# CAV-MAE Sync: Improving Contrastive Audio-Visual Mask Autoencoders via Fine-Grained Alignment

Supplementary Material

## 6. Implementation Details

In this section, we provide details on our data preprocessing, model architecture, and training hyperparameters.

### 6.1. Data Preprocessing

For input, we sample 16 frames uniformly from each video, along with corresponding 4-second audio segments with temporal alignment determined as described in Section 3.3. For audio, each waveform is first converted to a sequence of 128-dimensional log Mel filterbank (fbank) features computed with a 25 ms Hanning window every 10 ms., we extract 4-second segments from the spectrograms of size $128 \times 416$, chosen to enable non-overlapping patch extraction. We use a patch size of $16 \times 16$, resulting in 208 audio tokens. The RGB images are resized and center-cropped to $224 \times 224$ pixels, following the same patch extraction process, resulting in 196 visual tokens.

### 6.2. Model Architecture

For the model architecture, we initialize our modality-specific encoders from the same MAE checkpoints as CAV-MAE [14], but conduct our own pretraining rather than using their pretrained weights. Our single-modality encoders each contain 11 transformer layers, followed by a 1-layer joint encoder for cross-modal fusion. This was chosen to maintain the compatibility with the original MAE architecture, from which we initialize our weights. For the linear probing downstream task, the final transformer classifier consists of 2 layers followed by a single linear layer applied to the CLS token.

### 6.3. Training

In all experiments we use a single backbone model pretrained on AudioSet2M [11]. During pretraining, we use a masking ratio of 0.75 for both modalities with unstructured masking following [14]. We conduct ablation studies on the impact of different masking ratios in Table 8.

The contrastive and reconstruction loss weights are set to $\lambda_c = 0.1$ and $\lambda_r = 1.0$ respectively. For the contrastive loss weight $\lambda_c$, we use a higher value of 0.1 compared to CAV-MAE's 0.01, since aligning multiple fine-grained audio segments with their corresponding frames is a more challenging task than aligning a single global audio representation.

We use a batch size of 512 and an initial learning rate of $2 \times 10^{-4}$ with cosine learning rate scheduling. We pretrain for 25 epochs in total. Detailed hyperparameters for both pretraining and finetuning stages are provided in Table 9.

## 7. Modality-Specific Linear Probing

Table 10 presents the results of our modality-specific linear probing experiments. We compare the performance of models trained with audio-only, video-only, and audio-visual inputs on two datasets: AudioSet-20K (AS20K) [11] and VGGSound [6]. The results are reported using mean Average Precision (mAP) for AS20K and accuracy for VGGSound. The audio-visual model outperforms both the audio-only and video-only models, achieving the highest scores of 30.5 mAP on AS20K and 52.7 accuracy on VGGSound. This demonstrates the effectiveness of combining audio and visual modalities for classification tasks.

| | Pretraining | Probing | |
|---|---|---|---|
| Dataset | AS-2M | AS-20K | VGG |
| Optimizer | Adam, weight decay=5e−7, betas=(0.95, 0.999) | | |
| Learning Rate | 2e−4 | 5e−2 | 1e−3 |
| LR Scheduler | Cosine | Cosine | Cosine |
| Epochs | 25 | 15 | 10 |
| Linear Warmup Epochs | 2.5 | 1.5 | 1 |
| Batch size | $8 \times 64$ | 48 | 48 |
| GPUs | $8 \times$ AMD MI200 | $2 \times$ AMD MI200 | |
| Training time | 16h | 2h | 10h |
| Audio Input Size | $128 \times 416$ | $16 \times 128 \times 416$ | $16 \times 128 \times 416$ |
| Class Balance Sampling | No | No | Yes |
| Mixup | No | Yes | Yes |
| Random Time Shifting | Yes | Yes | Yes |
| Loss Function | - | BCE | CE |
| Weight Averaging | No | Yes | Yes |
| Input Norm Mean | −5.081 | −5.081 | −5.081 |
| Input Norm STD | 4.485 | 4.485 | 4.485 |

Table 9. Our pre-training and fine-tuning hyperparameters.

| **Modality** | AS20K↑ | VGGSound↑ |
|---|---|---|
| Audio Only | 8.7 | 30.3 |
| Video Only | 22.3 | 46.3 |
| Audio-Visual | **30.5** | **52.7** |

Table 10. Comparing audio-visual classification performance using linear probing. Numbers reported for AS20K are calculated using mAP and VGGSound with accuracy.

## 8. Retrieval Aggregation Methods

We evaluate different strategies for aggregating similarity scores in cross-modal retrieval, as shown in Table 11. For any pair of videos, we compute a similarity matrix where each element represents the similarity between a visual token from the query video and an audio token from the target video, as detailed in Section 3. The "diagonal mean" strategy averages only the diagonal elements of this matrix, focusing on temporally aligned token pairs, while "block mean" averages all pairwise similarities between the two videos. Our experiments show that "diagonal mean" consistently outperforms other approaches, including "block mean" and maximum-based strategies ("diagonal max" and "block max"). This suggests that emphasizing temporal alignment through diagonal averaging better captures the audio-visual correspondences compared to considering all possible token pairs or focusing on single maximum similarity values. The advantage is particularly pronounced on

AudioSet, where "diagonal mean" achieves 35.2% R@1, surpassing "block mean" by 2.7% and "diagonal max" by 6.7% absolute.

| Strategy | AudioSet Eval Subset | | | VGGSound Eval Subset | | |
|---|---|---|---|---|---|---|
| | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 |
| block max | 27.8 | 51.9 | 62.4 | 23.0 | 43.9 | 54.3 |
| diag max | 28.5 | 51.9 | 61.3 | 22.6 | 43.5 | 54.6 |
| block mean | 32.5 | 54.8 | 65.0 | 25.9 | 48.2 | 59.2 |
| **diag mean** | **35.2** | **58.3** | **67.6** | **27.9** | **51.7** | **61.8** |

Table 11. Comparison of retrieval aggregation strategies for cross-modal retrieval. The "diagonal mean" aggregation achieves the best performance, surpassing "block mean" by 2.7% and "diagonal max" by 6.7% absolute on AudioSet R@1. (V → A Retrieval.)

## 9. Register Tokens Analysis

In this section, we analyze the information captured by different token types through linear probing on the AudioSet-20k dataset. Table 12 shows the performance comparison between register tokens, patch tokens, and the global token. Our findings reveal that register tokens serve as an intermediate representation between highly localized patch tokens and the global token.

With our proposed 8 registers setup, the global token achieves the highest performance (30.8 mAP), followed by register tokens (17.8 mAP) and patch tokens (11.7 mAP). This hierarchy indicates that register tokens effectively aggregate information from patches while maintaining more specialized representations than the global token. The performance gap between register and patch tokens (17.8 vs 11.7 mAP) shows that registers capture more semantic information than individual patches.

Adding registers improves global token performance from 27.1 to 30.8 mAP, suggesting that registers serve as a "buffer" to aggregate information independently. Interestingly, we observe that register tokens reduce patch token performance from 12.3 to 11.7 mAP, indicating that registers are drawing contextual information away from patches. This supports our disentanglement hypothesis, while we don't observe the high-norm tokens reported in the paper that first introduced register tokens to vision transformers [10], this reduction in patch performance suggests registers are successfully serving as intermediaries between local and global representations. Our design uses registers to untangle the competing generative (patch tokens) and contrastive (global token) objectives. The empirical improvement in global token performance, coupled with the reduction in patch token performance, demonstrates that this additional buffer, not directly controlled by any loss, effectively helps the model develop more specialized representations.

| # Registers | AS20k (mAP) ↑ | | |
|---|---|---|---|
| | Register | Patch | Global |
| 0 | N/A | 12.3 | 27.1 |
| 8 | 17.8 | 11.7 | 30.8 |

Table 12. Linear probing of models with and without registers on AudioSet-20k, using various tokens as representation.

## 10. Sound Prompted Segmentation Examples

Figure 5 shows our model's sound-prompted segmentation results. As described in Section 3, we compute localization maps by calculating cosine similarities between the global audio token and visual patch tokens. Using VGGSound audios from class labels like "writing on blackboard with chalk", "roller coaster running", and "airplane" as prompts, our model generates localization maps highlighting relevant image regions. The results demonstrate strong audio-visual token alignment for scenes with clear objects like airplanes, also for more complex scenes like roller coasters with high visual clutter, which are naturally more challenging.

Notably, while specific classes like "writing on blackboard with chalk" and "roller coaster running" are not explicitly labeled in our AudioSet-2M pretraining dataset, examples of these sounds still exist under different labels. Despite this labeling discrepancy and the domain gap compared to datasets like VGGSound, our model demonstrates strong localization capabilities. For instance, in the "writing on blackboard" example, the model precisely highlights the blackboard region, while in the roller coaster examples, it effectively focuses on the coaster structure within visually cluttered scenes. These results are particularly encouraging given that our model was trained in a self-supervised manner on AudioSet-2M without explicit localization objectives. This robustness to unlabeled classes suggests that our global contrastive learning approach inherently learns some degree of spatial correspondences between audio and visual signals.

## 11. Intra-Instance Temporal Segmentation

To investigate how finer-grained audio representations impact the understanding of video clips, we conduct a qualitative analysis of temporal segmentation within samples from the AudioSet dataset. For this experiment, we manually annotate the occurrence of the classes throughout the video. In many cases, especially when multiple classes are present, different classes occur in separate segments of a video, not necessarily overlapping. We observe how well our model's features can discern between these classes of audio events by extracting features from each of the 16 frames and corresponding audio segments.

We apply a simple adaptive clustering algorithm to the extracted features to create temporal segments within each video. Using Agglomerative Clustering with a dynamic distance threshold, we iteratively adjust the threshold to achieve the desired number of segments, which if set to 5. If this fails, we fall back to K-means clustering. Figure 6 shows examples where our model can segment different classes based on the audio, even when the visual information remains nearly constant. We compare segmentation results using audio-only, video-only, and combined features to demonstrate how audio features capture most of the semantic changes occurring within videos. This highlights why using a single global audio representation would be insufficient, as it would fail to capture these important temporal variations in the audio signal.

In the first example with the red car, while the visual scene remains largely static, the model detects distinct "speech" and "toot" segments, demonstrating audio's ability to capture semantic transitions invisible in the visual domain. The second sequence shows clear delineation between speech and breathing segments, with

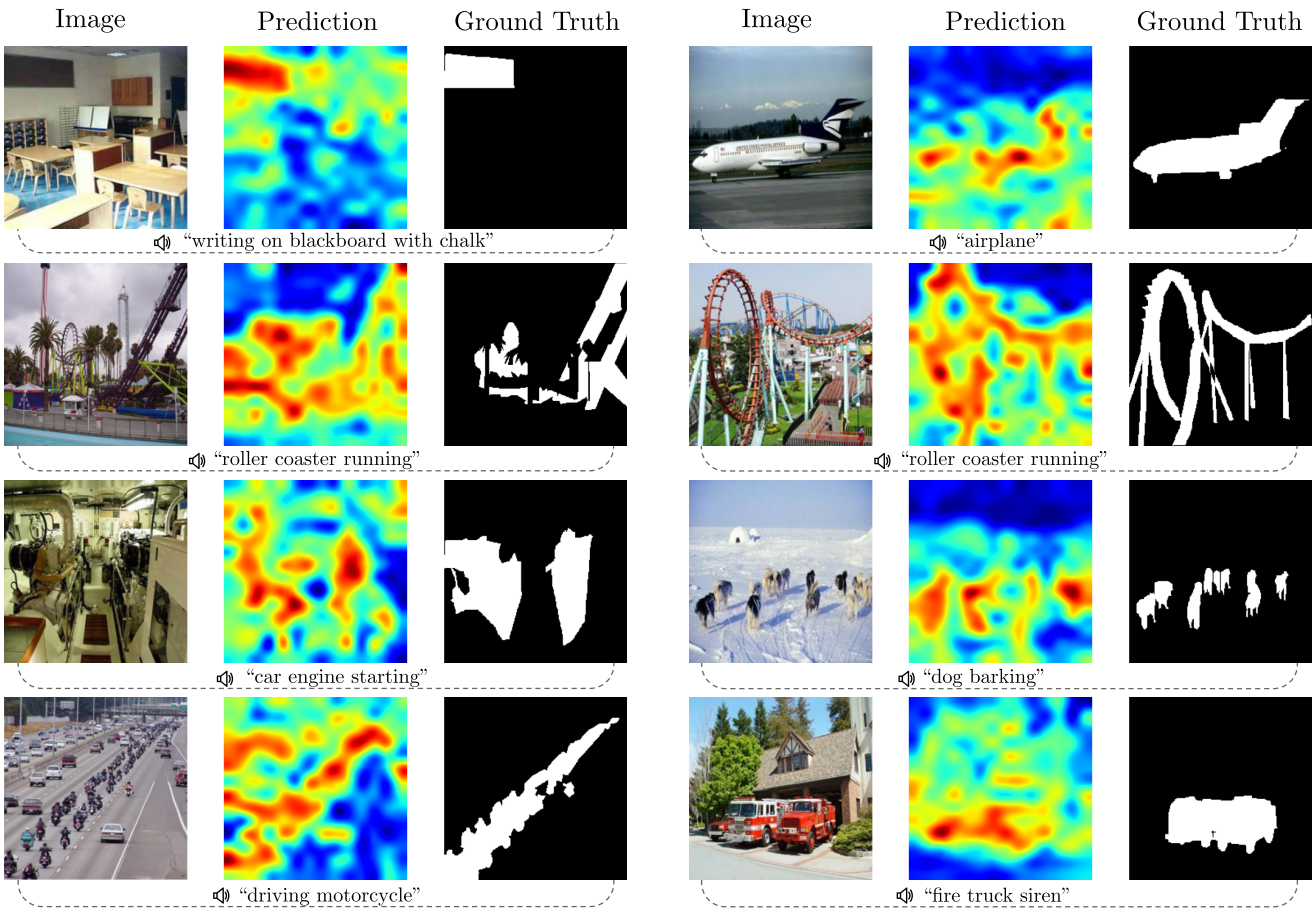


Figure 5. Sound-prompted segmentation results showing localization maps generated from audio prompts from VGGSound classes like ”writing on blackboard”, ”roller coaster”, and ”airplane”. The model highlights relevant image regions corresponding to the audio, demonstrating strong audio-visual alignment for clear objects while more complex, cluttered scenes remain challenging.

audio features driving the segmentation despite minimal visual changes. The third example captures the transition from applause to speech in a crowd setting, where both audio and visual cues contribute to the boundary detection. The final sequence shows gurgling transitioning to speech, with audio features again providing the primary signal for segmentation.

Notably, the audio-visual segmentation (middle bar in each set) often closely matches the audio-only segmentation (bottom bar), suggesting that audio features frequently dominate the temporal boundary detection. This makes intuitive sense for events like speech, breathing, and applause that have distinct acoustic signatures but may not correspond to major visual changes.

These examples highlight the importance of processing audio in smaller segments rather than using a single global representation. The audio features are often more relevant for segmenting these videos, demonstrating the value of the fine-grained audio processing approach of CAV-MAE Sync.


Figure 6. Temporal segmentation results showing audio-visual event boundaries across different scenarios. Each row shows frame sequences with corresponding segmentation bars for visual, audio-visual, and audio-only features, along with spectrograms. Labels indicate primary events (Speech, Text, Breathing, Applause, Gurgling) manually detected in different temporal segments