Supplementary Material for Pushing the Frontier of Audiovisual Perception with Large-Scale Multimodal Correspondence Learning

A. Overview

This supplementary material is organized as follows. We first discuss the related work in §B. Then we provide the details of building PE-AV's audiovisual data engine and the stage-1 and stage-2 prompts used for generating synthetic captions in §C. In §D, we present details of frame-level $\text{PE}_{\text{A-Frame}}$, where we leverage frame-level contrastive objective to enable fine-grained audio-to-text alignment. We evaluate $\text{PE}_{\text{A-Frame}}$ in the downstream sound-event-detection task in §E. In §F, we provide additional implementation details of PE-AV including the full hyperparameter setup, training recipe (§F.1), and an efficient implementation (§F.2) to expand sigmoid contrastive loss for audio-video-text training. We also provide more details for our evaluation protocol to ensure reproducibility in §F.3. Finally, we present additional experiments in §G. In §H we share the qualitative cross-modal retrieval results.

B. Related Work

Learning visual, acoustic, and textual representations has become central to building multimodal foundation models for perception. By aligning images, video, audio, and language in a shared embedding space, contrastive vision–language and audio–language encoders enable strong zero-shot performance across diverse benchmarks: zero-shot audio retrieval on AudioCaps [20, 71, 73], zero-shot image classification on ImageNet [16], and text-to-video retrieval [34, 57, 62] on MSR-VTT [78]. Furthermore, these encoders now serve as critical perception front-ends for multi-modal large language models (MLLMs) [4, 5, 47, 48, 55, 66, 79].

Vision-Language Representation Learning. Vision–language contrastive pretraining was established by early works such as Virtex [17], ICMLM [60], and ConVIRT [84], and later scaled up by CLIP [34, 57] and ALIGN [35] on significantly larger datasets and models.

Subsequently, a series of open-weight contrastive models [22, 42, 62, 65, 77, 83] have been developed to enhance CLIP's performance and robustness. Notably, SigLIP [83] replaces softmax with a sigmoid objective, and FLIP [44] employs masking to accelerate training. Additionally, researchers have explored incorporating auxiliary objectives, such as self-supervised losses [36, 51, 52] and captioning losses [67, 70, 81]. In the data part, a series of works [22, 23, 62, 77] have studied large-scale sourcing and filtering of web data. These efforts aim to boost model performance by scaling high-quality data through efficient data curation strategies. To further improve alignment and reduce noise in web-crawled data, several works [21, 41, 53, 76] explore re-captioning training images using MLLMs or VLMs. This strategy seeks to enhance text quality, thereby strengthening the robustness of the learned representations.

Recently, Perception Encoder (PE) [7] modernizes CLIP-style training and, with the Perception Language Model (PLM) [15] as a video data engine, scales image–video–language pretraining. Building upon PE and PLM, in this work, we further extend PE to build PE_{AV}, an audio-video-text encoder by incorporating the audio modality through model and data scaling with an audio-video data engine as described in §4.2 and §3 in the Main text.

Audio/Speech Representation Learning. Self-supervised learning (SSL) has become a dominant approach for audio representation learning, leveraging large amounts of unlabeled data. Notable models for speech representation include wav2vec 2.0 [2], HuBERT [30], and WavLM [9]. SSAST [26], Audio-MAE [31], data2vec [3], and BEATs [10] were developed for general audio. Moreover, MERT [45] and MuQ [86] have advanced music-domain audio representations. Recent advancements aim to learn audio representations at lower cost [12, 46] or across multiple domains within a single model [8]. However, these methods are limited to single-modality learning and do not use cross-modal information.

Recently, there has also been growing interest in leveraging paired audio–text data to better align audio and text modalities. Inspired by the success of CLIP in vision–language learning, CLAP [20] introduced a contrastive language–audio pre-training objective; subsequent work such as LAION-CLAP [73], M2D-CLAP [54], FLAP [80], and AF-CLAP [24] scaled this paradigm to more data, added SSL objectives, and incorporated syn-

thetic captions, yielding stronger and more transferable audio encoders (including for LLMs).

Toward Unified Audio–Video–Text Encoders. To move beyond audio-only or audio–text alignment, several works exploit the video modality to learn audiovisual representations. CAV-MAE [27] and MAViL [32] use video as complementary supervision and show strong results on classification and cross-modal retrieval. More recent "hub-style" approaches such as ImageBind [25], LanguageBind [85], and InternVid 2 [71] connect multiple modalities via a single anchor (image or language), but still suffer from scale mismatches between modality-pair datasets, which can hurt less-represented modalities, especially non-speech audio. In contrast, PE_{AV} focuses on large-scale, language-guided audiovisual representation learning by utilizing a robust audio-video data engine. This enables broader coverage of contrastive objectives, facilitating the learning of more robust audiovisual and text representations.

Sound Event Detection. Traditional sound event detection (SED) systems operate under a closed-vocabulary setting, targeting a predefined and limited set of sound classes, where each class is assigned a binary label at every time frame [50]. The performance of SED models has improved considerably on small-scale datasets centered on domestic environments [33, 63, 69]. More recently, self-supervised learning (SSL) and large-scale pretraining of audio spectrogram transformers have dramatically advanced SED capabilities, enabling the detection of diverse and complex acoustic scenes across hundreds of sound categories [43, 61]. These developments mark a significant shift from traditional, closed-vocabulary SED to flexible, open-vocabulary paradigms [28, 75], which aim to identify the temporal boundaries of any sound event described by natural language.

C. Audio-Video Data Engine

In the following, we provide details of the prompts and examples for the stage-1 and stage-2 pipelines used to generate audio, video, and audiovisual captions using the audio-video data engine.

C.1. Stage-1 Prompts

The stage-1 prompt used in the data engine is as follows. We leverage CoNeTTe [40] and ENCLAP [39], as well as an internal video captioner.

LLM Prompts for Visual Captions

Create primarily visual captions that focus on what can be seen in the video. Video captions are your reliable source -- **ALWAYS** create a caption from them, even if audio doesn't match. Handle repetitive video descriptions by summarizing while preserving all unique details. Audio can optionally enhance but should never drive the caption. Remember: (1) NEVER output an empty caption (2) ALWAYS create a caption from video content (3) If audio doesn't match, use only video details

Video Caption Principles

- **Primary Source:** ALWAYS use video captions
- **Detail Preservation:** Keep all distinct visual elements
- **Redundancy:** Clean up repetitive descriptions
- **Flow:** Create natural, coherent sentences

Required Output

- Video summary (clean, non-repetitive)
- Audio context (if used)
- Visual-focused caption between <BOS> and <EOS>
- Explanation of choices

Example 1

Input: INPUT FOR AUDIO-VISUAL:
- VIDEO CAPTIONS:
  - A professional surfer in a black wetsuit performs an aerial maneuver on a bright red surfboard against massive white waves.
  - An experienced surfer wearing dark gear rides along the crest of a towering ocean wave on their red board.
  - A skilled surfer executes a 360-degree turn while surfing on crystal clear blue waters.
- AUDIO CAPTIONS (WITH CONFIDENCE LABELS):
  - The thunderous crash of ocean waves fills the air. [confidence: HIGH_CONF]
  - Cat and dog meowing. [confidence: LOW_CONF]

Main Goal: Create a primarily visual caption that focuses on what can be seen in the video.

Video Caption Handling:
- Video captions are your primary and reliable source -- ALWAYS use them
- Preserve all distinct visual details (colors, actions, numbers, descriptions)
- If video captions are repetitive, summarize while keeping all unique details
- Combine multiple video perspectives into natural-flowing sentences

Audio Caption Handling (Optional):
- Audio is strictly optional -- visual details should stand alone
- Only consider HIGH_CONF audio that directly matches video content
- When using audio, add it at the end of the caption without disrupting visual flow

Output:
- Video summary: A professional surfer in a black wetsuit performs aerial maneuvers and a 360-degree turn on a bright red surfboard, riding along the crest of towering white waves in crystal clear blue waters.
- Audio summary: Wave sounds [HIGH_CONF] align with visible wave activity.
- Merged caption: <BOS> A professional surfer in a black wetsuit executes impressive aerial maneuvers and a 360-degree turn on their bright red surfboard, riding along the crest of towering white waves in crystal clear blue waters. <EOS>
- Explanation: Focused on rich visual details (wetsuit color, specific moves, board color, wave description). Though wave sounds matched, kept focus on visual elements.

LLM Prompts for Rewriting Audio Captions

Your task is to generate an **audio-focused caption** from model-generated video and audio captions for the same audio-visual input. All video captions are equally likely to be correct. Audio captions are generated using only audio, and different objects can produce similar sounds (e.g., machine low hum can be confused with crickets, lawn mower cutting grass may sound similar to engine whirring). Sometimes the video and audio may not correspond to each other as the recorded object may be far away. Consider if the described sounds are plausible given the video context. Each audio caption has a confidence label: LOW_CONF, MED_CONF, or HIGH_CONF: **LOW_CONF**: This caption is likely incorrect; when only LOW_CONF captions are present, use details that align with video. **MED_CONF**: At least one sound in the caption is correct; others may be incorrect. Only use details aligned with the video. **HIGH_CONF**: Generally reliable caption. Include all information with minor video-based adjustments. Prioritize over video caption if clear conflict. Your task is to merge the video and audio captions into a single caption, focusing on details determinable from audio. Summarize redundant captions before merging. If captions conflict, favor HIGH_CONF audio after verifying plausibility. When only LOW_CONF captions exist, try using common elements with video to create an audio-focused summary with general details.

Steps

1. Summarize the video captions.
2. Identify the audio captions relevant to the video scene and summarize them (dropping irrelevant ones). Verify HIGH_CONF details from given video context.
3. Merge the summarized captions into a single caption, focusing on plausible audio-based details, ignoring non-audio details like color.

Keep your answer succinct. Provide: (1) Summarized video caption.(2) Summarized audio caption. Include all plausible HIGH_CONF details. (3) A final merged caption enclosed between <BOS> and <EOS>.

Example 1

Input: INPUT FOR AUDIO-VISUAL:
- VIDEO CAPTIONS:
  - A person surfing in the clear blue ocean in front of the large white waves.
  - A tan man is surfing in the middle of the vast ocean and it starts to wave in the ocean.
  - A surfer on a bright red surfboard rides in a towering wave, then jumps up and starts surfing.
- AUDIO CAPTIONS (WITH CONFIDENCE LABELS):
  - The ocean waves can be heard. [confidence: HIGH_CONF]
  - Men talking. [confidence: HIGH_CONF]

YOUR TASK:
- Write an audio-focused summary caption.
- Summarize redundant or incomplete captions, focusing on details determinable from audio alone, such as gender, species, or car model, while ignoring vision-specific details like colors.
- If audio captions conflict, prioritize HIGH_CONF captions and include details, even if they seem to conflict with video. Make adjustments only when video clearly suggests a more plausible sound source, as the audio source may be off-camera or far away.
- Ignore LOW_CONF.
- Include MED_CONF details that seem plausible given the video.
- If no captions remain, create a short summary from the video. Output empty merged caption like <BOS><EOS> when caption cannot be created.

Output:
- Video summary: A tan man is surfing in the clear blue ocean with a bright red surfboard in front of large white waves.
- Audio summary: Sounds of waves and men speaking are heard.
- Merged caption: <BOS> A man is surfing in the ocean, riding large waves, accompanied by the sounds of waves and men speaking. <EOS>
- Explanation: Combining 'ocean waves' from first audio caption and 'men talking' from second caption to summarize using both HIGH_CONF. Added large waves and man details using video caption in the merged caption.

LLM Prompts for Rewriting Audio-Video Captions

You are tasked with generating **comprehensive audio-visual captions** that effectively combine information from both modalities. You will receive model-generated video captions and audio captions, each audio caption having a confidence label: LOW_CONF, MED_CONF, or HIGH_CONF.

Key Guidelines

- **Video captions are generally reliable** -- ALWAYS use video information.
- **Audio captions** are based on only audio, and different objects can produce similar sounds (e.g., lawn mower and car engine). For audio captions: (1) HIGH_CONF: Include if there's any plausible connection to the video context. (2) MED_CONF: Use only when clearly complementing video information. (3) LOW_CONF: Ignore these captions.
- When HIGH_CONF audio seems unrelated to video: (1) Include both video and audio information. (2) Add note that they might not correspond.
- Look for creative ways to interpret audio captions in the video context using generic terms.
- Be generous in finding plausible connections between modalities by using broader categories. For example: (1) If audio mentions specific vehicles (car/truck) and video shows any vehicle -- use generic terms like vehicle engine/sounds. (2) If audio describes water sounds and video shows any liquid -- include it. (3) If audio mentions speaking/talking and video shows people -- connect them.

For each input, provide:

- **Video summary:** Key visual elements and actions.
- **Audio summary:** Relevant sounds and speech from HIGH_CONF captions.
- **Merged caption:** Natural combination of both modalities between <BOS> and <EOS>.
- **Brief explanation** of your integration choices.

Remember: (1) NEVER output an empty caption. (2) ALWAYS include video information. (3) Be generous in finding plausible audio-video connections. (4) When using unrelated HIGH_CONF audio, note the potential mismatch.

Example 1

INPUT FOR AUDIO-VISUAL CAPTIONING:
- VIDEO CAPTIONS:
  - A drain on a tiled floor shows signs of clogging with debris.
  - The metallic drain cover is partially lifted, revealing dirty water inside.
  - The surrounding tiles appear stained and poorly maintained.
- AUDIO CAPTIONS (WITH CONFIDENCE LABELS):
  - Water is splashing with a scraping sound. [confidence: HIGH_CONF]
  - People are talking in a foreign language. [confidence: HIGH_CONF]

YOUR TASK:
- Always create a comprehensive caption using the video information as primary source.
- For audio captions:
  - Use HIGH_CONF audio captions that seem plausible given the video context.
  - Include MED_CONF details only when they clearly complement video information.
  - Ignore LOW_CONF audio captions.
- When video and HIGH_CONF audio seem unrelated:
  - Include both video and audio information.
  - Note that they might not correspond to each other.
- Never output an empty caption -- always use at least the video information.
- Be generous in finding plausible connections between audio and video.

Output:
- Video summary: A clogged metallic drain on a tiled floor shows debris and dirty water with a partially lifted cover.
- Audio summary: Sounds of water splashing and scraping are heard.
- Merged caption: <BOS> A clogged drain on a tiled floor contains dirty water, with splashing and scraping sounds as the water moves through the debris. <EOS>
- Explanation: Combined visual details of the drain with relevant HIGH_CONF audio of water and scraping sounds, ignoring unrelated audio about people talking.

C.2. Stage-1: Synthetic captioning examples

Fig. 1 shows the captions for EnCLAP, CoNeTTE, and our internal video captioning model. The LLM examples are shown in the *LLM rewrites* block of Fig. 1.

C.3. PLM-AV Training Details

In the warm-up phase we only train a MLP layer on 4M synthetic captions to project the audio-visual / audio embeddings to the same dimensions as LLM embeddings. In the mid-training phase, we fine-tune the entire model on a 30M synthetic captions previously described. In the final stage, we fine-tune on a curated mix of data focused on audio-visual QA, captioning, instrument recognition, sound tagging, and paralinguistic attributes. Our key goal is to have a model that can produce an audio caption in natural language, or produce a list of sound events in Noun-Verb format. We additionally focus on improving the understanding of the acoustic environment. In Tab. 1, we show the performance of the PLM-AV models when measured on held-out Audiocaps [38] and Clotho-V2 [18] datasets. We use CLAP score from LAION [73] model to measure the quality of audio captions. We find that even in out-of-domain settings the model produces CLAP scores in the same ballpark as ground-truth captions.

Dataset	Ground Truth	PLM-AV (NPVP)	PLM-AV (caption)
Audiocaps	0.52	0.54	0.46
Clotho-V2	0.57	0.34	0.54

Table 1. Performance comparison against ground-truth using CLAP scores from LAION model on AudioCaps and Clotho-V2 datasets. We find that PLM-AV produces high quality tags and captions even on the out-of-domain datasets.

In addition to improving the audio captions, we include speech-related attributes–transcript, language ID (LID), and accent–to enhance PE_{AV}'s capability in speech processing. We use Whisper Large-v3 and Medium ASR models [58] to transcribe training audio clips, and keep English-only transcripts and transcripts that the two models agree upon (low word error rate). Similarly, we create LID labels with MMS LID models with 126 and 256 languages [56], and keep the labels where the two models share the top-1 prediction. For accent labels, we first train an English accent classifier with Common Voice 13 [1], and use this classifier to produce accent labels for audio clips with English LID. During training, we randomly insert transcript, LID, and accent information into the audio caption. We find that assigning a subset of the data with transcripts and always replacing the original audio captions with the corresponding transcripts helps improve transcript retrieval tasks without compromising performance on other tasks.

C.4. Stage-2 Prompts

Stage-2: PLM Prompts for Fine-Grained Video Caption

Create a fine-grained caption of a video using the provided metadata (if applicable), video caption, and frame captions.
Task: Extract key information from the captions and combine it into an alt text format using single phrase or set of phrases that includes all relevant details.
Steps to Follow:
- Review the metadata if (title and description) for general context, you can rely it for entity names but do not rely on it as the primary source of information for your caption.
- Blend title / description with video caption and frame captions for the main storyline
- Extract the most relevant and concise information.
- Combine extracted information into a alt text format using short phrase or set of phrases with approximately 120 tokens, considering special characters like comma as part of the token count.
- Prioritize including all key information over sentence structure or grammar.
- Minimize the use of special characters and focus of key information.

What to Avoid:
- Avoid adding or inferring information not present in the original metadata and captions.
- Avoid using complex sentence structures or prioritizing sentence flow.

Create a concise caption with the full details of the video based on the metadata, video caption, and frame captions.

Stage-2: Final LLM Summarization Prompts for Stage-2 Video Captions

Task:
You are provided with two types of captions for the same video:
1. **Video-level captions:** A high-level summary of the entire video.
2. **Fine-grained captions:** Detailed descriptions of specific events within the video.

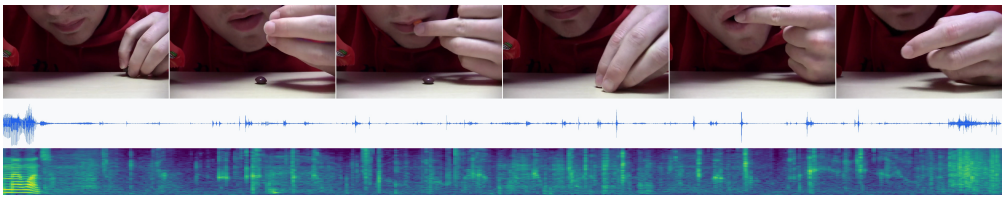
Goal:
Write a single, concise, and coherent summary that:
- Clearly captures the main events of the video.
- Preserves important actions, objects, and contextual information from both caption types.
- Avoids unnecessary repetition of frame-by-frame details.
- Resolves any inconsistencies by prioritizing the video-level caption, unless the fine-grained caption adds essential information.
- Is written as a single sentence, not exceeding 72 words.

Input Format:
- Video-level captions: <stage1_vcap>
- Fine-grained captions: <plm_vcap>

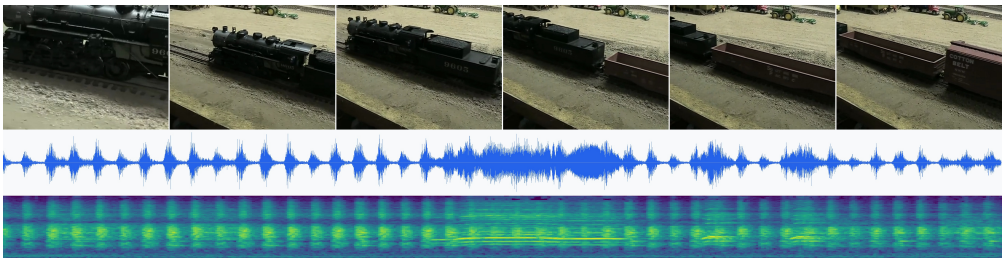
Output:
A single-sentence summary describing the video.

D. $\text{PE}_{\text{A-Frame}}$: Audio-Frame Level Language Alignment

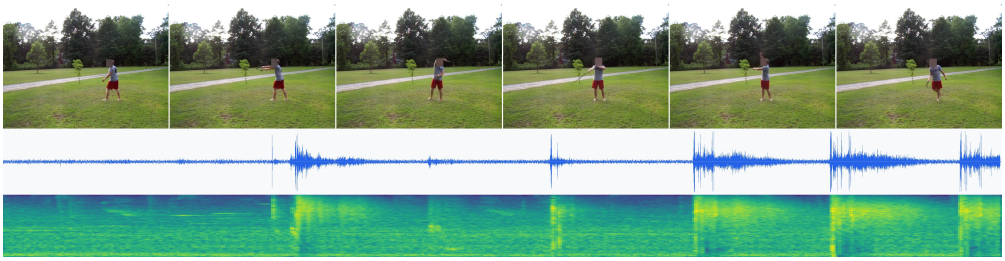
Typical language–audio models produce a single utterance-level embedding (a global token) for each modality. Then, they apply a contrastive loss to the global class tokens, which achieves coarse-grained cross-modal alignment but overlooks fine-grained interactions. As a result, the correspondence between audio at the frame-level and language remains underexplored, leading to low performance on tasks requiring detailed temporal alignment. To address this limitation, we propose *Perception Encoder Audio-Frame* ($\text{PE}_{\text{A-Frame}}$), a model fine-tuned from PE_{AV} with a frame-level audio to language contrastive loss.



EnCLAP: a man speaks followed by some crinkling and crumpling (score: 0.48)
CoNeTTE: a man is speaking and chewing on something (score: 0.57)
Video: The video shows a close-up shot of a man eating a small, round candy. In the background, he sits at a white table. The man has fair skin, brown hair and a red shirt. He sits at the table with his hands resting on the edge. He holds an orange candy in his hand to the right. The candy has a blue wrapper with white writing on it. The man puts the candy in his mouth. He chews the candy and swallows it. He then picks up a brown candy from the table. He puts the brown candy in his mouth. He chews the candy and swallows it. He then picks up another orange candy from the table. He puts the orange candy in his mouth. He chews the candy and swallows it. He then picks up another brown candy from the table. He puts the brown candy in his mouth. He chews the candy and swallows it. He then picks up another orange candy from the table. He puts the orange candy in his mouth. He chews the candy and swallows it. The video ends with the man chewing the candy.



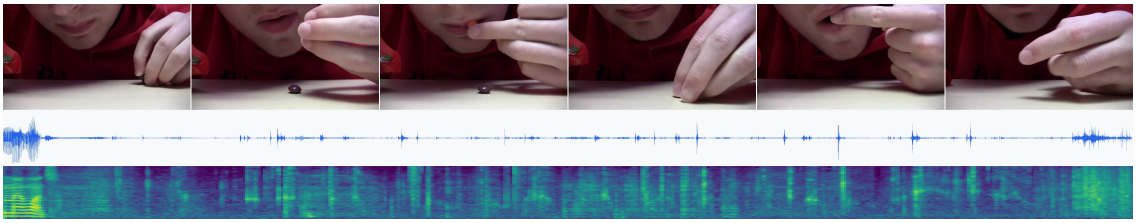
EnCLAP: a train is chugging along and a steam whistle goes off (score: 0.50)
CoNeTTE: a person is using a wrench on a typewriter (score: -0.08)
Video: The video shows a train traveling on a track. The train has a large, black engine with a black caboose behind it. The caboose has white lettering on the side that reads "COTTON BELT" and "9605." The train is traveling on a track that runs along the bottom of the frame. The background shows a sandy area with several pieces of farm equipment. The video begins with the train traveling toward the left side of the frame. As the video progresses, the train continues to travel along the track. The video ends with the train moving toward the left side of the frame.



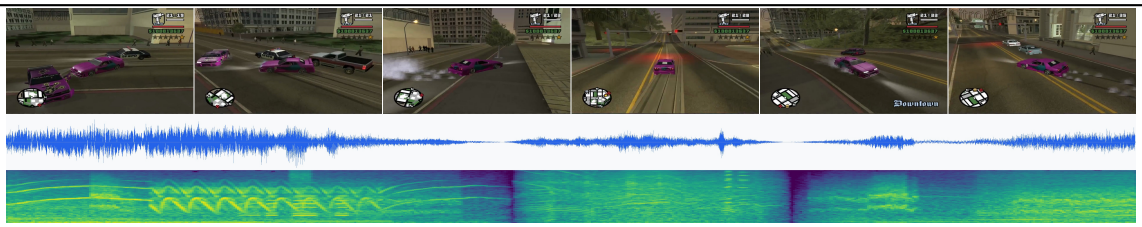
EnCLAP: a toy helicopter is flying around and making a lot of noise (score: -0.02)
CoNeTTE: a whip is being struck in the distance while birds are chirping in the background (score: 0.43)
Video: The video shows a young man playing with a whip in a grassy field. In the background, there is a gravel path and a line of trees with a house behind them. The man has fair skin and short brown hair. He is wearing a gray t-shirt and red shorts. The whip is long and thin. At the beginning, the man faces the right side of the video while holding the whip in his hand on the left side of the video. He swings the whip over his head and releases it. It whips around the man's body, then moves back toward the camera. The man moves toward the camera as the whip approaches. He catches the whip with his hand on the right side of the video and swings it over his head. The whip whips around the man's body, then moves back toward the camera. The man catches the whip with his hand on the left side of the video. The video is shot by a handheld camera.

Figure 1. EnCLAP and CoNeTTE captions often provide complementary information and the confidence scores reflect the accuracy reasonably, making them favorable to combine with an LLM. Video captions provide strong context. Together this provides strong audio and visual cues for LLM rewriting.

Training. We train $\text{PE}_{\text{A-Frame}}$ to predict the specific frames within an audio signal x^a that contain the sound described by the free-form text description x^t. Building on the pre-trained PE_{AV} model, we fine-tune frame-level audio and instance-level text encoders using contrastive learning, constructing a joint embedding space where similar audio-frame and text pairs are close, and dissimilar pairs are distant. The audio encoder outputs a sequence of L^a frame-



Type	Caption
Audio	A man is speaking and eating candies, accompanied by the sounds of crinkling and crumpling wrappers.
Video	A man with fair skin, brown hair, and a red shirt eats various candies at a white table, alternating between orange and brown candies with blue and white wrappers.
Audio–Visual	The video shows a man eating various candies at a table, with sounds of him speaking and chewing. He picks up and consumes multiple orange and brown candies with blue and white wrappers, accompanied by some sounds of crinkling and crumpling.



Type	Caption
Audio	A police vehicle is moving, triggering a siren, as cars drive by on the city streets.
Video	A purple car with blue and black design drives through the city, passing a black and white police car with flashing red and blue lights, a black and red car, and a white and black car, with the city's buildings, road, and pedestrians in the background.
Audio–Visual	The video shows a purple car racing through the streets of a city, passing by various other cars, including a police car with flashing red and blue lights. The audio captures the sound of a police siren being triggered and a vehicle moving.

Figure 2. Examples of Audio, Video, and Audio–Visual captions generated using the data engine.

level representations, where the sequence length L^a depends on the length of the audio and the feature rate of the audio encoder (here 25 Hz). We adopt a frame-level variant of the sigmoid contrastive loss [83]. For each frame l, we compute the logit between the frame-level audio embedding $\mathbf{e}_l^a$ and the global text embedding $\mathbf{h}^{t_a}$ as $\tilde{h}_l = \mathbf{e}_l^a \cdot \mathbf{h}^{t_a}$.

Input Data and Ground-Truth Label. Each element in a batch of size B consists of an audio sample x_b^a and a single sound event described by its associated text description x_b^t. Although an audio clip may contain multiple sound events, we sample only one text description per element in each batch to simplify implementation. Nevertheless, we provide all annotated sound events and their corresponding frame-level activity masks m for every audio sample. Accordingly, each batch element contains

- an audio file x_b^a,
- a sampled text query x_b^t describing one sound event in x_b^a, and
- given the other batch elements can contain audio events for the input audio x_b^a, we pass an event-activity mask m_b that captures all the events for the b-th audio input. For the b-th audio with K_b events, we define $m_b \in \{0,1\}^{L_a \times K_b}$ over all annotated events $\{x_{b,1}^t, \ldots, x_{b,K_b}^t\}$ in x_b^a, where $m_{b,l,k} = 1$ indicates that $x_{b,k}^t$ is active at frame l.

Even though only one text query per audio is used for contrastive learning, we leverage all annotated events and their *ontology-aware*[1] semantic expansions to construct frame-level supervision.

Let $\mathrm{Ont}(x^t)$ be the set of ontology-linked variants of event x^t (e.g., "speech" includes "female speech", and "dog" includes "barking"). Then, for each audio frame l in x_b^a and each text query $x_{b'}^t$ in the batch, we assign:

$$z_{b,l,b'} = \begin{cases} +1, & \exists\, k \in \{1,\ldots,K_b\} :\ x_{b'}^t \in \mathrm{Ont}(x_{b,k}^t) \\ & \text{and } m_{b,l,k} = 1, \\ -1, & \text{otherwise.} \end{cases} \tag{1}$$

Thus, semantically equivalent sound expressions activate the same frames in supervision, improving robustness to linguistic variation and reinforcing hierarchical concept generalization.

[1] By "ontology" we mean a hierarchical taxonomy of sound event labels, such as the AudioSet ontology.

Frame-Level Objective. For this task, the model must learn not only *which* sound events are present in an audio clip but also *when* they occur over time. To capture both aspects, we employ two complementary objectives: a *local-activity loss* (computed per batch item) that emphasizes fine-grained temporal localization, identifying *when* events occur within each audio sample, and a *global-activity loss* (computed across the batch) that introduces contrastive context between samples to determine *which* events are active. The local-activity loss helps the model detect event boundaries, while the global-activity loss promotes global event understanding and cross-sample alignment.

The resulting local-activity loss (per-batch-item) yields logits of shape (B, L), while the global-activity loss (across-batch) yields (B, L, B):

$$\tilde{h}_{b,l(,b')} = \begin{cases} \mathbf{e}_l^a(x_b) \cdot \mathbf{h}^{t_a}(y_b) & \text{(local-activity)} \\ \mathbf{e}_l^a(x_b) \cdot \mathbf{h}^{t_a}(y_{b'}) \; \forall\, b' \in \{1, \ldots, B\} & \\ & \text{(global-activity)}, \end{cases} \tag{2}$$

A learnable logit scale α and bias β are applied to obtain the final scaled logits $h = \alpha\tilde{h} + \beta$. The frame-level SigLIP loss is computed over these logits. For the local-activity case, the loss is defined as

$$\mathcal{L} = -\frac{1}{BL_a} \sum_{b,l} \log \sigma\big(z_{b,l}\big(\underbrace{\alpha\, \tilde{h}_{b,l} + \beta}_{h_{b,l}}\big)\big), \tag{3}$$

where $z_{b,l,b} \in \{\pm 1\}$ is the binary label indicating whether frame l of audio x_b corresponds to the paired text y_b. This formulation naturally generalizes to the global-activity case by computing the loss over $\tilde{h}_{b,l,b'}$ and averaging across all text queries $y_{b'}$ in the batch. During training, we probabilistically sample between the two objectives at each iteration, with the probability p_{local} for the local-activity loss. This stochastic choice allows the model to balance precise event-boundary detection and global event activity modeling.

Training Data. For training, we use a combination of real-world audio mixtures annotated by humans and synthetic mixtures automatically generated from diverse isolated audio sources. To enhance robustness to reverberation and spatial variability, the audio mixtures are convolved with room impulse responses collected from a variety of acoustic environments. This process allows the model to better generalize across different recording conditions and scene types.

Table 2 provides a summary of the training data, reporting the total durations and number of sound events for both real and synthetic recordings. The *Speech* and *Music* subsets correspond to data explicitly annotated with these categories, whereas the *General* subset comprises a broader range of sounds, which may also include speech or music instances not specifically labeled as such. We use dataset-specific sampling ratios to ensure a balanced mixture of sound events, avoid over-representation, and maintain comprehensive coverage.

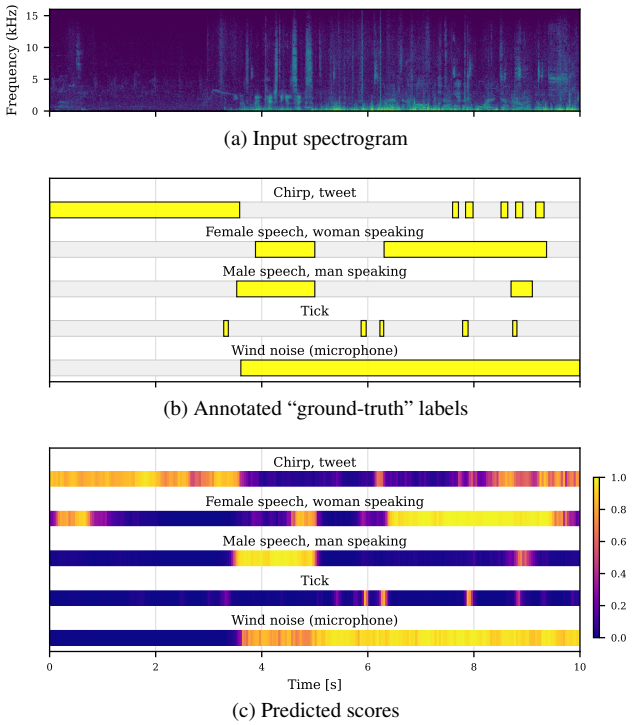
	Duration [hours]		**Sound Events**	
Type	**Real**	**Synthetic**	**Real**	**Synthetic**
Speech	0.4 k	0.3 k	70.3 k	108.7 k
Music	0.2 k	0.2 k	0.4 M	0.4 M
General	0.6 k	12.3 k	0.8 M	4.1 M
Total	1.2 k	12.8 k	1.3 M	4.6 M

Table 2. **$\text{PE}_{\text{A-Frame}}$ Training Data**. Durations and sound event counts for real and synthetic recordings across three sound-type categories.

E. Downstream Application: Sound Event Detection (SED)

We evaluate $\text{PE}_{\text{A-Frame}}$ on the task of polyphonic sound event detection SED, which is defined as the detection of sound events from multiple classes, where sound events can occur simultaneously [49]. Traditional (closed-vocabulary) SED typically targets a fixed set of classes, predicting a binary label for each class per time frame. In contrast, open-vocabulary SED aims to detect the temporal boundaries of arbitrary sound events conditioned on a free-form textual description [28, 75]. Our model is designed to address both closed- and open-vocabulary SED, supporting free-form textual queries for arbitrary sound events. For closed-vocabulary evaluation, it is prompted with each class from the predefined ontology, and detection proceeds as in the traditional setting. For open-vocabulary evaluation, we assume access to a free-form textual description of the sound events present in the audio, and the model predicts the precise onset and offset boundaries of every instance of the specified events.

Test sets and metrics. To evaluate performance, we employ both open-vocabulary (Internal Bench, ASFX-SED [74]) and closed-vocabulary test sets (AudioSet-Strong [29], DESED [69]—a.k.a. "Youtube" subset in DCASE19 [68], and UrbanSED [59]), where Internal Bench denotes an internal benchmark. We assess model performance using two threshold-independent metrics: the intersection-based polyphonic sound detection score (PSDS) [6] and the segment-based area under the receiver operating characteristic (AUROC). For open-vocabulary SED datasets, AUROC is computed only over the true positive and false positive rates of events that actually occur in each audio clip. For closed-vocabulary test



(a) Input spectrogram

(b) Annotated "ground-truth" labels

(c) Predicted scores

Figure 3. **Sound event detection example using PE$_{\text{A-Frame}}$.** The model successfully detects all sound events, accurately distinguishing between male and female speech, and capturing short transient events such as "Tick" enabled by its high temporal resolution (25 Hz).

sets, predictions are generated for all classes included in the respective datasets. We apply a median filter of 9 to the raw predictions and use the `sed_scores_eval` package [19] with parameters $\rho_{\text{DTC}} = 0.7$, $\rho_{\text{GTC}} = 0.7$, $\alpha_{\text{ST}} = 1$, $\alpha_{\text{CT}} = 0$, and $e_{\text{max}} = 100$, corresponding to the standard PSDS1 configuration [6]. However, consistent with [43, 61], we omit the variance penalty ($\alpha_{\text{ST}} = 0$) for AudioSet-Strong, as PSDS was originally designed for datasets with fewer and less imbalanced classes [6]. Following [28], we refer to PSDS1 computed across all classes as PSDS1$_{\text{A}}$, which emphasizes accurate temporal alignment but still penalizes false positives, and adopt its variant, PSDS1$_{\text{T}}$, which focuses solely on target sounds.

Baselines. We compare our model with PretrainedSED [61], FLAM [75], and FlexSED [28]. PretrainedSED, for which we use the best-performing checkpoint based on the BEATs transformer [11], is not freeform, as it includes a final task-specific layer that outputs probabilities for the Audioset-Strong classes. Its training pipeline employs a balanced sampler, extensive data augmentation, and ensemble knowledge distillation, making it highly optimized for the Audioset-Strong test set. In contrast, FLAM

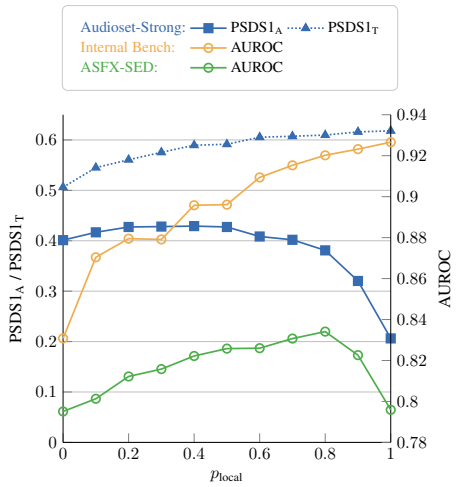


Figure 4. Ablation on the probability p_{local} controlling the ratio between local-activity and global-activity loss calculations, conducted with the base model trained for 10 k steps.

and FlexSED accept free-form textual descriptions and are therefore also suitable for open-vocabulary SED.

Results. Table 3 presents the results across the different SED test sets, grouped into open-vocabulary and closed-vocabulary categories. Overall, PE$_{\text{A-Frame}}$ demonstrates strong performance, achieving the highest scores on all test sets in PSDS1$_{\text{T}}$, indicating superior capability in accurately detecting temporal boundaries of target sounds. In particular, PE$_{\text{A-Frame}}$ attains the best performance on DESED across all metrics, highlighting its robustness in real-world acoustic environments, as DESED comprises real recordings with fine-grained human annotations [69]. As expected, PretrainedSED performs well on AudioSet-Strong, as it is optimized for that specific ontology; however, its closed-vocabulary nature prevents its application to the other test sets. Finally, FLAM performs best on UrbanSED, which is a synthetic and relatively unrealistic dataset. We hypothesize that this is because, unlike our model, their system was also trained on UrbanSED, giving it an inherent advantage on this benchmark. Moreover, FLAM operates with a coarser frame rate of 3.2 Hz, which limits its ability to perform fine-grained temporal detection and may lead to lower performance on intersection-based metrics.

E.1. Ablation studies for PE$_{\text{A-Frame}}$

Figure 4 presents an ablation study on the sampling probability p_{local}, which controls the ratio between the local-activity and the global-activity objective (see §D), conducted with the base model trained for 10 k steps. We observe that higher values of p_{local} lead to improved PSDS$_{\text{T}}$ scores, emphasizing that the local-activity loss benefits the detection of target sounds under the intersection-based eval-

	Freeform	Rate [Hz]	Open-vocabulary SED		Closed-vocabulary SED								
			Internal Bench	*ASFX-SED*	*AudioSet-Strong* (407 classes)			*DESED* (10 classes)			*UrbanSED* (10 classes)		
			AUROC	AUROC	$PSDS1_A$	$PSDS1_T$	AUROC	$PSDS1_A$	$PSDS1_T$	AUROC	$PSDS1_A$	$PSDS1_T$	AUROC
PretrainedSED [61]	✗	25	-	-	**0.47**	0.52	**0.98**	-	-	-	-	-	
FLAM [75]	✓	3.2	-	0.81	0.35	-	0.95	0.09	-	0.92	**0.30**	-	**0.94**
FlexSED [28]	✓	25	0.62	0.74	0.45	0.58	0.96	0.16	0.27	0.93	0.05	0.11	0.71
$PE_{A\text{-}Frame}$	✓	25	**0.91**	**0.83**	0.43	**0.61**	0.96	**0.34**	**0.58**	**0.97**	0.12	**0.22**	0.89

Table 3. **Sound Event Detection Results.** Performance of $PE_{A\text{-}Frame}$ on open-vocabulary (Internal Bench, ASFX-SED) and closed-vocabulary (AudioSet-Strong, DESED, UrbanSED) SED test sets. $PE_{A\text{-}Frame}$ achieves the best $PSDS1_T$ across all benchmarks, indicating superior temporal localization. PretrainedSED is strong but limited to AudioSet-Strong due to its closed-vocabulary design, while FLAM mainly excels on UrbanSED, likely because it is trained on this synthetic dataset and operates at a coarser 3.2 Hz frame rate.

	Open-vocabulary SED		Closed-vocabulary SED								
	Internal Bench	*ASFX-SED*	*AudioSet-Strong* (407 classes)			*DESED* (10 classes)			*UrbanSED* (10 classes)		
	AUROC	AUROC	$PSDS1_A$	$PSDS1_T$	AUROC	$PSDS1_A$	$PSDS1_T$	AUROC	$PSDS1_A$	$PSDS1_T$	AUROC
$PE_{A\text{-}Frame}$L	0.91	0.83	0.43	0.61	0.96	0.34	0.58	0.97	0.12	0.22	0.89
$PE_{A\text{-}Frame}$B	0.92	0.83	0.42	0.60	0.96	0.39	0.56	0.98	0.12	0.21	0.89
$PE_{A\text{-}Frame}$S	0.91	0.83	0.39	0.59	0.96	0.32	0.54	0.96	0.09	0.19	0.88
$PE_{A\text{-}Frame}$B (from scratch)	0.89	0.76	0.22	0.55	0.89	0.10	0.52	0.89	0.01	0.08	0.82

Table 4. Ablation results for model sizes small (S), base (B), large (L), and a base model trained from scratch. Larger models give only slight gains, while training from scratch leads to a substantial drop, highlighting the importance of large-scale pretraining.

uation metric. However, this comes at the cost of reduced $PSDS_A$, which penalizes false positives that may arise because the model contrasts fewer non-target sound events and instead focuses more narrowly on local alignment. Furthermore, we observe a monotonic increase in AUROC on the internal benchmark and an increase up to $p_{\text{local}} = 0.8$ on the ASFX-SED benchmark, followed by a drop thereafter. A value of $p_{\text{local}} = 0.7$ provides a favorable trade-off between these metrics, and we therefore adopt it as the default setting in all subsequent experiments.

An optimal value of p_{local} ultimately depends on the intended application of the SED system. If the goal is to precisely detect sound event boundaries for a given set of target events, it is recommended to use a higher p_{local} value. Conversely, if the application prioritizes minimizing false positives, such as in continuous environmental monitoring or safety-critical detection scenarios (e.g., false alarms in smart home or surveillance systems), a lower p_{local} may be more favorable.

Table 4 reports ablation results for different model sizes (large, base, and small), and the base model trained from scratch without pretrained weights. The results show that larger models yield only a slight improvement in performance metrics. However, there is a notable drop when the model is trained from scratch, highlighting the importance of large-scale pretraining for achieving strong performance.

F. Implementation Details

F.1. Architecture and Training Setups

Model Architecture. We provide the detailed parameters of PE_{AV} in Tab. 5. For the frame encoder, we utilize the pre-trained PE-L [7] ($\sim$320M parameters) as the base frame encoder to capture spatial context, and employ a video encoder on top of it to encode temporal context. By default video frames are sampled under 30 frames-per-second (fps) from up to 30 seconds videos. Each frame is transformed into 336×336 resolution and then encoded into one *CLS* token via PE-L. To capture temporal context across frames, we stack 4 additional shallow Transformer layers (30M$\sim$180M parameters) as the video encoder on top of the frame encoder outputs. Note that we choose to freeze the frame encoder in PE_{AV} to ensure comparable image-only performance as in PE-L.

For the audio modality, we pre-train a DAC-VAE with in-house audio data. We use it to encode raw audio into 25×128-dimensional vectors for a 1-second audio clip (and 750×128 for a 30-second audio), which serve as input to PE_{AV}. For the random-projection quantization module in BEST-RQ, we first project DAC-VAE features to a 16-dimensional latent space and quantize these features with four codebooks, each consisting of 16384 codewords. The base audio encoder, PE_{AV}B, is composed of 16 Transformer layers with around 0.21B parameters, while the large variant, PE_{AV}L, contains 28 layers and 1.11B parameters. The small model PE_{AV}S contains 12 layers and 0.09B parameters. We scale the hidden dimension proportionally to the number of layers with a factor of 64, and adjust the number of heads with a factor of 0.5.

To integrate audio and video features, we interpolate the video and audio token sequences to the same sequence length for alignment, then concatenate them along the channel dimension. This combined representation is fed into

the audio-visual fusion module, which comprises 6 Transformer layers designed to incorporate both audio and video context within the video. The hidden dimension of the fusion tower also scales with the number of layers in the audio encoder, using a factor of 64. For the text encoder, to extend our support to transcript data which requires long context length, instead of using paired text encoder in PE-L, we use pre-trained ModernBERT with 28 layers to accommodate input texts up to 512 tokens. Based on early experiments, we use the 22nd-layer output, and we keep the text tower unfrozen during training.

We employ customized Transformer configurations as detailed in Tab. 5. For pooling, we add an attention pooling block in the last-layer of video, audio, and audio-video Transformer. Regarding positional embedding, we use 2D RoPE [64] for relative positional embeddings. We additionally add a 2D learnable absolute positional embeddings (abs) the same size as the model's input resolution for the frame encoder. Finally, for simplicity, we use an input mean and standard deviation of $(0.5, 0.5, 0.5)$.

F.2. Efficient Scaling of Contrastive Pairs

Fig. 5 sketches our efficient implementation for contrastive loss scaling. The default strategy performs two `all_gather` operations for every loss pair, so with P pairs the step performs $2P$ collectives. As node count grows, the all gather operations dominate runtime.

In our efficient implementation, we *reduce the number of all gather calls to two* irrespective of the number of loss pairs involved. We stack the first and second arguments of all P pairs along the batch dimension. We then issue a single `all_gather` over the stacked tensors to collect all modalities, and then split the result using recorded batch sizes before evaluating each loss.

Batch-wise concat/split is far cheaper than multiple cross-node collectives, yielding fewer synchronizations and better bandwidth utilization; in practice (4 pairs, 8 nodes) this reduces step time by approximately 40–50%.

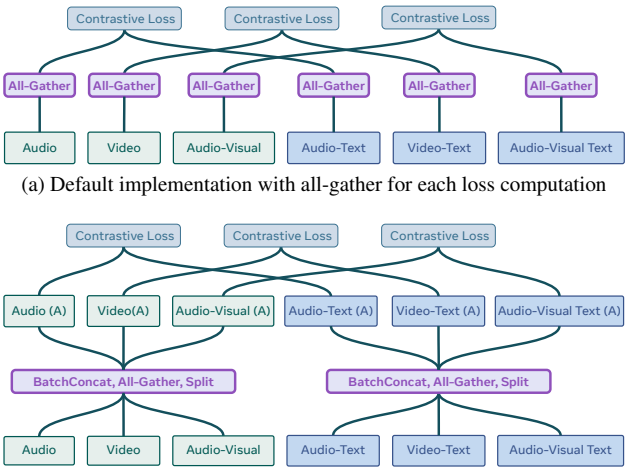
Training Recipe As discussed in §2 in the Main text, the training of PE_{AV} involves two stages: 1) audio-video pre-training; 2) video and speech fine-tuning. These two stages work together to develop a robust and effective PE_{AV} model.

We first provide the complete training recipes for 1) audio-video pre-training in Tab. 6 and 2) video and speech fine-tuning in Tab. 7.

F.3. Zero-Shot Classification and Retrieval

Zero-Shot Evaluation on Images and Videos. We use CLIPBench[2] for zero-shot classification and retrieval benchmarking. The benchmark datasets and splits are obtained from the original dataset websites or HuggingFace.

[2] https://github.com/LAION-AI/CLIP_benchmark



(a) Default implementation with all-gather for each loss computation

(b) Efficient implementation with fewer all-gather

Figure 5. **Efficient Implementation for SigLIP scaling** *Top:* Naïve computation involves **two** `all_gather` calls per loss pair which makes it hard to scale as the number of loss terms increase. *Bottom:* Our approach concatenates the first terms across all pairs along the batch axis (and likewise for the second terms), performs a **single** `all_gather` independent of the number of pairs, then splits by batch sizes before computing losses. This reduces collectives and improves throughput; in our setup (4 pairs, 8 nodes) we observe ∼40–50% speedup.

We extend zero-shot classification and retrieval in CLIP-Bench to include additional audio and video datasets such as AudioCaps, MSR-VTT, and Kinetics. We release our model checkpoints, evaluation code, and scripts for reproducibility.

Prompt Design. For zero-shot video-text retrieval, we rely solely on the original captions without any additional prompts. In contrast, for zero-shot classification, we utilize task-specific prompts graciously provided by the InternVL [13] authors. All additional dataset-specific prompts are released for reproducibility. For example, we employ specific prompts for zero-shot video classification on Kinetics datasets (e.g., K400, K600, K700).

Zero-Shot Video Classification Prompts - Kinetics

```
a photo of {c}. a photo of a person {c}. a photo of a person using {c}. a photo of a person doing {c}. a photo of a person during {c}. a photo of a person performing {c}. a photo of a person practicing {c}. a video of {c}. a video of a person {c}. a video of a person using {c}. a video of a person doing {c}. a video of a person during {c}. a video of a person performing {c}. a video of a person practicing {c}. a example of {c}. a example of a person {c}. a example of a person using {c}. a example of a person doing {c}. a example of a person during {c}. a example of a person performing {c}. a example of a person practicing {c}. a demonstration of {c}. a demonstration of a person {c}. a demonstration of a person using {c}. a demonstration of a person doing {c}. a demonstration of a person during {c}. a demonstration of a person performing {c}. a demonstration of a person practicing {c}.
```

Scale	Tower	Params	Width	Depth	MLP	Heads	CLIP Dim	Pooling	Positional Embedding	Resolution & Context Len	Patch Size
S	Audio	0.09B	768	12	2048	6	1024	Attn Pool	RoPE	-	-
	Frame (PE-L)	0.32B	1024	24	4096	16	-	Attn Pool	RoPE+Abs	336	14
	Vision (Temporal)	0.03B	768	4	2048	6	1024	Attn Pool	RoPE	-	-
	Audio-Video	0.05B	768	6	2048	6	1024	Attn Pool	RoPE	-	-
	Text	0.39B	1024	28	5248	16	1024	First Token	RoPE	512	-
B	Audio	0.21B	1024	16	2752	8	1024	Attn Pool	RoPE	-	-
	Frame (PE-L)	0.32B	1024	24	4096	16	-	Attn Pool	RoPE+Abs	336	14
	Vision (Temporal)	0.06B	1024	4	2752	8	1024	Attn Pool	RoPE	-	-
	Audio-Video	0.08B	1024	6	2752	8	1024	Attn Pool	RoPE	-	-
	Text	0.39B	1024	28	5248	16	1024	First Token	RoPE	512	-
L	Audio	1.11B	1792	28	4800	14	1024	Attn Pool	RoPE	-	-
	Frame (PE-L)	0.32B	1024	24	4096	16	-	Attn Pool	RoPE+Abs	336	14
	Vision (Temporal)	0.18B	1792	4	4800	14	1024	Attn Pool	RoPE	-	-
	Audio-Video	0.25B	1792	6	4800	14	1024	Attn Pool	RoPE	-	-
	Text	0.39B	1024	28	5248	16	1024	First Token	RoPE	512	-

Table 5. PE_{AV} Model Configurations.

config	values
optimizer	AdamW
β_1, β_2	(0.9, 0.999)
weight decay	0.0
learning rate	1e-4
batch size	3024
warm-up steps	500
training steps	250K
data quantity	92M
samples seen	750M

Table 6. **Detailed Pre-training Setup.**

config	values
optimizer	AdamW
β_1, β_2	(0.9, 0.999)
weight decay	0.0
learning rate	1e-4
batch size	1344
warm-up steps	500
training steps	50K
data quantity	32M(stage2) + 92M(stage1)
samples seen	67M

Table 7. **Detailed Fine-tuning Setup.**

config	values
optimizer	AdamW
β_1, β_2	(0.9, 0.999)
weight decay	0.0
learning rate	1e-4
batch size	800
warm-up steps	500
training steps	100K
data quantity	92M
samples seen	80M

Table 8. **Detailed Ablation Setup.**

Evaluation Method. For all the zero-shot evaluation, we follow [13] in using *retrieval reweighting* (DSL) to apply normalization over the softmax score distribution to the similarities used for retrieval:

```
sims = sims * softmax(sims, dim=0)   (4)
```

This slightly improves retrieval for most models, so we do it for all models we evaluate for fairness. Notably, we were able to reproduce the reported numbers for most papers with these techniques, but for cases where we could not, we default to the reported number.

For all retrieval tasks, we use dual-softmax [14] for both our models and other baselines. Empirically, we find that sharpening the final logits by a factor of 10 improves downstream performance. This adjustment aligns with the intuition that the model was trained with a scaled logit space to classify paired samples effectively. We also ignore the bias term, as it does not affect relative rankings and softmax is invariant to additive shifts.

G. Additional Experimental Results

In this section, we provide the complete benchmark results for the data ablation in Tab. 9 (data engine), Tab. 10 (real vs synthetic data), and Tab. 11 (data scaling). The complete model scaling results are shown in Tab. 12 and for contrastive loss in Tab. 13.

Additionally, we include additional ablation studies on (1) choice of the text encoder; (2) impact of video frame rate; and (3) BEST-RQ loss in audio encoder training. Moreover, we provide extensive retrieval results of Tab. 2-3, and joint-modal results of Tab. 4 in the main paper.

Text encoder . In Tab. 14, we present an ablation of different choices of text encoders for PE_{AV} with PE-L as the default visual encoder. We compare the performance of ModernBERT [72] with the original paired CLIP text encoder from PE-L [7]. After audio-video-text pre-training, we observe that the original PE-L text encoder performs better on video-centric tasks such as VTT [78] and Kinetics [37]. However, its performance lags on out-of-visual-domain concepts, such as audio events, in audio-focused tasks like text-to-audio retrieval in AudioCaps [38], and LID, emotion, vocal classification in Dynamic-SUPERB [82], where ModernBERT excels. Additionally, the PE-L text encoder supports a shorter context length (32 tokens) compared to ModernBERT (512 tokens). Therefore, we choose the pre-trained ModernBERT as our default text encoder. After the later fine-tuning phase, PE_{AV} catches up and outperforms the original PE models as shown in Tab. 3 in the main paper.

Video Frame Rate . Tab. 15 compares different video frame rates—30 FPS versus sampling a fixed set of 16

Caption Type	*Sound-Retrieval*				*Sound-Classification*				*Speech-Classification*				*Video-Retrieval*			*Video-Classification*		
	AudioCaps		VALORInternal		VGGSound		GTzan	Cremad	CV-13	D-SUPERB			VTT	MSVD	ANet	K400	K700	HMDB
	T→A	A→V	T→AV	A→V	A→T	AV→T	A→T	A→T	accent	lid	emo	vocal	T→V	T→V	T→V	V→T	V→T	V→T
EnCLAP	23.7	30.8	56.8	24.0	20.3	19.8	50.5	35.9	10.9	24.0	**40.0**	57.1	31.3	47.1	55.1	49.1	38.0	44.7
CoNeTTE	26.8	**36.1**	59.6	24.4	25.2	29.3	55.6	28.8	12.2	21.5	30.8	67.4	31.1	49.2	56.8	49.5	38.7	46.4
Stage-1	30.3	31.6	62.1	22.6	28.4	39.3	57.2	**38.4**	15.1	21.5	32.1	61.3	36.2	53.7	56.8	**56.6**	44.9	49.1
Stage-2	**32.2**	32.0	**64.6**	**25.2**	**29.7**	**44.3**	**59.8**	32.8	**16.8**	**30.0**	34.2	**73.6**	**36.2**	**53.9**	**57.7**	55.8	**45.3**	**51.1**

Table 9. **Data Engine**. Compared to off-the-shelf captioners (EnCLAP [39] and CoNeTTE [40]), the proposed data engine significantly improves the data quality by taking into account video context and confidence score.

Real Data	Syn. Data	*Sound-Retrieval*				*Sound-Classification*				*Speech-Classification*				*Video-Retrieval*			*Video-Classification*		
		AudioCaps		VALORInternal		VGGSound		GTzan	Cremad	CV-13	D-SUPERB			VTT	MSVD	ANet	K400	K700	HMDB
		T→A	A→V	T→AV	A→V	A→T	AV→T	A→T	A→T	accent	lid	emo	vocal	T→V	T→V	T→V	V→T	V→T	V→T
0x	1x	26.1	39.1	58.4	31.0	30.2	44.3	60.7	28.2	18.1	37.5	15.0	69.7	32.7	47.1	55.4	68.4	57.2	55.4
1x	0x	16.4	28.1	0.0	1.7	18.3	25.5	58.4	31.1	11.3	20.5	27.9	54.6	0.1	0.3	0.0	0.3	0.1	1.4
1x	1x	27.1	34.9	53.7	14.4	27.3	41.3	**65.1**	28.8	18.5	36.0	25.4	64.9	29.8	46.7	42.3	63.7	51.4	52.8
1x	10x	32.5	**44.3**	**63.2**	25.2	**31.9**	**44.8**	62.0	30.7	**23.5**	**46.0**	**37.5**	74.2	32.8	**47.8**	53.8	**66.4**	**54.9**	**58.2**
1x	20x	30.6	42.9	63.7	26.5	31.2	44.4	63.0	**31.3**	16.8	43.0	23.8	**77.1**	33.5	47.2	**56.1**	63.6	52.6	51.5
1x	30x	**30.8**	43.6	60.5	**29.0**	30.4	43.1	58.9	30.2	23.1	51.0	30.4	75.6	**33.8**	46.9	55.1	61.7	50.3	51.3

Table 10. **Comparing different mixing ratios of real and synthetic caption data.** Mixing both data types outperforms using only real or synthetic data. Higher synthetic ratios (till 1:10) further boost performance by improving diversity.

Data Scale	*Sound-Retrieval*				*Sound-Classification*				*Speech-Classification*				*Video-Retrieval*			*Video-Classification*		
	AudioCaps		VALORInternal		VGGSound		GTzan	Cremad	CV-13	D-SUPERB			VTT	MSVD	ANet	K400	K700	HMDB
	T→A	A→V	T→AV	A→V	A→T	AV→T	A→T	A→T	accent	lid	emo	vocal	T→V	T→V	T→V	V→T	V→T	V→T
$\mathcal{O}(2M)$	27.0	38.1	57.8	18.1	27.4	40.4	60.2	30.6	**20.2**	36.0	32.9	66.3	32.8	47.6	50.4	48.0	36.9	41.6
$\mathcal{O}(4M)$	29.6	46.1	64.6	22.1	30.1	43.4	63.9	28.6	17.2	41.5	25.8	66.4	34.9	51.6	53.1	51.5	40.5	51.7
$\mathcal{O}(8M)$	31.3	44.8	65.2	23.9	32.0	44.7	61.8	29.9	18.9	39.5	**39.2**	73.9	34.5	52.8	55.5	54.0	42.8	48.8
$\mathcal{O}(16M)$	32.1	48.7	65.9	24.1	33.1	45.2	62.1	26.1	19.3	41.0	35.4	74.4	**36.2**	**54.0**	56.5	54.2	43.0	50.3
$\mathcal{O}(32M)$	32.8	**50.6**	**67.5**	23.7	32.9	45.4	62.0	27.9	18.1	39.0	30.4	**76.5**	35.6	53.7	56.5	**55.8**	43.7	**51.9**
$\mathcal{O}(64M)$	**33.6**	47.0	67.0	**26.2**	**34.3**	**46.2**	**63.8**	**33.3**	16.0	**43.0**	24.2	71.8	35.6	53.7	**57.7**	55.1	**43.9**	50.7

Table 11. **Comparing performance as synthetic-caption data scale increases.** Performance increases with synthetic-caption data scale (peaking at 64M), underscoring the value of diverse set of audio-visual-text data.

A-layers	A-params	V-params	*Sound-Retrieval*				*Sound-Classification*				*Speech-Classification*				*Video-Retrieval*			*Video-Classification*		
			AudioCaps		VALORInternal		VGGSound		GTzan	Cremad	CV-13	D-SUPERB			VTT	MSVD	ANet	K400	K700	HMDB
			T→A	A→V	T→AV	A→V	A→T	AV→T	A→T	A→T	accent	lid	emo	vocal	T→V	T→V	T→V	V→T	V→T	V→T
8	0.03B	0.34B	29.5	38.3	67.1	16.6	28.7	45.0	58.0	28.4	19.3	32.0	26.7	70.7	35.8	**54.1**	54.9	55.3	44.1	51.2
12	0.09B	0.35B	32.0	46.0	67.8	23.1	31.4	45.4	63.1	27.9	21.9	38.0	35.4	72.2	36.6	54.0	56.3	55.7	44.5	51.6
16	0.21B	0.38B	33.2	48.4	**68.1**	25.6	33.3	45.4	61.8	32.0	19.3	41.0	32.9	73.5	36.6	53.7	56.3	55.2	43.6	48.9
20	0.41B	0.42B	**34.4**	**57.2**	67.9	**27.3**	33.6	**46.2**	62.8	**35.9**	21.9	**44.0**	31.7	74.0	**37.3**	53.2	**56.7**	**56.0**	**44.6**	**52.4**
24	0.70B	0.45B	**34.4**	53.4	66.9	24.4	33.2	45.7	62.6	30.1	**22.7**	38.0	**35.8**	76.9	35.2	53.3	55.7	54.4	43.7	50.1
28	1.11B	0.50B	34.3	56.7	66.6	24.7	**34.2**	44.9	**65.0**	32.7	16.0	34.0	32.5	**78.1**	35.5	53.3	**56.7**	53.3	43.0	49.0

Table 12. **Scaling the audio encoder**. Scaling from 0.03B to 1.11B parameters shows consistent performance gains with depth. The observed saturation around 20 layers is likely due to limited training steps and data in the ablation setup.

Loss	A-V	A-AT	A-AVT	V-AT	V-VT	V-AVT	AV-VT	AV-AVT	*Sound-Retrieval*			*Sound-Classification*				*Speech-Classification*				*Video-Retrieval*			*Video-Classification*		
									AudioCaps		VALOInterna	VGGSound		GTzan	Cremad	CV-13	D-SUPERB			VTT	MSVD	ANet	K400	K700	HMDB
									T→A	A→V	T→AV	A→V	A→T	AV→T	A→T	accent	lid	emo	vocal	T→V	T→V	T→V	V→T	V→T	V→T
SigLIP	-	✓	-	-	-	-	-	-	31.9	0.1	0.0	0.0	32.4	0.5	60.4	30.8	**23.5**	**52.5**	30.4	76.1	0.1	0.2	0.0	0.3	0.1	2.2
SigLIP	-	✓	-	-	✓	-	-	-	32.2	0.1	0.1	0.0	31.2	0.3	62.2	27.3	20.6	33.0	**32.5**	71.4	27.2	44.5	55.5	42.2	33.0	40.8
SigLIP	-	✓	-	-	✓	-	-	✓	33.0	0.0	47.1	0.1	31.9	0.3	56.6	30.6	18.1	36.0	32.1	71.8	26.1	42.5	53.0	41.8	31.3	42.1
SigLIP	✓	✓	-	-	✓	-	-	✓	31.9	45.6	47.5	24.7	30.6	0.4	53.3	25.2	21.9	38.5	32.1	70.0	27.3	44.2	47.0	39.7	30.1	36.4
SigLIP	✓	✓	-	✓	✓	-	✓	✓	31.4	**49.5**	66.3	**25.1**	32.5	45.1	61.0	26.2	18.1	43.5	**32.5**	**76.5**	**34.5**	53.9	56.7	**55.8**	**43.8**	**49.7**
SigLIP	✓	✓	✓	✓	✓	✓	✓	✓	**32.9**	45.7	**68.3**	21.8	**33.3**	**45.5**	**62.4**	**33.2**	17.2	47.5	30.4	74.7	33.9	**54.1**	**57.2**	54.2	43.1	48.9

Table 13. **Scaling the SigLIP objective.** A: Audio, V: Video, AT: Audio text caption, VT: Video text caption. Expanding the contrastive objective to cover more modality pairs strengthens cross-modal alignment and improves zero-shot retrieval and classification. Audio–text–only training lags behind, while adding cross-modality pairs (e.g., V→AT, AV→VT) yields further gains. Performance peaks when the objective includes all eight pairs (bottom row).

Text Encoder	*Sound-Retrieval*				*Sound-Classification*				*Speech-Classification*				*Video-Retrieval*			*Video-Classification*		
	AudioCaps		VALORInternal		VGGSound		GTzan	Cremad	CV-13	D-SUPERB			VTT	MSVD	ANet	K400	K700	HMDB
	T→A	A→V	T→AV	A→V	A→T	AV→T	A→T	A→T	accent	lid	emo	vocal	T→V	T→V	T→V	V→T	V→T	V→T
PE-Text	30.5	**49.1**	**66.6**	23.1	33.3	**47.3**	62.1	25.9	**19.3**	46.5	23.3	59.7	**45.3**	**60.4**	52.2	**66.7**	**57.6**	**55.3**
ModernBERT	**34.1**	49.0	66.5	**25.3**	**34.1**	46.1	**63.0**	**29.5**	18.1	**47.0**	**37.9**	**74.2**	36.2	53.3	**57.4**	54.6	44.0	50.2

Table 14. **Comparison of text encoder choices for PE_{AV} using the PE-L visual encoder.** ModernBERT outperforms the original PE-L text encoder on audio-focused tasks due to its longer context (512 tokens vs. 32) and its support of general text domain, while PE-L performs better on video-centric tasks. As noted in main results, ModernBERT catches up with and surpasses the PE-L text encoder after fine-tuning, making it the preferred choice for PE_{AV}.

PT frames	FT frames	*Sound-Retrieval*				*Sound-Classification*				*Speech-Classification*				*Video-Retrieval*			*Video-Classification*		
		AudioCaps		VALORInternal		VGGSound		GTzan	Cremad	CV-13	D-SUPERB			VTT	MSVD	ANet	K400	K700	HMDB
		T→A	A→V	T→AV	A→V	A→T	AV→T	A→T	A→T	accent	lid	emo	vocal	T→V	T→V	T→V	V→T	V→T	V→T
16	16	43.1	87.1	80.4	25.5	45.7	51.2	73.8	34.2	24.4	**65.0**	45.4	85.0	46.8	58.6	63.4	75.8	66.2	62.2
16	All	42.0	87.5	79.3	43.2	45.9	51.3	**74.2**	37.5	21.0	59.5	38.3	85.3	46.4	58.9	63.9	75.9	66.5	62.4
All	16	44.7	85.9	**83.7**	23.7	46.7	51.8	72.3	42.0	23.1	62.0	**47.9**	85.4	49.0	60.5	65.4	78.4	68.2	**66.0**
All	All	**45.8**	**89.0**	**83.7**	**49.0**	**47.1**	**52.4**	72.2	**43.3**	**25.6**	64.5	43.8	**86.1**	**51.9**	**60.8**	**66.5**	**78.9**	**69.0**	65.1

Table 15. **Impact of video frame rate.** "All": Encode all frames at 30 FPS. "16": Uniformly sample and encode 16 frames per video. Both configurations yield similar performance overall. However, a notable exception arises in the internal video-music retrieval task, which involves videos with wide variations in duration. In this case, the 30 FPS models capture duration information more effectively and achieve better performance.

SSL Loss	*Sound-Retrieval*				*Sound-Classification*				*Speech-Classification*				*Video-Retrieval*			*Video-Classification*		
	AudioCaps		VALORInternal		VGGSound		GTzan	Cremad	CV-13	D-SUPERB			VTT	MSVD	ANet	K400	K700	HMDB
	T→A	A→V	T→AV	A→V	A→T	AV→T	A→T	A→T	accent	lid	emo	vocal	T→V	T→V	T→V	V→T	V→T	V→T
None	30.7	44.1	65.7	**25.7**	31.4	44.6	61.2	30.3	17.7	35.5	**33.3**	**74.9**	34.6	53.6	55.6	54.2	43.1	49.3
NCE	31.9	46.9	67.7	25.3	31.6	45.0	**63.1**	26.3	14.7	36.0	32.9	72.2	35.6	**53.9**	**56.7**	54.0	43.0	48.9
BEST-RQ	**33.2**	**48.4**	**68.1**	25.6	**33.3**	**45.4**	61.8	**32.0**	**19.3**	**41.0**	32.9	73.5	**36.6**	53.7	56.3	**55.2**	**43.6**	**50.9**

Table 16. **Audio encoder losses: NCE vs. BEST-RQ.** NCE follows wav2vec 2.0 contrastive objective using DAC-VAE features as negatives but skips quantization. BEST-RQ delivers the strongest results, outperforming NCE and no-SSL baselines. Speech and sound tasks benefit the most from the finer-grained representations encouraged by BEST-RQ.

frames—during pre-training and fine-tuning. Because this ablation addresses a critical design choice and training various models at both frame rates is computationally intensive, we performed this ablation with our largest model PE_{AV}L. We train for the full pre-training duration of 250K steps followed by 50K steps of fine-tuning, ensuring that conclusions are drawn from the strongest configuration.

For most tasks, models trained with these different frame rate configurations exhibit similar performance, while 30 FPS sampling provides a modest advantage, especially during in the pre-training phase. However, models operating at higher frame rates achieve better results on downstream tasks that require fine-grained temporal understanding, such as ActivityNet and audio-video retrieval on the internal dataset with a wide duration variation. Notably, models trained with 30 FPS sampling inherently encode duration information, enabling them to retrieve audios or videos of similar length as the query. This also highlights a key limitation of existing audio-video retrieval benchmarks, which do not evaluate robustness to variation in duration. With this, for all other models, we adopt 30 FPS sampling during pre-training and later fine-tune with the same setup or with fixed 16-frame inputs.

BEST-RQ vs NCE Loss . In Tab. 16, we compare different SSL losses for the audio encoder. The NCE loss is similar to the contrastive loss in wav2vec 2.0, except that we skip the quantization step and use the DAC-VAE features as the negative samples directly. BEST-RQ offers the best overall results, significantly outperforming NCE loss and no SSL loss conditions. Video tasks retain performance while some even show slight improvement when the BEST-RQ loss is present. The results demonstrate the necessity of including BEST-RQ loss to enhance the performance of sound and speech tasks without compromising video capabilities, corroborating the hypothesis that encouraging fine-grained representations benefits speech-related tasks.

H. Qualitative Results

This section presents some examples of multimodal retrieval for qualitative analysis. First, Fig. 6 and 7 demonstrate qualitative video→text and text→video retrieval results by PE_{AV}. In Fig. 6, the ground truth video is successfully retrieved, while the top 2 and 3 retrieved videos show similar scenarios as well (water sports). Fig. 7 shows a similar phenomenon, but retrieved in the opposite direction. These examples showcase PE_{AV}'s natural capabilities

Model	AudioCaps $A \to T$	AudioCaps $T \to A$	AudioCaps $V \to T$	AudioCaps $T \to V$	AudioCaps $A \to V$	AudioCaps $V \to A$	AudioCaps $A{+}V \to T$	AudioCaps $T \to A{+}V$	AudioCaps $A{+}VT \to V$	AudioCaps $VA{+}T \to A$	Clotho $A \to T$	Clotho $T \to A$	Valor $A \to T$	Valor $T \to A$	Valor $V \to T$	Valor $T \to V$	Valor $A{+}V \to T$	Valor $T \to A{+}V$	VCTK $A \to T$	VGGSound $V \to A$	VGGSound $A \to V$	Internal $V \to A$	Internal $A \to V$	MSR-VTT $T \to V$	MSR-VTT $V \to T$	MSVD $T \to V$	MSVD $V \to T$	ActivityNet $T \to V$	ActivityNet $V \to T$	DiDeMo $T \to V$	DiDeMo $V \to T$	VATEX $T \to V$	VATEX $V \to T$
Baselines																																	
AFlamingo2	45.7	29.8	-	-	-	-	-	-	-	-	20.4	16.9	7.4	7.3	-	-	-	-	0.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ImgBind	9.6	6.6	11.3	7.6	51.6	51.3	-	-	-	-	5.5	3.9	4.9	5.4	35.8	36.1	-	-	0.4	10.5	10.8	2.8	2.8	40.6	42.9	47.9	70.9	36.6	34.1	36.0	38.2	69.8	69.8
CLAP-Fusion	43.3	35.4	-	-	-	-	-	-	-	-	20.2	17.7	5.4	5.5	-	-	-	-	0.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CLAP	43.7	31.6	-	-	-	-	-	-	-	-	21.0	16.6	6.5	5.8	-	-	-	-	0.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LangBind	27.1	19.7	14.2	10.6	10.7	9.1	-	-	-	-	17.1	13.3	5.6	6.5	46.9	46.8	-	-	0.2	1.8	1.6	1.3	1.4	48.6	48.7	55.6	78.8	48.0	48.8	43.5	44.7	82.9	83.1
M2D-CLAP	27.4	27.4	-	-	-	-	-	-	-	-	11.4	10.5	5.9	6.3	-	-	-	-	0.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MS-CLAP[23]	32.4	23.4	-	-	-	-	-	-	-	-	23.4	17.8	8.0	5.9	-	-	-	-	0.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16 Frames																																	
$PE_{AV}S$	59.4	41.2	26.2	18.6	75.4	75.4	56.1	40.9	71.9	89.7	33.6	24.0	30.2	29.8	70.3	70.1	75.9	76.3	**96.1**	33.3	34.1	17.7	17.9	46.7	49.6	60.1	86.4	63.4	64.8	48.7	49.0	94.2	93.7
$PE_{AV}B$	59.7	43.1	26.9	19.8	81.6	80.6	57.1	41.1	78.1	91.6	33.4	23.4	31.2	31.9	70.7	70.0	76.6	76.0	94.8	38.4	39.0	20.4	20.4	48.6	50.3	60.8	87.6	64.0	64.9	46.2	47.8	94.3	93.8
$PE_{AV}L$	62.0	44.7	26.9	19.5	85.9	86.1	58.0	41.6	83.1	94.6	32.6	22.8	35.2	35.0	70.8	70.9	76.0	76.8	85.6	44.6	45.2	23.7	23.9	49.0	50.5	60.5	**88.4**	65.4	66.5	48.9	50.1	94.9	94.4
30 FPS																																	
$PE_{AV}S$-OOD	55.2	40.2	25.4	17.7	76.6	75.3	52.6	36.8	73.0	89.1	32.3	23.4	28.9	28.0	70.5	69.8	76.0	76.1	73.1	27.7	28.5	41.4	40.8	50.9	51.0	60.8	87.6	66.7	65.9	51.4	51.8	95.1	94.6
$PE_{AV}B$-OOD	56.8	40.4	24.6	17.7	81.8	82.2	54.0	38.0	77.3	92.6	32.7	24.3	30.5	30.7	70.0	70.0	76.5	75.6	57.3	31.0	31.8	46.7	45.1	50.5	49.7	**61.2**	86.3	67.3	67.7	51.8	**52.4**	94.0	94.0
$PE_{AV}L$-OOD	60.0	43.4	26.2	18.2	87.5	86.1	53.3	38.7	82.9	93.3	33.3	23.7	34.7	34.2	71.4	70.2	75.8	76.0	50.7	35.6	36.5	**52.2**	**50.3**	50.4	**51.5**	61.0	87.5	**67.6**	**68.0**	**51.9**	51.5	95.1	94.4
$PE_{AV}S$	58.2	41.8	27.2	18.8	77.7	77.4	56.5	40.1	73.6	90.3	33.2	23.9	30.1	29.3	71.6	70.9	76.6	76.4	94.9	35.4	35.4	41.0	40.5	49.3	49.4	59.8	87.5	64.8	65.5	50.0	49.0	94.5	94.5
$PE_{AV}B$	60.0	42.7	28.3	19.6	83.5	83.7	56.1	41.0	79.9	93.2	33.7	23.8	31.0	30.8	**72.1**	**71.2**	**76.9**	**76.9**	94.9	40.7	40.7	45.9	44.6	47.7	48.4	60.7	87.6	65.7	65.9	49.3	50.1	94.9	94.4
$PE_{AV}L$ (PT)	48.5	33.7	22.6	14.7	83.4	83.3	47.9	33.2	-	-	26.3	17.5	24.2	24.0	56.1	57.1	62.6	63.3	16.7	32.6	33.9	50.6	47.8	35.5	36.6	49.8	79.6	62.0	64.0	44.8	46.3	87.1	87.2
$PE_{AV}L$	**63.3**	**45.8**	**29.1**	**20.8**	**89.0**	**88.3**	**58.2**	**42.6**	**84.0**	**95.2**	32.7	23.0	**36.4**	**35.1**	71.6	70.9	**76.9**	76.8	85.6	**47.8**	**48.3**	49.0	46.5	**51.9**	51.2	60.8	87.6	66.5	67.7	51.6	51.7	**95.1**	**94.8**

Table 17. **Full Zero-Shot Retrieval Results**. Per-dataset Recall@1 for all audio–text, video–text, and audio–video retrieval directions corresponding to the main audio and video tables. PE_{AV} consistently outperforms baseline models across most benchmarks and retrieval directions.

Model	*Zero-Shot Retrieval*												*Zero-Shot Classification*		
	AudioCaps $T\to A+V$	AudioCaps $T+A\to V$	AudioCaps $T+V\to A$	VALOR $T\to A+V$	VALOR $T+A\to V$	VALOR $T+V\to A$	VTT $T\to A+V$	VTT $T+A\to V$	VTT $T+V\to A$	DiDeMo $T\to A+V$	DiDeMo $T+A\to V$	DiDeMo $T+V\to A$	VGGSound $A\to T$	VGGSound $V\to T$	VGGSound $A+V\to T$
ImageBind† [25]	7.6	51.6	51.3	36.1	36.1	24.2	41.9	41.9	23.9	36.1	36.1	18.3	28.2	40.4	40.4
LanguageBind† [85]	19.7	10.7	19.7	46.8	46.8	6.5	50.9	50.9	3.7	44.2	44.2	5.5	26.0	45.4	45.4
PE_{AV}S-OOD†	40.2	76.6	75.3	69.8	69.8	57.7	52.0	64.0	65.3	53.7	55.2	55.5	40.0	46.3	46.3
PE_{AV}S-OOD	36.8	73.0	89.1	76.1	88.8	65.2	48.3	86.7	63.9	48.0	76.2	51.4	40.0	46.3	52.5
PE_{AV}B-OOD†	40.4	81.8	82.2	70.0	70.0	63.9	51.4	64.4	66.0	52.0	60.6	62.0	41.4	47.0	47.0
PE_{AV}B-OOD	38.0	77.3	92.6	75.6	90.5	70.9	50.1	87.8	65.8	48.0	81.0	57.8	41.4	47.0	52.1
PE_{AV}L-OOD†	43.4	87.5	86.1	70.2	72.2	73.1	51.5	70.0	71.6	52.7	66.8	67.7	43.9	46.7	46.7
PE_{AV}L-OOD	38.7	82.9	93.3	76.0	92.0	77.2	49.9	89.0	69.8	48.8	82.1	62.1	43.9	46.7	52.0
PE_{AV}S†	41.8	77.7	77.4	70.9	70.9	60.1	50.1	59.6	61.1	49.8	53.2	56.9	43.0	47.3	47.3
PE_{AV}S	40.1	73.6	90.3	76.4	89.8	67.6	48.3	85.9	58.6	40.0	75.5	44.5	43.0	47.3	52.2
PE_{AV}B†	42.7	83.5	83.7	71.2	71.2	65.3	50.1	62.3	63.8	49.6	61.6	63.4	44.5	47.8	47.8
PE_{AV}B	41.0	79.9	93.2	76.9	91.1	72.0	48.1	86.3	60.2	39.8	80.7	51.9	44.5	47.8	52.7
PE_{AV}L†	45.8	89.0	88.3	70.9	73.8	74.5	52.9	63.6	67.4	51.4	69.3	69.5	47.1	48.0	48.0
PE_{AV}L	42.6	84.0	95.2	76.8	93.0	78.8	49.0	85.3	65.1	43.0	80.8	61.6	47.1	48.0	51.8

Table 18. **Zero-shot joint-modal retrieval and classification.** For baselines and PE_{AV} variants marked with †, joint queries are approximated via the max over unimodal results: T+V→A = max(T→A, V→A), T→A+V = max(T→A, T→V), and T+A→V = max(T→V, A→V). All other PE_{AV} variants use native joint embeddings for T+V, A+V, and T+A. For all PE_{AV} models, we observe that joint embeddings are helpful when the input modalities are complimentary to each others. Specifically (i) for audio-only captions (*AudioCaps*) V+T→A significantly outperforms V→A and T→A; (ii) for visual captions (*DiDeMo* & *MSR-VTT*) A+T→V improves over A→V and T→V; (iii) for audio-visual captions (*VALOR*) both V+T→A and A+T→V help; (iv) *VGGSound*: A+V→T exceeds A→T and V→T.

for capturing information from unimodal data and aligning content across modalities.

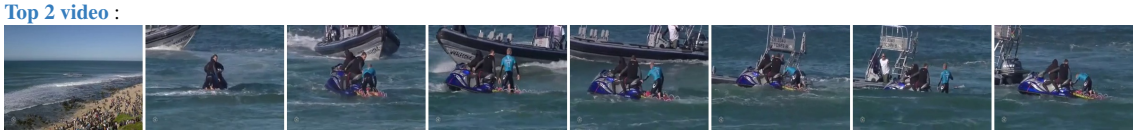
Next, the following examples demonstrate PE_{AV}'s novel capability to extract and relate multiple modalities. Fig. 8 showcases joint audio+text→video retrieval results. The additional audio context helps break the ties of the video and retrieve the corresponding video correctly compared with using either text or audio as the query. Moreover, in Fig. 9, retrieval based solely on video or audio omits key information. E.g., "audio → text" is unsuccessful because the visual cue of "car" is challenging to extract from the audio. By leveraging joint multimodal retrieval, PE_{AV} incorporates both audio and video context, enabling it to correctly identify the top-1 result.

Furthermore, Fig. 10 demonstrates the speech → audio caption/transcript retrieval capabilities. First, when the audio caption is perturbed (similar and wrong examples), the retrieval score decreases, indicating the success of identifying the correct speaker, speaking style, and recording environment. In the second section, we replace some words in the correct transcript with similar-sounding words and find slightly lower scores. In contrast, rewriting the transcript with different words while preserving meaning significantly decreases scores, implying that PE_{AV} captures pronunciation more than meaning in speech. Moreover, completely irrelevant transcripts lead to even worse scores. The final section shows the case in which both the caption and the transcript are present in the retrieved text. The highest score is achieved when both the caption and the transcript are correct, indicating that providing more textual information helps retrieve the desired speech signal. Finally, we replace the captions and transcripts with incorrect ones and find that incorrect transcripts decrease scores the most. The results reveal transcripts have a higher impact on the score than audio descriptions, offering more accurate retrieval between speech and text when the transcript is presented. Overall, the results strongly suggest the usefulness of PE_{AV} for speech-related tasks.

Query: in the ocean a man on a surfboard rides a wave

Top 1 video : (**ground truth**)

Top 2 video :



Top 3 video :

Figure 6. Video-caption → Video retrieval results from PE_{AV}. Ground truth (if present) is bolded.

Query Video :

Top 1: a helicopter moving in air and red and yellow dress man hand touching speaking in snow land wearing helmet displaying on screen (**ground truth**)

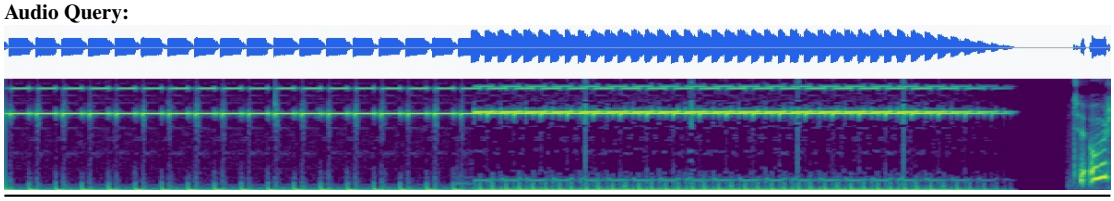
Top 2: a man rides a lift to the top of a mountain

Top 3: flight is shaken and the pilots trying to land the flight while they opened the air

Figure 7. Video → Video-caption retrieval results from PE_{AV}. Ground truth (if present) is bolded.

Text Query: In the room, a man pressed the alarm with his index finger, and the alarm rang.

Audio Query:



Text → Video

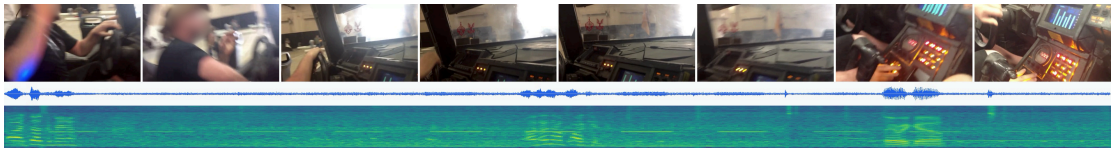
(**Caption:** A person presses the button of the instrument watch on the wall, and the instrument drips.)

Audio → Video

(**Caption:** In the enclosed space, a mouse whirled in the dripping sound, the picture turned into two rats.)

Text + Audio → Video (ground truth)

Figure 8. T + A → Video retrieval results from PE_{AV}. Ground truth (if present) is bolded.



Video

Audio → Text

Top 1: A man in a life-saving suit stood beside the manhole cover, directing the rumbling engineering vehicle to reverse, and then angling the vehicle's gear to the sewer.
Top 2: In the field, a command officer in a fluorescent green work suit was waving a flag to direct a farm machine vehicle, with the sound and beeping of vehicles and the voice of men.
Top 3: Outside, a man sits in a car talking while driving slowly as the car dribbles. (**ground truth**)

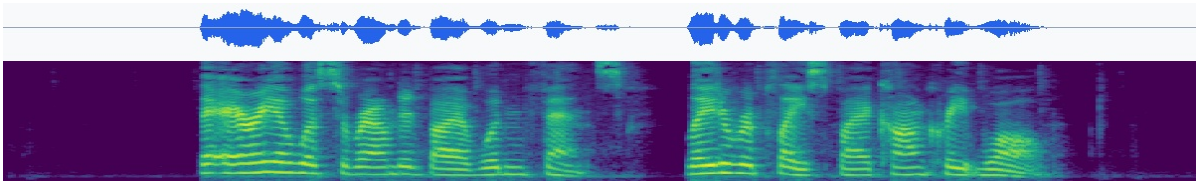
Video → Text

Top 1: A man fiddled with the steering wheel in the driver's seat, making a rustling sound, and the roar of machine operation from time to time in the distance.
Top 2: The man was sitting in the driver's seat talking, the picture shaking, saw the co-pilot and the windows open inside and behind the car.
Top 3: A man is introducing virtual technology to his buddies at the creaking edge of the wheel.

Audio + Video → Text

Top 1: Outside, a man sits in a car talking while driving slowly as the car dribbles. (**ground truth**)
Top 2: A man fiddled with the steering wheel in the driver's seat, making a rustling sound, and the roar of machine operation from time to time in the distance.
Top 3: The man was sitting in the driver's seat talking, the picture shaking, saw the co-pilot and the windows open inside and behind the car.

Figure 9. A/V/AV → AV-caption retrieval results from PE_{AV}. Ground truth (if present) is bolded.



Caption	Score
Correct: A middle-aged female voice, spoken at a normal pace, with a normal pitch and quality.	**0.571**
Similar 1: A young female voice, spoken at a normal pace, with a normal pitch and quality.	0.250
Similar 2: A middle-aged male voice, spoken at a normal pace, with a normal pitch and quality.	−0.207
Similar 3: A middle-aged female voice, spoken at a fast pace, with a normal pitch and quality.	0.109
Wrong 1: A young female voice, spoken at a fast pace, with a high pitch and low recording quality.	−0.538
Wrong 2: A middle-aged male voice, spoken at a slow pace, with a low pitch and normal quality.	−0.582
Wrong 3: A young male voice, spoken at a fast pace, with a normal pitch and quality.	−0.739

Transcript	Score
Correct: The person says: "The area was surrounded by a wooden fence, later replaced by a concrete wall."	**0.399**
Similar Pronunciation 1: The person says: "The era was surrounded by a wooden sense, later replayed by a convict call."	0.227
Similar Pronunciation 2: The person says: "The aria was confounded by a warden fence, later replaced by a con fleet wall."	0.268
Similar Pronunciation 3: The person says: "The airy was surrendered by a wooden lens, later rephrased by a concrete mall."	0.172
Similar Meaning 1: The person says: "The yard was enclosed by a timber fence, later swapped for a stone wall."	−0.331
Similar Meaning 2: The person says: "The field was bordered by a wooden fence, which was later rebuilt in concrete."	−0.233
Similar Meaning 3: The person says: "A wooden fence once circled the property, but it was replaced by a solid wall."	−0.478
Wrong 1: The person says: "Man in red tshirt and baseball cap viewed from above he is has a pile of posters."	−0.956
Wrong 2: The person says: "Hash trees allow efficient and secure verification of the contents of large data structures."	−2.469
Wrong 3: The person says: "Armand immigrated to the United States from France and sold hats as an occupation."	−1.981

Caption + Transcript	Score
Correct: A middle-aged female voice, spoken at a normal pace, with a normal pitch and quality. The person says: "The area was surrounded by a wooden fence, later replaced by a concrete wall."	**0.984**
Wrong Caption + Correct Transcript 1: A young female voice, spoken at a fast pace, with a high pitch and low recording quality. The person says: "The area was surrounded by a wooden fence, later replaced by a concrete wall."	0.170
Wrong Caption + Correct Transcript 2: A middle-aged male voice, spoken at a slow pace, with a low pitch and normal quality. The person says: "The area was surrounded by a wooden fence, later replaced by a concrete wall."	0.205
Wrong Caption + Correct Transcript 3: A young male voice, spoken at a fast pace, with a normal pitch and quality. The person says: "The area was surrounded by a wooden fence, later replaced by a concrete wall."	0.018
Correct Caption + Wrong Transcript 1: A middle-aged female voice, spoken at a normal pace, with a normal pitch and quality. The person says: "Man in red tshirt and baseball cap viewed from above he is has a pile of posters."	−0.536
Correct Caption + Wrong Transcript 2: A middle-aged female voice, spoken at a normal pace, with a normal pitch and quality. The person says: "Hash trees allow efficient and secure verification of the contents of large data structures."	−1.737
Correct Caption + Wrong Transcript 3: A middle-aged female voice, spoken at a normal pace, with a normal pitch and quality. The person says: "Armand immigrated to the United States from France and sold hats as an occupation."	−1.425

Figure 10. Speech → Audio-caption and transcript retrieval results from PE_{AV}. The scores indicate the embedding similarity scores between the [CLS-A] and [CLS-AT].

References

[1] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

[2] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, 2020.

[3] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pages 1298–1312. PMLR, 2022.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv:2308.12966*, 2023.

[5] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey A. Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bosnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier J. Hénaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. PaliGemma: A versatile 3b VLM for transfer. *arXiv:2407.07726*, 2024.

[6] Çağdaş Bilen, Giacomo Ferroni, Francesco Tuveri, Juan Azcarreta, and Sacha Krstulović. A framework for the robust evaluation of sound event detection. In *ICASSP*, pages 61–65, 2020.

[7] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv:2504.13181*, 2025.

[8] Heng-Jui Chang, Saurabhchand Bhati, James Glass, and Alexander H Liu. Usad: Universal speech and audio representation via distillation. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2025.

[9] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16 (6):1505–1518, 2022.

[10] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*, 2022.

[11] 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 *International Conference on Machine Learning*, pages 5178–5193, 2023.

[12] Wenxi Chen, Yuzhe Liang, Ziyang Ma, Zhisheng Zheng, and Xie Chen. Eat: Self-supervised pre-training with efficient audio transformer. *arXiv preprint arXiv:2401.03497*, 2024.

[13] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024.

[14] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. arXiv preprint arXiv:2109.04290, 2021. Version v3, Nov 2021.

[15] Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad Maaz, Yale Song, Tengyu Ma, Shuming Hu, Hanoona Rasheed, Peize Sun, Po-Yao Huang, Daniel Bolya, Suyog Jain, Miguel Martin, Huiyu Wang, Nikhila Ravi, Shashank Jain, Temmy Stark, Shane Moon, Babak Damavandi, Vivian Lee, Andrew Westbury, Salman Khan, Philipp Krähenbühl, Piotr Dollár, Lorenzo Torresani, Kristen Grauman, and Christoph Feichtenhofer. Perceptionlm: Open-access data and models for detailed visual understanding. *arXiv:2504.13180*, 2025.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[17] Karan Desai and Justin Johnson. VirTex: Learning visual representations from textual annotations. In *CVPR*, 2021.

[18] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: an audio captioning dataset. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740, 2020.

[19] Janek Ebbers, Reinhold Haeb-Umbach, and Romain Serizel. Threshold independent evaluation of sound event detection scores. In *ICASSP*, pages 1021–1025, 2022.

[20] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. CLAP: Learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[21] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving CLIP training with language rewrites. In *NeurIPS*, 2023.

[22] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. In *ICLR*, 2024.

[23] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran

Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. DataComp: In search of the next generation of multimodal datasets. In *NeurIPS*, 2023.

[24] 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 *Forty-second International Conference on Machine Learning*, 2025.

[25] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15180–15190, 2023.

[26] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10699–10709, 2022.

[27] Yuan Gong, Andrew Rouditchenko, Alexander H. Liu, David Harwath, Leonid Karlinsky, Hilde Kuehne, and James R. Glass. Contrastive audio-visual masked autoencoder. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Jiarui Hai, Helin Wang, Weizhe Guo, and Mounya Elhilali. Flexsed: Towards open-vocabulary sound event detection. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2025.

[29] Shawn Hershey, Daniel PW 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*, pages 366–370, 2021.

[30] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29: 3451–3460, 2021.

[31] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35: 28708–28720, 2022.

[32] Po-Yao Huang, Vasu Sharma, Hu Xu, Chaitanya Ryali, haoqi fan, Yanghao Li, Shang-Wen Li, Gargi Ghosh, Jitendra Malik, and Christoph Feichtenhofer. Mavil: Masked audio-video learners. In *Advances in Neural Information Processing Systems*, pages 20371–20393. Curran Associates, Inc., 2023.

[33] Hyeonuk Nam and Seong-Hu Kim and Byeong-Yun Ko and Yong-Hwa Park. Frequency Dynamic Convolution: Frequency-Adaptive Pattern Recognition for Sound Event Detection. In *Interspeech 2022*, pages 2763–2767, 2022.

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

[35] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[36] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, Oriane Siméoni, Huy V. Vo, Patrick Labatut, and Piotr Bojanowski. DINOv2 meets text: A unified framework for image- and pixel-level vision-language alignment. In *CVPR*, 2025.

[37] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv:1705.06950*, 2017.

[38] 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, Volume 1 (Long and Short Papers)*, pages 119–132, 2019.

[39] Jaeyeon Kim, Jaeyoon Jung, Jinjoo Lee, and Sang Hoon Woo. Enclap: Combining neural audio codec and audio-text joint embedding for automated audio captioning. *arXiv preprint arXiv:2401.17690*, 2024.

[40] Étienne Labbé, Thomas Pellegrini, and Julien Pinquier. Conette: An efficient audio captioning system leveraging multiple datasets with task embedding, 2023.

[41] Zhengfeng Lai, Haotian Zhang, Bowen Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, Yinfei Yang, and Meng Cao. VeCLIP: Improving CLIP training via visual-enriched captions. In *ECCV*, 2024.

[42] Xianhang Li, Zeyu Wang, and Cihang Xie. CLIPA-v2: Scaling CLIP training with 81.1% zero-shot imagenet accuracy within a $10,000 budget; an extra $4,000 unlocks 81.8% accuracy. *arXiv:2306.15658*, 2023.

[43] Xian Li, Nian Shao, and Xiaofei Li. Self-supervised audio teacher-student transformer for both clip-level and frame-level tasks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1336–1351, 2024.

[44] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *CVPR*, 2023.

[45] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, et al. Mert: Acoustic music understanding model with large-scale self-supervised training. In *International Conference on Learning Representations*, 2024.

[46] Alexander H Liu, Heng-Jui Chang, Michael Auli, Wei-Ning Hsu, and Jim Glass. Dinosr: Self-distillation and online clustering for self-supervised speech representation learning. In *Advances in Neural Information Processing Systems*, pages 58346–58362, 2023.

[47] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2024.

[48] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Guoli Yin, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. MM1: methods, analysis and insights from multimodal LLM pre-training. In *ECCV*, 2024.

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

[50] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, and Mark D Plumbley. Sound event detection: A tutorial. *IEEE Signal Processing Magazine*, 38(5):67–83, 2021.

[51] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. SLIP: Self-supervision meets language-image pre-training. In *ECCV*, 2022.

[52] Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. SILC: Improving vision language pretraining with self-distillation. In *ECCV*, 2024.

[53] Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. In *NeurIPS*, 2023.

[54] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, Masahiro Yasuda, Shunsuke Tsubaki, and Keisuke Imoto. M2D-CLAP: Masked Modeling Duo Meets CLAP for Learning General-purpose Audio-Language Representation. *Interspeech*, 2024.

[55] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv:2306.14824*, 2023.

[56] Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52, 2024.

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[58] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[59] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello. Scaper: A library for soundscape synthesis and augmentation. In *WASPAA*, pages 344–348, 2017.

[60] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *ECCV*, 2020.

[61] Florian Schmid, Tobias Morocutti, Francesco Foscarin, Jan Schlüter, Paul Primus, and Gerhard Widmer. Effective pre-training of audio transformers for sound event detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[62] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks*, 2022.

[63] 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)*, pages 911–915. IEEE, 2024.

[64] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

[65] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved training techniques for clip at scale. *arXiv:2303.15389*, 2023.

[66] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, 2024.

[67] Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. In *NeurIPS*, 2023.

[68] Nicolas Turpault, Romain Serizel, Ankit Shah, and Justin Salamon. DESED_public_eval. https://doi.org/10.5281/zenodo.3588172, 2019. Dataset. Creative Commons Attribution 4.0 International (CC-BY 4.0).

[69] Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin Salamon. Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. In *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[70] Bo Wan, Michael Tschannen, Yongqin Xian, Filip Pavetic, Ibrahim M Alabdulmohsin, Xiao Wang, André Susano Pinto, Andreas Steiner, Lucas Beyer, and Xiaohua Zhai. LocCa: Visual pretraining with location-aware captioners. In *NeurIPS*, 2024.

[71] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Jilan Xu, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. InternVideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, 2024.

[72] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning

and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, Vienna, Austria, 2025.

[73] Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[74] Yusong Wu, Christos Tsirigotis, Ke Chen, Cheng-Zhi Anna Huang, Aaron Courville, Oriol Nieto, Prem Seetharaman, and Justin Salamon. SED-augmented adobe audition sound effects dataset (ASFX-SED). https://doi.org/10.5281/zenodo.15866339, 2025. Dataset. Adobe Research License.

[75] 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 *ICML*, 2025.

[76] Hu Xu, Po-Yao Huang, Xiaoqing Ellen Tan, Ching-Feng Yeh, Jacob Kahn, Christine Jou, Gargi Ghosh, Omer Levy, Luke Zettlemoyer, Wen tau Yih, Shang-Wen Li, Saining Xie, and Christoph Feichtenhofer. Altogether: Image captioning via re-aligning alt-text. In *EMNLP*, 2024.

[77] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In *ICLR*, 2024.

[78] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[79] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

[80] Ching-Feng Yeh, Po-Yao Huang, Vasu Sharma, Shang-Wen Li, and Gargi Gosh. Flap: Fast language-audio pre-training. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8, 2023.

[81] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *TMLR*, 2022.

[82] Chien yu Huang, Wei-Chih Chen, Shu wen Yang, Andy T. Liu, Chen-An Li, Yu-Xiang Lin, Wei-Cheng Tseng, Anuj Diwan, Yi-Jen Shih, Jiatong Shi, William Chen, Chih-Kai Yang, Wenze Ren, Xuanjun Chen, Chi-Yuan Hsiao, Puyuan Peng, Shih-Heng Wang, Chun-Yi Kuan, Ke-Han Lu, Kai-Wei Chang, Fabian Ritter-Gutierrez, Kuan-Po Huang, Siddhant Arora, You-Kuan Lin, Ming To Chuang, Eunjung Yeo, Kalvin Chang, Chung-Ming Chien, Kwanghee Choi, Jun-You Wang, Cheng-Hsiu Hsieh, Yi-Cheng Lin, Chee-En Yu, I-Hsiang Chiu, Heitor R. Guimarães, Jionghao Han, Tzu-Quan Lin, Tzu-Yuan Lin, Homu Chang, Ting-Wu Chang, Chun Wei Chen, Shou-Jen Chen, Yu-Hua Chen, Hsi-Chun Cheng, Kunal Dhawan, Jia-Lin Fang, Shi-Xin Fang, Kuan-Yu Fang Chiang, Chi An Fu, Hsien-Fu Hsiao, Ching Yu Hsu, Shao-Syuan Huang, Lee Chen Wei, Hsi-Che Lin, Hsuan-Hao Lin, Hsuan-Ting Lin, Jian-Ren Lin, Ting-Chun Liu, Li-Chun Lu, Tsung-Min Pai, Ankita Pasad, Shih-Yun Shan Kuan, Suwon Shon, Yuxun Tang, Yun-Shao Tsai, Jui-Chiang Wei, Tzu-Chieh Wei, Chengxi Wu, Dien-Ruei Wu, Chao-Han Huck Yang, Chieh-Chi Yang, Jia Qi Yip, Shao-Xiang Yuan, Vahid Noroozi, Zhehuai Chen, Haibin Wu, Karen Livescu, David Harwath, Shinji Watanabe, and Hung yi Lee. Dynamic-superb phase-2: A collaboratively expanding benchmark for measuring the capabilities of spoken language models with 180 tasks, 2025.

[83] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[84] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. Contrastive learning of medical visual representations from paired images and text. In *MLHC*, 2022.

[85] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, Wang HongFa, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Cai Wan Zhang, Zhifeng Li, Wei Liu, and Li Yuan. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment, 2023.

[86] Haina Zhu, Yizhi Zhou, Hangting Chen, Jianwei Yu, Ziyang Ma, Rongzhi Gu, Yi Luo, Wei Tan, and Xie Chen. Muq: Self-supervised music representation learning with mel residual vector quantization. *arXiv preprint arXiv:2501.01108*, 2025.