions above):
`AUDIO CLIP: (attached --- please listen)`
`duration_sec: {duration}`
DETECTED EVENTS (label → list of intervals, from an automated model — verify against the audio):
`{events_block}` — one line per label: `- <label>: [<on>, <off>], [<on>, <off>], ...`
`Now produce the JSON array with exactly {K} queries, or [] if you must abstain.`

Table 21: Prompt used to generate coarse audio queries for temporal grounding using audio timeline events. We use $K = 2$.

Task. You are annotating temporal audio grounding (TAG) data. Given the audio and a coarse, time-aligned narrative, produce {K} temporal audio grounding QAs that locate specific sounds in time, or abstain if the narrative clearly mismatches the audio.

Inputs. You are given:
1. the audio recording — listen carefully;
2. a coarse, time-aligned narrative (COARSE CAPTIONS) produced by an earlier model, which may be imperfect or hallucinated.

Guidelines.
Procedure: (1) LISTEN carefully; (2) READ the coarse captions; (3) VERIFY whether they plausibly describe what you hear; (4) if clearly wrong or mismatched, ABSTAIN by returning []; (5) otherwise produce {K} QAs, cross-checking each interval against the audio.
Query style — specific, descriptive (critical): every query must read like a descriptive acoustic annotation sentence, NOT a terse class-like tag. Include the sound's texture, loudness, rhythm, and prominence, using what you HEAR. Good: "Gentle, calming music can be faintly heard in the background"; "A steady, coarse scraping of sandpaper on wood dominates the foreground"; "Sharp, sudden metallic clanks interrupt the ambience at irregular intervals". Bad (do NOT produce): "a woman laughing"; "sanding sound"; "background talking"; "car engine idling".
Extract specific sounds and find every occurrence: captions summarize whole scenes, so a single caption may contain multiple distinct sounds, and the SAME sound may recur across captions worded differently. Decompose the captions into specific perceivable sounds; collect intervals from EVERY caption whose scene contains that sound.
Do NOT invent events: every query MUST correspond to a sound explicitly mentioned or unambiguously implied in the COARSE CAPTIONS block. Do not add sounds absent from the captions, even if audible; the purpose is to ground and refine the captions, not extend them.
Avoid near-full-duration intervals: any single interval whose length exceeds 80% of {duration} is FORBIDDEN unless the underlying caption span itself is that long AND no tighter interval fits.
Requirements for each QA:
• query: a descriptive sentence (8–20 words) about ONE specific sound with acoustic modifiers (soft/loud/faint, rhythmic/irregular/continuous, sharp/gentle/muffled, foreground/background). NOT a class name or terse tag.
• intervals: list of [start, end] pairs (seconds, 1-decimal), TIGHT to the actual span you hear; no single interval may exceed 80% of {duration}; timestamps within [0.0, {duration}].
• n_occurrences_hint: "multi" (2+ intervals) or "single" (1).
• verified: true on every emitted QA.
Verification, balance, abstain: seek to each interval before writing; aim for roughly half multi and half single; return [] if the audio is silent while captions describe clear events, if captions describe events not heard at all, or if captions and audio disagree on more than roughly half the events.

Output format. Return ONLY a JSON array of {K} objects (or [] to abstain). No prose, no markdown fences. Example: `[{"query": "A steady, coarse scraping of sandpaper on wood dominates the foreground", "intervals": [[0.0, 12.5],[18.0, 30.0]], "n_occurrences_hint": "multi", "verified": true}, {"query": "A brief, muffled cough is barely audible in the background", "intervals": [[14.2, 14.9]], "n_occurrences_hint": "single", "verified": true}]`.

continued on next page

User-turn template (audio attached; appended below the instructions above):

```
AUDIO CLIP: (attached --- please listen)
duration_sec:  {duration}
```

COARSE CAPTIONS (from an automated model — verify against the audio):

{events_block} — one line per caption: `- [<on> - <off>] <caption text>`

```
Now produce the JSON array with exactly {K} queries, or [] if you must abstain.
```