Benchmarking Single-Factor Physical Video-to-Audio Generation

Supplementary Material

A. Project Webpage

We provide a project webpage that shows the physical correctness of current video-to-audio models through the following features:

- **Guided Physics Case Studies:** A set of curated examples from FlatSounds that isolate specific physical factors, helping readers understand physically-grounded audio generation.
- **Counterfactual Pairs Showcase:** A demonstration of time-aligned video pairs where a single physical variable is manipulated. By warping videos to align impact timings, these examples allow users to directly attribute acoustic differences to the controlled factor, illustrating the benchmark's core methodology.
- **Interactive Video Explorer:** A filtering interface to browse and compare samples from all tested models, under both captioned and no-caption conditions.
- **Quantitative Visualizations:** Interactive plots, including a trade-off analysis scatter plot and a multi-metric radar chart, that provide a holistic view of model performance across physical, semantic, and temporal dimensions.

B. Human Evaluation Details

We set up human evaluation as a pairwise preference test to validate our benchmark metric, using a representative subset of 40 video clips from FlatSounds. We chose pairwise comparisons because the alternative mean opinion score may not capture fine-grained differences between two models of roughly equal quality as each model's output is viewed in isolation.

Due to the increased number of comparisons necessary for a pairwise setup, we limit our evaluation to the captioned versions of MMAudio-Phys, MMAudio, FoleyCrafter, Hunyuan-V2A, and ThinkSound. For each comparison, we first randomly select two models, one of the 40 videos to compare, and one of the 10 randomly generated audio tracks for each model, conditioned on the selected video. The human rater is asked to select a preferred output based on plausibility, audio-video synchronization, and presence of auditory hallucinations. They are given the option to select the preferred output or "no preference". We show an example comparison screen in Fig. 1.

In total, we collected 578 valid pairwise comparisons from 382 workers (Fig. 2). The mean number of comparisons per worker was 1.5 with a median of 1. 31 workers were rejected for failing our noise test, where they must rate a video paired with brown noise as less preferred than the other video.

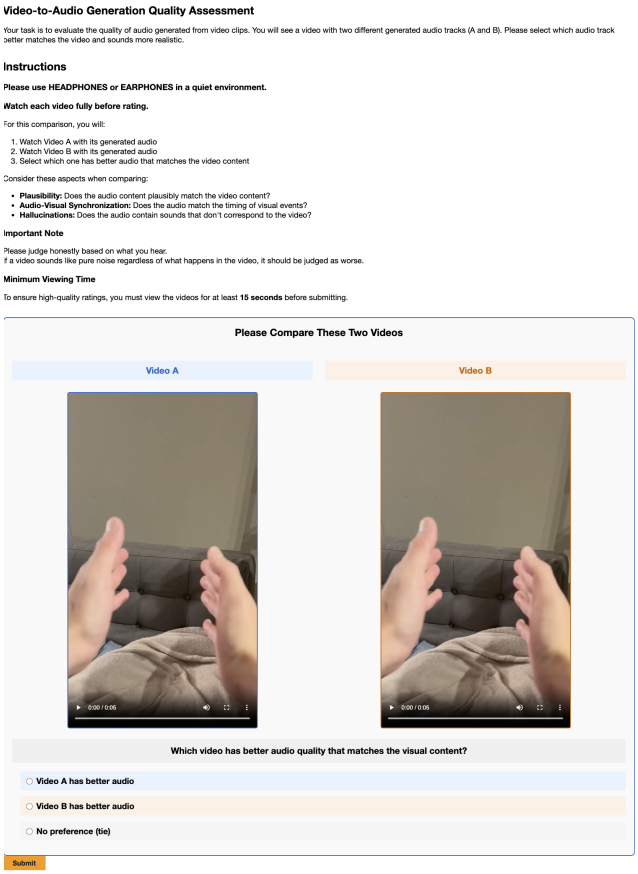


Figure 1. **Example pairwise comparison page as presented to raters during the human eval study.** Each rater watches both videos and selects the one that best fits the specified criteria. We use a minimum viewing time to prevent raters from rating without watching. We also include a listening test by inserting a video paired with pure brown noise, which we ask the rater to always rate as less preferred, otherwise they will be rejected.

ELO formulation. We compute ELO ratings to produce a global ranking of models from pairwise comparison data as an efficient means of summarizing the pairwise results. The ELO system assigns each model i a rating $R_i \in \mathbb{R}$, initialized to $R_i = 1500$. After each pairwise comparison between model A and model B, the ratings are updated based on the outcome.

Let R_A and R_B denote the current ratings of models A and B. The expected score of model A is defined as:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}},$$

and similarly for model B:

$$E_B = \frac{1}{1 + 10^{(R_A - R_B)/400}}.$$

These values represent the probability that each model would win under the current ratings.

After observing the result, the ratings are updated using a fixed learning rate $K = 32$:

$$R'_A = R_A + K \cdot (S_A - E_A),$$

$$R'_B = R_B + K \cdot (S_B - E_B),$$

where S_A, S_B are the outcome scores: - $S_A = 1, S_B = 0$ if A wins, - $S_A = 0, S_B = 1$ if B wins, - $S_A = S_B = 0.5$ in case of a tie.

This update is applied sequentially for all comparisons. The final ratings provide a latent performance score for each model, enabling a consistent global ranking even when not all models are directly compared.

C. Hit Detection Details

Here, we provide some more details on our setup for hit event detection from audio. Audio samples are first resampled to 44.1kHz, then we compute an STFT over the audio using a Hann window size of 1024 and hop size 256, and take the RMS along the time dimension of the STFT spectrogram to obtain a one-dimensional energy envelope.

However, peak detection performed this way is agnostic to anything other than the energy levels and is highly prone to false positives. Most of our cases look like the example in Fig. 3, which shows a *clink* sound of metal on glass against a relatively quiet backdrop and very distinct, clean peaks. However, we show an example in Fig. 4, which corresponds to the sound of a metal spoon hitting a metal cup four times. The decay of the metal-on-metal sound oscillates in such a way as to generate multiple sub-peaks on its way down. We are able to automatically reject most of these by spacing out our "hits" roughly half a second apart, and rejecting anything within half a second of a detected peak. However, it is difficult to tune this approach to be foolproof, as there are numerous valid sources of sounds other than those we annotated. As such, we measure the coverage/recall of annotated hits in our evaluation.

D. Dataset Distribution Details

FlatSounds comprises 185 unique indoor clips. A pie chart for object material appearance distributions is shown in Fig. 5, which shows the relative counts of each time an object of the specified material appears in a clip. Note that each object can appear in multiple clips, and most clips feature at least two interacting objects. Based on the pie chart, we can see that we heavily feature glass, metal, and wood. This is

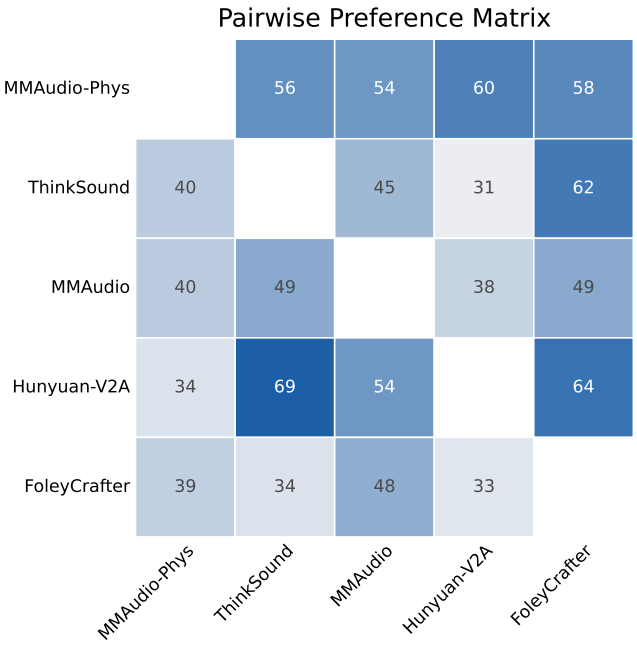


Figure 2. **Win-rates for pairwise preference test.** Each cell (i, j) represents how often the model for row i is preferred over the model for column j. Overall, the pairwise preferences broadly support the ranking trends reported in the main paper's Human Evaluation discussion. We refer the reader there for the final ELO-based ordering.

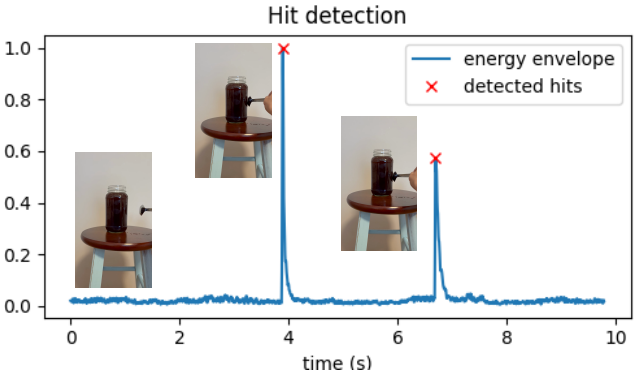


Figure 3. Visualization of a clean example of the energy envelope peak detection used to annotate sound event timing.

because we expect single-factor material changes between these materials to have relatively easy-to-detect differences with respect to our curated set of audio metrics. This is also reflected in the relatively large amount of counterfactual pairs with annotated expected changes in spectral-centroid as shown in Fig. 5. Room metrics such as DRR and RT60, which measure reverberation levels of an acoustic environment, appear less frequently in our benchmark due to the relative difficulty in conveying a room's acoustic properties unambiguously within a video. We rely primarily on hall-

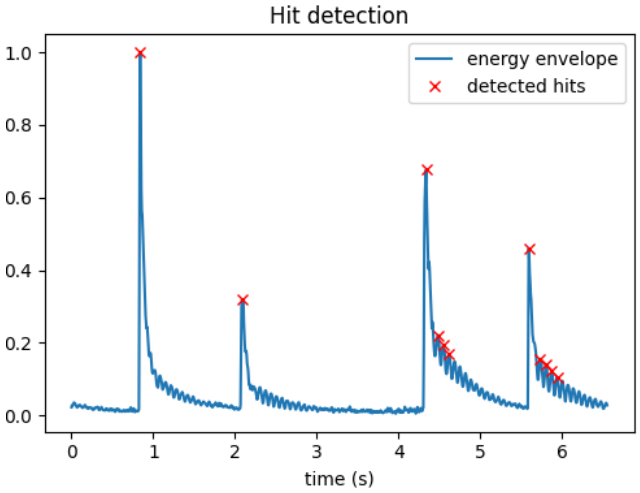


Figure 4. Example of a difficult case for automatically detecting sound events using our setup. The oscillations during the decay result in false positives.

way/stairwell environments as settings where one would expect higher reverb, but we will likely look into other creative alternative approaches as we attempt to expand the size of our benchmark.

E. Metrics Implementation Details

Our benchmark evaluates physical correctness through nine acoustic metrics computed using robust signal processing techniques (Fig. 6, 7, and 8). We employ a hybrid computation strategy: six metrics use *per-hit averaging* (attack time, decay rate, fundamental frequency, spectral centroid, spectral rolloff, spectral flux), while temporal modulation is computed globally. RT60 and DRR remain room-acoustics metrics, but in multi-hit clips we estimate them in a hit-aware manner and aggregate them across hits. To ensure consistent spectral analysis across models with varying native sampling rates, all audio is resampled to 16kHz prior to analysis, with unified parameters (64ms analysis window and 8ms frame shift).

E.1. Per-Hit Segmentation Strategy

For multi-hit videos, the audio track is segmented at annotated hit times $\mathbf{t} = [t_1, t_2, \ldots, t_N]$. Each segment s_i is defined to span from 50ms before the hit t_i to the next hit t_{i+1} (or the signal end). A default dynamic window adapted to the estimated RT60 is applied:

$$s_i = \text{audio}[\max(0, t_i - 50\text{ms}) : \min(t_{i+1} - 20\text{ms}, t_i + w_{\text{dur}})] \quad (1)$$

Metrics are computed independently on each segment. For the per-hit metrics below, the resulting hit-level values are averaged across valid hits to obtain a video-level score.

E.2. Attack Time

We define the Attack Time as the duration required for the signal envelope to rise from 10% to 90% of its peak value. The computation process is detailed in Algorithm 1.

Algorithm 1 Attack Time Calculation

Input: Signal segment y, sampling rate f_s
$\text{env} \leftarrow \text{GaussianSmooth}(|\text{Hilbert}(y)|, \sigma = 3\text{ms})$
Step 1: Onset Detection via Derivative Gating
$\mu_{\text{noise}}, \sigma_{\text{noise}} \leftarrow \text{Median}(\text{env}_{\text{pre}}), \text{MAD}(\text{env}_{\text{pre}})$
$\text{env}' \leftarrow \nabla \text{env}$
$\sigma_{\text{der}} \leftarrow \text{MAD}(\text{env}'_{\text{pre}})$
$i_{\text{onset}} \leftarrow$ First i where $(\text{env}[i] > \mu_{\text{noise}} + 3\sigma_{\text{noise}}) \wedge (\text{env}'[i] > 3\sigma_{\text{der}})$
Step 2: Peak Finding & Monotonicization
$i_{\text{peak}} \leftarrow \arg\max_{i \in [0, 200\text{ms}]} \text{env}[i_{\text{onset}} + i]$
$\text{env}_{\text{mono}} \leftarrow \text{CumulativeMax}(\text{env}[i_{\text{onset}} : i_{\text{peak}}])$
Step 3: Rise Time Calculation
$i_{10} \leftarrow \min\{i : \text{env}_{\text{mono}}[i] \geq 0.10 \times \text{peak}\}$
$i_{90} \leftarrow \min\{i : \text{env}_{\text{mono}}[i] \geq 0.90 \times \text{peak}\}$
Return: $1000 \cdot (i_{90} - i_{10})/f_s$

E.3. Decay Rate

We quantify the exponential energy dissipation of an impact event as $A(t) = A_0 e^{-\lambda t}$. The decay parameter λ is estimated using a robust hierarchical fitting strategy described in Algorithm 2.

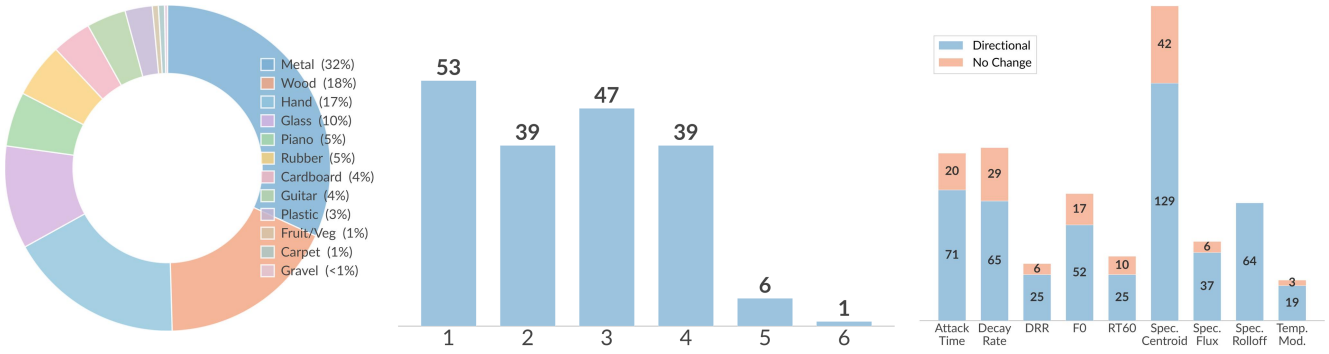
Algorithm 2 Decay Rate Estimation

Input: Signal envelope env
Preprocessing:
$\text{env}_{\text{norm}} \leftarrow \text{CumulativeMin}(\text{env}[i_{\text{peak}} :]/\text{env}[i_{\text{peak}}])$
$\text{env}_{\text{dB}} \leftarrow 20 \log_{10}(\text{env}_{\text{norm}})$
Hierarchical Fitting (T30/T20/T10):
for range $\in \{[-5, -35], [-10, -30], [-5, -25]\}$ **do**
  $i_{\text{start}}, i_{\text{end}} \leftarrow \text{FindCrossings}(\text{env}_{\text{dB}}, \text{range})$
  if $\text{Length}(i_{\text{start}}, i_{\text{end}}) \geq 6$ **then**
    $m \leftarrow \text{TheilSenRegression}(\text{env}_{\text{dB}}[i_{\text{start}} : i_{\text{end}}])$
    if $m < 0$ **and** $|m| > 10^{-6}$ **then**
      $\lambda \leftarrow -m/(20/\ln 10)$
      Return $\text{Clip}(\lambda, 0.02, 50.0)$
    end if
  end if
end for

E.4. Fundamental Frequency (F0)

We employ a two-tier strategy to handle both harmonic and non-harmonic sounds robustly.

- **Parselmouth Autocorrelation:** We first extract a 300ms segment starting from the onset, discarding the initial



(a) Object Material Distribution (b) Hit Count Distribution (c) Metrics Distribution

Figure 5. Distribution of (a) material types, (b) hit counts, and (c) metric-change annotations used to create counterfactual pairs. A pair may contain one or more metric change annotations, and the same object instance may appear in multiple videos, with each video featuring one or more interacting objects.

10ms transient. We compute the pitch using Praat-style autocorrelation [4] within the range [27.5, 4186] Hz. We identify valid voiced frames $\mathcal{F}$ and compute the voiced ratio r_{voiced}. If $r_{\text{voiced}} \geq 0.1$ and at least three voiced frames are present, we compute F0 as the 10% trimmed mean of $\mathcal{F}$. Additionally, we apply an octave error correction step: if F0 $>$ 1200 Hz, we check divisors $[2, 3, 4, 6, 8]$ and select the first candidate that falls within [80, 1500] Hz.

- **Modal Peak Detection (Fallback):** If the voiced evidence is insufficient, we fall back to a spectral peak detection method. We extract a short 20-110ms window and compute the Welch Power Spectral Density (PSD) [13]. We then detect peaks in the [80, 4000] Hz range that exceed an adaptive threshold $\tau = \text{median(PSD)} + 2.5 \times \text{MAD(PSD)}$ and return the frequency of the lowest significant peak.

E.5. Spectral Centroid & Rolloff

We compute both spectral metrics using a unified processing pipeline to capture the timbre of the impact's sustain phase [6]. We extract a fixed analysis window y_{timbre} spanning from $t_{\text{onset}} + 60$ms to $t_{\text{onset}} + 180$ms to avoid the initial transient broadband noise. After removing the DC offset, we compute the Short-Time Fourier Transform (STFT) using an FFT size of $N_{\text{fft}} = 1024$ and a hop size of 128 samples.

Spectral centroid. We calculate the spectral centroid $\bar{f}[n]$ for each frame n as the weighted mean of the component frequencies, representing the mass of the spectrum center:

$$\bar{f}[n] = \frac{\sum_{k=0}^{N_{\text{fft}}/2} f[k] \cdot |S[k,n]|}{\sum_{k=0}^{N_{\text{fft}}/2} |S[k,n]|} \tag{2}$$

where $f[k]$ represents the center frequency of bin k, and $|S[k,n]|$ denotes the magnitude of the STFT at bin k and frame n.

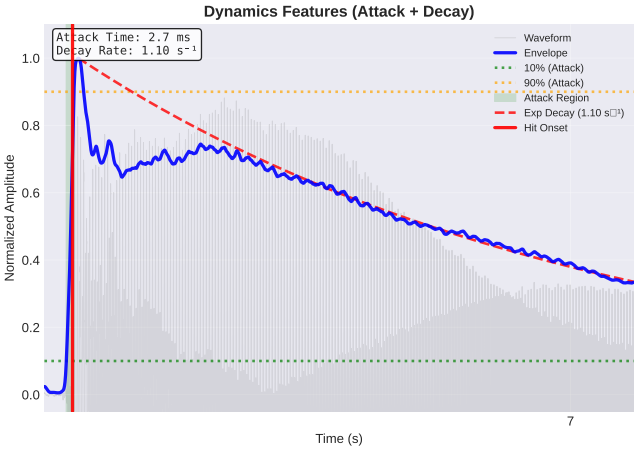


Figure 6. Plot showing the Attack time and Decay Rate of a detected hit.

Spectral rolloff. We compute the spectral rolloff $f_{85}[n]$ as the frequency threshold below which 85% of the total spectral magnitude is contained:

$$\sum_{k=0}^{K_{85}} |S[k,n]| \geq 0.85 \sum_{k=0}^{N_{\text{fft}}/2} |S[k,n]| \tag{3}$$

where K_{85} is the bin index corresponding to $f_{85}[n]$.

Finally, we filter out invalid frames (e.g., silent frames resulting in division by zero) and aggregate the frame-wise values into a single scalar using a 10% trimmed mean to ensure robustness against outliers.

E.6. Spectral Flux

We measure the strength of the attack transient by analyzing the rate of spectral change. An attack window of 180ms is

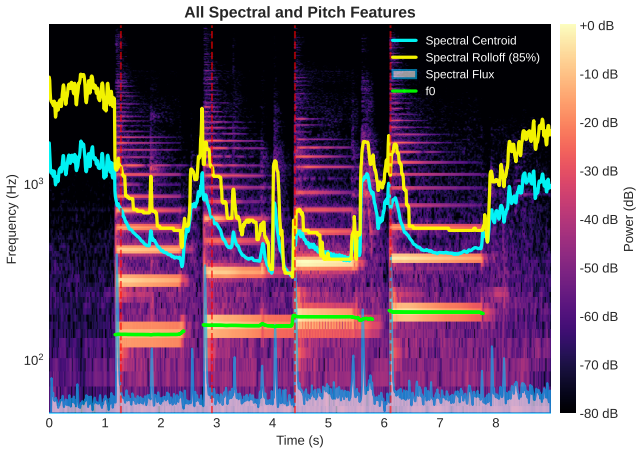


Figure 7. Plot showing all the spectral features, and the F0 contour overlaid on the audio spectrogram.

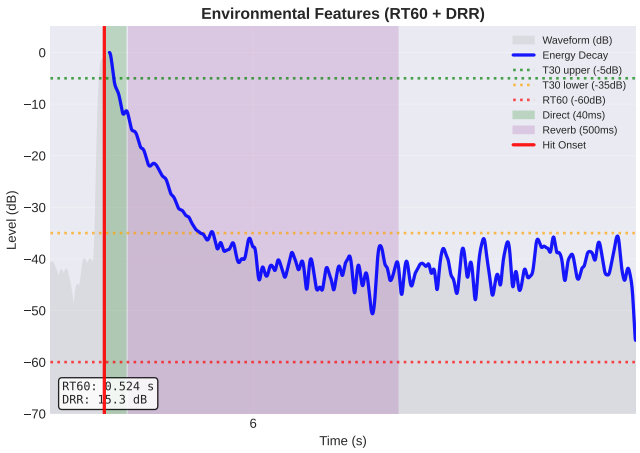


Figure 8. Plot showing RT60 and DRR measures of a detected hit.

extracted from the onset and RMS normalized. We then compute an onset-strength sequence,

$$\mathrm{OS}[n] = \sum_{k=0}^{N_{\text{fft}}/2} \max(0, |S[k,n]| - |S[k,n-1]|) \qquad (4)$$

which serves as our spectral-flux proxy. We retain positive onset-strength values and apply Median Absolute Deviation (MAD) filtering to remove noisy frames [7]. The final metric is the mean of the retained onset-strength values.

E.7. Reverberation Time (RT60)

We estimate the reverberation time using Schroeder integration [10] combined with a hierarchical fitting approach.

Energy decay curve. We first compute the Energy Decay Curve by integrating the squared signal energy backwards from the end of the signal to the peak amplitude N:

$$E[i] = \sum_{j=i}^{N} y[j]^2 \qquad (5)$$

This curve is normalized and converted to dB scale. We also estimate the noise floor μ_{noise} from the last 10% of the signal to determine a validity threshold.

Hierarchical linear fitting. We attempt to fit a linear slope to the decay curve over three standard ranges: T30 (-5 to -35 dB), T20 (-5 to -25 dB), and T10 (-5 to -15 dB). For each range, segment length and duration are verified against minimum quality thresholds. Linear regression is performed to find the slope m, requiring $R^2 \geq 0.9$. RT60 is calculated as $-60/m$. If the T30 fit fails, the system progressively falls back to T20 and then T10.

E.8. Direct-to-Reverberant Ratio (DRR)

DRR calculation involves separating the signal into direct and reverberant components based on temporal windows. We define the direct sound window w_{direct} as 40ms (for 16kHz audio) and adaptively set the reverberant window w_{reverb} based on the estimated RT60.

We also apply a perceptual bandpass filter (125-4000 Hz) to both segments. The final DRR is computed as an energy ratio,

$$\mathrm{DRR} = 10 \log_{10} \left(\frac{\sum_n d[n]^2}{\sum_n r[n]^2} \right), \qquad (6)$$

where $d[n]$ and $r[n]$ denote the direct and reverberant segments, respectively. The result is clipped to the range $[-20, 40]$ dB.

E.9. Temporal Modulation

We combine three complementary measures to capture envelope variations: the Coefficient of Variation (CV), the Peak Factor (PF), and the Modulation Spectrum Energy (E_{mod}). We first extract the Hilbert envelope, apply anti-aliasing, and downsample it to approximately 200Hz.

- **CV:** Calculated as the standard deviation of the high-passed envelope divided by the mean of the downsampled envelope.
- **Peak Factor:** Defined as the 99th percentile of the envelope divided by its RMS value.
- **Modulation Spectrum:** We compute the FFT of the envelope (P_{mod}) and calculate the energy ratio in the 4-16 Hz band:

$$E_{\text{mod}} = \frac{\sum_{f=4}^{16} P_{\text{mod}}[f]}{\sum_{f>0} P_{\text{mod}}[f]} \qquad (7)$$

These three components are combined into a single weighted score, with the modulation-spectrum term heuris-

tically upweighted to emphasize rhythmic structure:

$$\text{ModIndex} = 0.85 \times (0.4\text{CV}_{\text{norm}} + 0.3\text{PF}_{\text{norm}} + 0.6E_{\text{mod}}) \tag{8}$$

E.10. Robust Statistical Methods

Throughout our pipeline, we employ robust statistics to ensure metric stability against noise and outliers. In particular, MAD-based thresholds and estimators are used extensively to suppress noisy frames and outliers [7]. We also utilize Theil-Sen regression [11] for slope estimation, which offers a breakdown point of 29% compared to 0% for ordinary least squares, making it highly effective for noisy decay curves.

E.11. Onset Detection

We implement a hybrid onset detection strategy to align generated audio with expected visual events. We primarily use an onset-strength-based detector with high temporal resolution (≈3.3ms), and fall back to an envelope-based peak detector with adaptive thresholding when onset detection is unreliable. A greedy matching algorithm with adaptive temporal tolerance (100-250ms, depending on hit density) is then used to align detected audio onsets with ground-truth video timestamps.

E.12. Temporal Alignment Metrics

Once onsets are detected and aligned, we compute three metrics to quantify synchronization quality:

- **Hit Coverage:** This metric measures the recall of sound events. We consider a ground-truth event "covered" if at least one detected onset falls within an adaptive tolerance window around its annotated timestamp. Hit Coverage is simply the percentage of ground-truth events that are successfully covered.
- **Timing Error:** For the subset of successfully covered events, we calculate the absolute temporal deviation between the ground-truth timestamp and the nearest detected onset. The Timing Error is reported as the mean deviation in milliseconds.
- **Perfect Alignment:** To assess sequence-level consistency, we define Perfect Alignment as the percentage of generated clips where *all* ground-truth events are successfully covered (i.e., 100% Hit Coverage).

E.13. Quality Weighting Framework

We implement a quality weighting framework based on the premise that measuring fine-grained physical correctness is futile if the generated audio lacks basic temporal alignment or semantic accuracy. For example, analyzing the decay rate of a sound is meaningless if the sound onset is missed or if the generated class is incorrect. To address this, we down-weight the contribution of low-quality generations in our final confidence scores.

Quality weight calculation. For each seed in a factual-counterfactual pair, we compute two scalar terms. The temporal term is defined as the minimum of the factual and counterfactual Hit Coverage scores for that seed. Similarly, the semantic term is defined as the minimum of the factual and counterfactual CLAP similarity scores for that seed. For single-video tests, the corresponding terms are taken from the single generated clip.

The final per-seed quality weight is the arithmetic mean of these two components ($w_i = 0.5w_{\text{temporal},i} + 0.5w_{\text{semantic},i}$). These per-seed weights are then incorporated into the voting procedure for the physical metrics while retaining all seeds in the denominator. This soft weighting approach allows us to preserve all samples in the evaluation while ensuring that the reported confidence reflects not just physical responsiveness, but also the fundamental quality of the generation.

Dimension-aware handling. For seeds that fail to produce valid hits (< 2 detected hits), we set per-hit metrics (attack time, decay rate, F0, spectral centroid, rolloff, flux) to NaN and count them as failures in the denominator. However, room-acoustics metrics (RT60, DRR) and temporal modulation are still computed for these seeds, as they do not strictly require precise hit timing. All seeds, regardless of validity, count in the denominator when computing confidence scores.

F. Physics-aware Caption Details

To generate the physics-aware captions used in our study (e.g., for fine-tuning to get MMAudio-Phys), we employ a multi-stage pipeline for our audio-visual data. This process is designed to first capture modality-specific details from the audio and video streams independently, and then fuse them into a single physically grounded caption.

F.1. Modality-Specific Caption Generation

Audio captioning. First, we generate audio-only captions. This is accomplished by processing the soundtrack of each clip through the Qwen3-Omni model (Omni-Captioner) [8, 14] to produce a descriptive caption focused solely on the acoustic events. Note that this model does not accept text prompts and only takes audio as input.

Video captioning. Concurrently, we generate visual-only captions from the silent video frames using Qwen3-VL [12]. This model is guided by a comprehensive system prompt that enforces a visual-only analysis, with emphasis on physics, materials, and spatial reasoning. The specific prompts used for this stage are detailed below.

System Prompt (Qwen3-VL)

You are an advanced, visual-only AI specialized in generating descriptive captions for silent videos. Work strictly from visual evidence (no audio; ignore OCR unless text is clearly legible) and be physics- and materials-aware.

Objectives:

- Describe what is happening (people, objects, actions, motions, interactions).
- Capture spatial layout and physics-informed context (forces, constraints, material behavior).
- Produce immersive, human-readable captions for accessibility, indexing, and understanding.

Guidelines:

Visual Content.

- Identify main subjects: people, animals, objects, vehicles, scenery.
- Describe actions and gestures precisely (e.g., ‘‘a person raises a hammer and strikes’’).
- Include salient visual details: clothing, facial expressions, colors, textures, clearly readable text/logos.

Spatial / Physical Reasoning.

- Infer environment layout (indoor/outdoor, approximate space shape, obstacles, dominant surfaces).
- Use cues like shadows, reflections, lighting falloff, occlusion, deformation, motion trajectories.
- Indicate relative positions/distances/sizes when visually supportable.
- Note dynamics implying forces/constraints (accel/decel, recoil, friction/traction, elastic vs. inelastic contact, balance).

Materials Awareness.

- When justified by visuals, identify likely materials (metal, wood, glass, plastic, fabric, stone, concrete, rubber, water/fluids).
- Base calls on visible cues: reflectance/specular highlights, translucency, grain/texture, deformation, wear/scratches, corrosion, residue/dust, joint types, fracture/splinter patterns, splash/flow/viscosity.
- If motion suggests material interaction (e.g., rubber skids on asphalt, glass shatters), briefly add that physical interpretation.

Environmental Context.

- Name the setting when supported (workshop, office, street, kitchen, forest, stadium, etc.).
- Use indicators (walls, furniture, vegetation, weather, signage) to contextualize.
- Mention visible dynamics: dust, smoke, vibration, spray, lighting changes.

Repetition & Counting.

- If an action repeats and the repetitions are clearly distinguishable, include the count (e.g., ‘‘knocks twice’’, ‘‘jumps three times’’).
- If the exact count is uncertain (due to occlusion, motion blur, or ambiguity), do not give a number; use qualitative phrasing like ‘‘repeatedly’’, ‘‘in quick succession’’, or ‘‘several times’’.

Inference Discipline (‘‘two-cue rule’’).

- You may include at most two cautious inferences in total that are not directly observed but are strongly suggested by ≥ 2 visual cues each (optionally state the cues briefly).
- Example: ‘‘likely steel (sharp specular reflections; no visible flex under load)’’.
- If cues are insufficient or confidence is low, omit the inference.
- Never infer identities, ages, or private attributes; avoid lip reading or audio-based claims.

Style & Length.

- Clear, precise, natural language; concrete verbs over vague phrasing.
- Default to a longform caption: multi-sentence (and multi-paragraph if needed) with layered detail on actions, space, materials, and physical interactions.
- Target length: $\sim$150–300 words for typical clips; may extend for complex scenes. Avoid filler and speculation.
- Use approximate metrics only when visually justified.

Output.

- Provide a longform caption (no time codes). If the user explicitly requests ‘‘brief’’, supply a shorter single-paragraph version.

User Prompt (Qwen3-VL)

You will receive a silent video. Please generate a longform caption following the system rules, with emphasis on spatial layout, materials, physics-aware details, and counting clear repetitions.

Requirements for this video:

- Visual-only analysis; do not assume any audio.
- Highlight spatial relationships, material properties, and physically plausible interactions.
- If an action repeats and repetitions are clearly visible, include the count; if the count is uncertain, describe it qualitatively without a number.
- Apply the two-cue rule: include up to two cautious inferences total, each supported by at least two visible cues; omit any inference you are not confident about.

- Default to a longform caption (approximately 150-300 words; longer if complexity warrants).

Output:
- A longform caption with no time codes.

F.2. Audio-Visual Caption Fusion

After generating the separate audio-only and video-only captions, we fuse them into a single, coherent, physics-aware caption. This fusion is performed by GPT-OSS [1]. The model is given a highly constrained system prompt to act as a technical expert, synthesizing the two text inputs into an objective, audio-anchored, and physically grounded event log. The prompts for this fusion stage are provided below.

System Prompt (GPT-OSS)

You are an expert AI for producing one objective, physically grounded technical caption by fusing two text-only inputs: (1) an audio-only caption and (2) a video-only caption.

Primary Objective. Output only a concise, technical Final Caption that reads like an acoustic event log or a forensic analysis. Your task is to synthesize the most plausible physical event described by the two input texts. The Final Caption must:
- infer object materials and environment from the textual evidence, embedding these facts directly and economically into the caption;
- precisely count and disambiguate distinct audio events and their physical causes as described in the captions;
- follow the constraint: no meta-reasoning, no analytical commentary, and no literary or poetic language; write only the objective physical description.

Length & Structure (STRICT).
- Produce 80-100 words in English.
- Write in a single paragraph using clear, declarative sentences; avoid complex clauses and narrative storytelling.
- Focus on a clear chronological and causal sequence of events based on the fused information.

Information Fusion Protocol (STRICT).
- The two provided captions are your **only** source of information. Do not assume details not present in them.
- Synthesize a single, coherent description of the most physically plausible event consistent with both captions.
- Conflict resolution: if the captions conflict, prioritize the audio caption for acoustic details (what a sound is) and the video caption for spatial/object details (where an object is). Resolve contradictions through logical inference to create the most likely scenario. Do not mention the conflict itself or the source captions in the final output.
- Remove redundancy: if both captions mention the same event, merge the details into a single, efficient description.

Audio-Anchored Content Policy (STRICT).
- Admission test for entities: an entity from the captions is included in the final output only if it meets at least one of these criteria based on the input texts:
  1. it emits sound; or
  2. it is in direct contact with a sounding source at the moment of sound; or
  3. it shapes the acoustics (reflects/absorbs/occludes/transmits) needed to explain a described sound; or
  4. it is the agent of a described sounding action.

  If none apply, exclude the entity from the final caption.
- Silent-object cap: at most two brief mentions of non-sounding context surfaces (e.g., ‘‘concrete floor’’, ‘‘glass wall’’). Use generic superclasses only (metal/wood/plastic/glass/stone/rubber/fabric), no colors, branding, micro-textures, or fine detail for silent items.
- Sentence anchoring: each sentence must contain a sound-linked action, count, acoustic cue, or causal link to a sounding event described in the captions.
- Material inference scope: infer or describe materials only for sounding objects or the immediate contact surfaces involved in those sounds, as suggested by the input texts.

Vocabulary & Granularity.
- Controlled terms: metal (steel/aluminum), wood (solid/plywood), plastic (rigid/flexible), glass, ceramic, stone, concrete, rubber, fabric/textile, leather, paper/cardboard, water/ice.
- Composites allowed (e.g., ‘‘wood tabletop with metal fasteners’’).

Hedging Policy (STRICT).
- Do not use hedging words (‘‘likely’’, ‘‘probably’’, ‘‘appears’’, ‘‘seems’’, etc.). If uncertain about a detail due to conflicting or vague captions, fall back to a more generic description or omit the detail.

Style (STRICT, Technical Description).
- Third person, present tense.

- Tone must be clinical and direct, prioritizing physical accuracy over narrative flair. Avoid literary or poetic language.

Output Requirement.
- Output only the Final Caption paragraph, as specified above. No lists, no headings, no time-codes.

User Prompt (GPT-OSS)

Here are the two pre-generated captions describing the same event.
Audio-only caption: {audio_caption}
Video-only caption: {video_caption}
Task: Fuse the two captions to produce ONE Final Caption.
Requirements:
- Language & length: Write 80-100 words in English, in a single paragraph.
- Source of Truth: The provided captions are your only source of information. Synthesize the most plausible physical event they describe.
- Tone: The caption must be a technical, objective description of acoustic events and their physical causes. AVOID literary, poetic, or overly descriptive language.
- Content: Strictly follow the audio-anchored policy. Infer materials only for sounding objects or their immediate contact surfaces as suggested by the captions.
- Synthesis: Do not simply copy phrases from the input captions. Write a new, coherent description that logically integrates the information from both.
- No hedging or meta language. No lists, no headings, no time-codes.

Output: Final Caption

F.3. Caption Comparison

To visually demonstrate the efficacy of our captioning pipeline, we compare original captions against our physics-aware captions. Fig. 9 presents a qualitative comparison across three examples, sourced from AudioCaps [5], VGGSound [2], and the Greatest Hits [9].

Our generated captions explicitly ground acoustic events in their physical causes. For the train example (top), the caption captures dynamic acoustic shifts consistent with the Doppler effect and specifies material interactions like "steel wheels" on "steel rails", which is not covered in the original "whistles twice". Similarly, in the banjo clip (middle), the pipeline identifies the instrument's construction, explicitly citing the "metal capo" and "synthetic-skin drumhead" to explain the "bright metallic timbre". Finally, in the cloth example (bottom), our model details the contact dynamics, describing the "high-frequency rubber-like squeak" and "frictional rustle" of wood sliding against a mattress.

Table 1. Overview of datasets used in MMAudio-Phys.

Dataset	Type	Hours
WavCaps	Audio-Text	1739.2
FreeSound	Audio-Text	625.6
AudioSet	Audio-Text	436.5
VGGSound	Audio-Video	405.8
AudioCaps	Audio-Text	117.8
UrbanSound8K	Audio-Text	18.9
Greatest Hits	Audio-Video	6.4
ESC-50	Audio-Text	4.2
MusicInstrument	Audio-Text	4.1
Sound of Water	Audio-Video	2.1
Total	/	3360.6

G. MMAudio-Phys Training Details

To test whether explicit physics-aware textual conditioning will improve semantic and physical correctness, we employ MMAudio [3] as the base architecture for our experiments. Our primary objective is not to propose a new model architecture, but to isolate the impact of enhanced caption quality on physical performance. MMAudio serves as an ideal test bed for this purpose due to its classic latent diffusion transformer design, accessible codebase, and state-of-the-art performance on standard benchmarks.

Training data. As summarized in Table 1, our training dataset comprises about 3,360 hours of audio-visual and audio-text pairs sourced from 10 diverse datasets. To balance the data distribution, we oversample VGGSound samples by a factor of 3.

Implementation details. We fine-tune the pre-trained MMAudio `large_44k` checkpoint, which features a 21-layer transformer architecture (14 heads, 896 hidden dimension) operating on a latent sequence length of 345. The model processes 44.1kHz audio conditioned on both video features (CLIP: 64 frames, Synchformer: 192 frames) and text embeddings (1024-dim). Training is conducted with a total batch size of 256 for 700k iterations. We optimize using AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$, weight decay 10^{-6}) with a learning rate of 10^{-4}. The schedule includes a 1k-step linear warmup, constant rate until 400k steps, and subsequent decays by 0.1 at 500k and 600k steps. Classifier-free guidance is enabled by dropping input conditions with $p = 0.1$, and exponential moving average (EMA, $\sigma_{\text{rel}} \in \{0.05, 0.1\}$) is applied throughout training. All training is performed using mixed-precision and the flow matching objective.

H. Inference Setting

For all tested models, we adhere to the default inference parameters specified in their official codebases. In the cap-

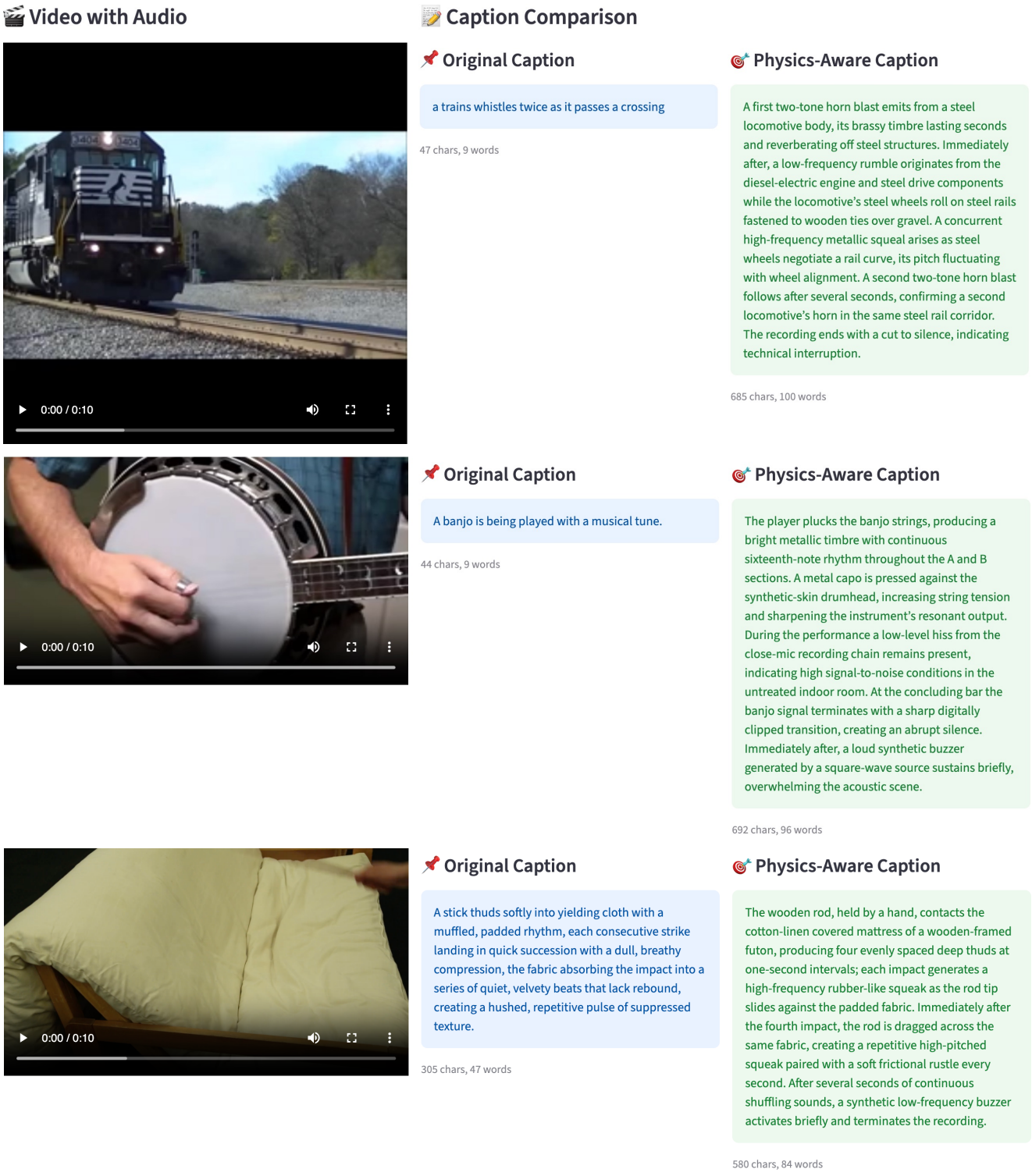


Figure 9. Qualitative comparison of original and physics-aware (our) captions for train (top), banjo (middle), and cloth (bottom) samples.

tioned setting, each test case is paired with one fixed descriptive caption that is shared across all methods to maintain a standardized inference interface. For ThinkSound, specifically, we use the provided chain-of-thought (CoT) prompts on the standard VGGSound benchmark. To ensure a fair evaluation across our FlatSounds benchmark, we

utilize a general CoT prompt for this model: ‘‘Generate high-quality audio that matches the visual content’’. Notably, we observe that ThinkSound exhibits limited sensitivity to variations in the CoT prompt within the FlatSounds benchmark. This behavior might stem from saturation in its training distribution, a phenomenon not observed in other evaluated models.

References

[1] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. 8

[2] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725, 2020. 9

[3] Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Mmaudio: Taming multimodal joint training for high-quality video-to-audio synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28901–28911, 2025. 9

[4] Yannick Jadoul, Bill Thompson, and Bart De Boer. Introducing parselmouth: A python interface to praat. *Journal of Phonetics*, 71:1–15, 2018. 4

[5] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long and Short Papers)*, pages 119–132, 2019. 9

[6] Alexander Lerch. *An Introduction to Audio Content Analysis: Music Information Retrieval Tasks and Applications*. Wiley-IEEE Press, Hoboken, N.J, 2 edition, 2023. 4

[7] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013. 5, 6

[8] Ziyang Ma, Ruiyang Xu, Zhenghao Xing, Yunfei Chu, Yuxuan Wang, Jinzheng He, Jin Xu, Pheng-Ann Heng, Kai Yu, Junyang Lin, et al. Omni-captioner: Data pipeline, models, and benchmark for omni detailed perception. In *International Conference on Learning Representations (ICLR)*, 2026. 6

[9] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2405–2413, 2016. 9

[10] Manfred R Schroeder. New method of measuring reverberation time. *The Journal of the Acoustical Society of America*, 37(3):409–412, 1965. 5

[11] Pranab Kumar Sen. Estimates of the regression coefficient based on kendall's tau. *Journal of the American Statistical Association*, 63(324):1379–1389, 1968. 6

[12] Qwen Team. Qwen3-vl: the multimodal large language model series. https://github.com/QwenLM/Qwen3-VL, 2025. 6

[13] Peter Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 2003. 4

[14] 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*, 2025. 6