

## A Appendix

In this document, we provide additional insights, experimental results, and hold other discussions on AVROBUSTBENCH. We organize all of this as follows,

1. Section A.1 describes, in great detail, the benchmark datasets we propose i.e., AUDIOSET-2C, VGG SOUND-2C, KINETICS-2C, and EPICKITCHENS-2C. We also describe the implementation details of our proposed real-world audio-visual corruptions, and show visuals of EPICKITCHENS-2C.
2. In Section A.2, we dive deep into the architectures and implementation details of all the supervised and self-supervised audio-visual models that are used for our study. We discuss the training settings of supervised models on Kinetics-Sounds, which is then used for evaluation purposes on KINETICS-2C. We also give details of the online test-time adaptation methods that are used.
3. We give a complete formulation of AV2C in Section A.3. Section A.4 has other detailed results from the main paper. We also touch upon other experiments-a subjective test on humans (Section A.5), audio-visual retrieval (Section A.6), and the recognition performance of audio-visual large language models (Section A.7).

### A.1 Proposed Benchmark: AVROBUSTBENCH

#### A.1.1 Datasets

As mentioned in the main paper, AVROBUSTBENCH consists of four benchmark audio-visual datasets, AUDIOSET-2C, VGG SOUND-2C, KINETICS-2C, and EPICKITCHENS-2C, constructed from the test sets of popular audio-visual datasets: AudioSet [8], VGGSound [3], Kinetics-Sounds [1], and Epic-Kitchens [5], respectively. These datasets span diverse domains, environments, and action categories, offering a broad and realistic evaluation suite for audio-visual recognition models. AudioSet is one of the largest audio-visual datasets in terms of training samples. It is released as YouTube URLs, and after filtering out invalid URLs, AUDIOSET-2C contains 16,742 audio-video test pairs. Each clip is roughly 10s and spans 527 classes. Due to its multilabel nature, mean average precision (mAP) is the standard evaluation metric. VGGSound consists of roughly 10s of YouTube videos spanning 309 classes, including human actions. After filtering invalid URLs, VGG SOUND-2C contains 14,046 test pairs. From Kinetics-Sounds’ test set, we construct KINETICS-2C, comprising YouTube videos capturing a diverse range of human actions. KINETICS-2C contains 3,111 clips across 32 classes, each around 10s long. EPICKITCHENS-2C is the corrupted test set of Epic-Kitchens that has 205 egocentric video clips capturing daily kitchen tasks of an average duration of 7.4 mins each. We follow the protocol, as set by [5], for action evaluation. Each action is uniquely defined by a combination of a “Verb” and a “Noun”. In the main paper, we give a summary of the number of samples and classes.

We adopt the standard evaluation protocol for robustness, as outlined in [7, 17], and introduce real-world audio-visual corruptions that are applied *simultaneously* to both modalities during testing. Each corruption type is used with a specific severity level sampled from a fixed scale (typically 1–5), ensuring consistency across evaluations. Visual corruptions are applied to every frame in the video, while audio corruptions are added directly to the video’s corresponding audio waveform.

The corruptions are chosen to reflect real-world challenges. They are designed to be *co-occurring* and *correlated*, mimicking the natural interplay of noise that might affect both modalities in deployment settings like autonomous vehicles or wearable devices. To facilitate further research, we also release the code, enabling easy reproducibility and extension of the benchmark.

#### A.1.2 Implementation of Audio-Visual Corruptions

Here, we provide the implementation details of the audio-visual corruptions. As mentioned earlier, we group the 15 corruptions, each spanning 5 severity levels, into three categories, i.e., *Digital*, *Environmental*, and *Human-Related*. In total, we propose 75 audio-visual corruptions.

- *Digital*: For visual corruptions, we adopt the exact implementations of *Gaussian*, *Impulse*, *Shot*, and *Speckle* from ImageNet-C [17]. For *Compression*, we utilize the JPEG-based

compression proposed in the same work. Throughout, we apply audio corruptions at signal-to-noise ratios (SNRs) ranging from 40 to 0 in intervals of 10, where a lower SNR corresponds to higher corruption severity. For example, a severity level of 5 indicates an SNR of 0. In the case of *Gaussian*, we generate a noise vector matching the shape of the audio signal, sample it from a standard normal distribution, scale it according to the desired SNR, and add it to the original audio waveform. For *Impulse*, we sample a salt-and-pepper noise vector based on a uniform random mask. For *Shot*, zero-mean Poisson noise is derived from the normalized audio waveform. In *Speckle*, we multiply zero-mean Gaussian noise element-wise with the audio waveform to create speckle distortions. For each corruption type, the noise is scaled using the audio signal power  $P_{sig}$  and the raw noise power  $P_n$ , both computed as the mean squared amplitude of their respective signals. The noise scaling factor  $\beta$  is computed as  $\sqrt{\frac{P_{sig}}{10^{SNR/10} * P_n}}$ , making sure that the noise meets the desired SNR, i.e., severity. Then, the scaled noise ( $\beta$ -noise) is added back to the original audio waveform. For audio *Compression*, we control the severity based on the bitrate quantization levels, computed as  $2^c$  where  $c \in [24, 16, 8, 4, 2]$ . A severity of 5 would refer to bitrate levels of 4. We split the mono waveform into fixed-size blocks of size 1024, apply an orthonormal DCT to each block, normalize, and quantize its coefficients to the required bitrate level. We then reconstruct the audio waveform via inverse DCT before concatenating the blocks.

- *Environmental*: The visual corruptions in *Snow*, *Frost*, and *Spatter* are directly taken from the implementation in [17]. For *Wind*, we use the implementation of "Motion blur", as in the same work. For the visual effects of *Rain* on video frames, we control the severity based on droplet density, scale, zoom factor, threshold, motion-blur settings, and blend weight. A monochrome rain mask is generated by sampling Gaussian noise, applying a clipped zoom to cluster droplets, and thresholding to isolate individual raindrops. We also add a tinted bluish color by expanding it into the RGB channels with custom scaling factors. Similarly, for *Underwater*, we control the severity based on Gaussian blur kernel size, red-channel attenuation, contrast reduction, and haze intensity. These are used to mimic light absorption and scattering underwater. We first reduce the red channel by the red-channel attenuation factor to simulate the color shift, then apply a Gaussian blur to soften edges as light diffuses. Additionally, the contrast is lowered via linear scaling, and a semi-transparent white haze overlay, based on the haze intensity, is blended in.

As mentioned in the main text, we borrow recorded samples from Freesound for the audio corruptions. We ensure that their sampling rates match those of the target audio and are converted to mono. Each corruption is overlaid directly onto the waveform, with its intensity precisely controlled by the specified SNR (severity), as in the case of Digital. To introduce diversity within each corruption type, we avoid using a fixed noise pattern across all audio samples. Instead, for every corruption, we randomly select one noise sample from a pool of  $N$  options, where  $N \in [15, 5, 8, 8, 8, 31]$  for *Snow*, *Frost*, *Spatter*, *Wind*, *Rain*, and *Underwater*, respectively.

- *Human-Related*: We introduce human-level corruptions that closely reflect real-world conditions. In the *Concert* setting, we adopt the "Brightness" visual effect from ImageNet-C, and overlay loud music samples from Freesound as the audio corruption, with severity controlled via the SNR. For *Smoke*, we simulate grayish visual effects by generating a Gaussian-blurred noise map, scaled by a factor to control the standard deviation and replicated across RGB channels. Corresponding audio corruptions use recorded smoke alarms and loud sirens from FreeSound. For *Crowd*, we project random human occlusions onto a video frame. The size of the occlusion denotes the severity of it. We place it at a random location, and blend it over the original frame. For an audio effect, we overlay random crowd noises from FreeSound. For *Interference*, we randomly rotate a video frame with the angle sampled from  $(-(6 \times \alpha + 5), +(6 \times \alpha + 5))$  in degrees where  $\alpha$  is the severity. So, for an  $\alpha$  of 5, a video frame could be randomly rotated between -35 to 35 degrees. We randomly silence a fraction of the audio (a higher severity denotes more silencing) between  $[0.1, 0.2, 0.3, 0.4, 0.5]$ . Throughout this group, we make sure that, for each introduced audio corruption, we have diverse noise patterns within the same task/corruption as in *Environmental*.

The full code implementation is released here: <https://github.com/sarthaxxxx/AV-C-Robustness-Benchmark/tree/master>

### 107 A.1.3 Visualizations

108 In Figure 1, we showcase sample visual corruptions from EPICKITCHENS-2C. For the full audio-  
 109 visual experience, we urge the reader to see and hear the difference in action by watching our demo  
 110 on YouTube: <https://www.youtube.com/watch?v=hYdcRO3BuY>.

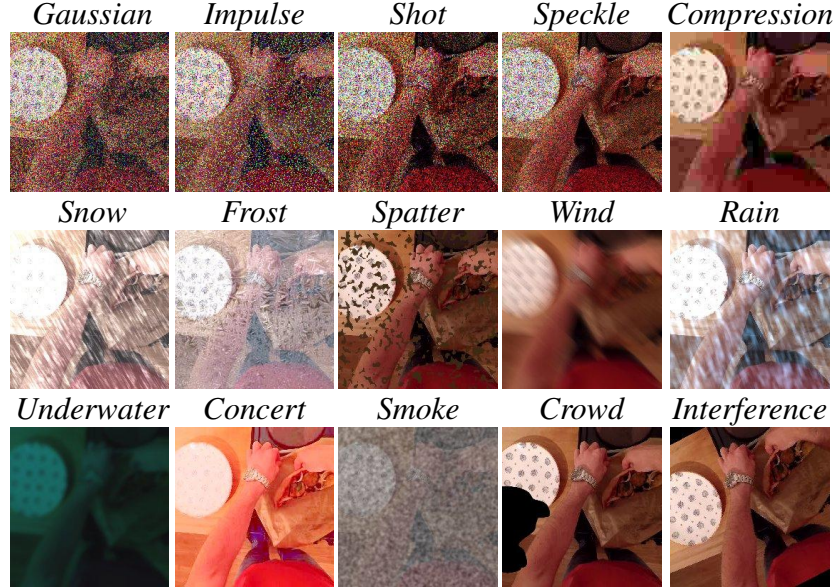


Figure 1: We sample a random video frame from Epic-Kitchens [5] and show visualizations of the proposed 15 audio-visual corruptions on it, at a severity level of 5. This constitutes EPICKITCHENS-2C.

## 111 A.2 Implementation Details

### 112 A.2.1 Models

113 On AUDIOSET-2C and VGG SOUND-2C, we directly infer using *supervised* models like UAVM  
 114 [11], CAV-MAE [13], and EquiAV [20]. UAVM employs modality-specific transformers that process  
 115 audio and video features in parallel. Audio and video features are extracted via ConvNeXt-Base  
 116 [21] backbones for spectrograms and frames, respectively, before being fed into their respective  
 117 transformers. A shared transformer is then applied twice (once per modality), and the resulting logits  
 118 are averaged, followed by a softmax to produce final predictions. CAV-MAE has separate audio  
 119 and visual encoders that process spectrograms and a randomly sampled video frame, respectively,  
 120 followed by a joint encoder trained through contrastive learning on audio-video masked tokens. Since  
 121 pre-trained weights for Kinetics-Sounds [1] are not publicly available, we adhere to the recommended  
 122 training recipes from the cited works. We then use the respective trained models for inference on  
 123 KINETICS-2C. The training details are provided later. In addition to these models, we also *self-*  
 124 *supervised* models like AudioCLIP [16], ImageBind [9], and Wav2CLIP [31]. Since our focus would  
 125 be on their respective audio and visual encoders, this would allow us to gauge their robustness to the  
 126 proposed corruptions. For experiments on EPICKITCHENS-2C, we conduct evaluations using TIM  
 127 [2] and TBN [18], as supervised models, following their official methodologies. TIM processes both  
 128 modalities through separate encoder streams before applying cross-modal attention to capture the  
 129 intricate relationships between sounds and visual actions typical in kitchen environments. We use the  
 130 same feature extraction pipeline with Omnivore [10] and VideoMAE-L [28] for visual features and  
 131 Auditory SlowFast [19] for audio features. The model encodes time intervals from each modality as  
 132 queries to identify actions occurring during specific timeframes. TBN performs mid-level fusion of  
 133 video frames and audio within temporal binding windows. We corrupt the frames and audio during  
 134 inference time and feed them directly into the model for prediction.

135 **Predictions from self-supervised models.** For AudioCLIP and ImageBind, we sample a single  
 136 frame from the video to extract  $f_v$ , while Wav2CLIP operates over a tensor of video frames. Audio

features  $f_a$  are extracted accordingly, ensuring sampling rates align with each model’s specifications. To obtain the text features  $\{f_{t,c}\}_{c=1}^C$ , where  $f_{t,c}$  is the text feature of class  $c$  out of  $C$  classes, we compute the audio-text ( $S^{a,t}$ ) and image-text logits ( $S^{v,t}$ ) as,

$$S^{a,t} = \frac{\langle f_a, f_{t,c} \rangle}{\|f_a\|_2 \cdot \|f_{t,c}\|_2}, \quad S^{v,t} = \frac{\langle f_v, f_{t,c} \rangle}{\|f_v\|_2 \cdot \|f_{t,c}\|_2} \quad (1)$$

We then compute  $\frac{S^{a,t} + S^{v,t}}{2}$  to obtain the averaged logits of class  $c$ . Applying a softmax operation over all classes yields the final likelihood of each audio-visual pair.

**Prompt Templates.** We use the default prompt templates provided with each zero-shot model. In the main paper, we also show that different prompt templates for ImageBind yield minimal improvements.

## A.2.2 Training settings of supervised models for KINETICS-2C

**CAV-MAE** consists of 11 transformer layers per modality to extract features from both audio and visual inputs. 10 frames are sampled from each clip, and one frame is randomly selected as input to the visual transformer encoder. For the audio stream, the 10 s waveform is converted into a spectrogram, which is then passed through the audio transformer encoder. During the fine-tuning phase on Kinetics-Sounds, following the CAV-MAE setup<sup>1</sup>, we freeze the pretrained visual and audio encoders from the pretrained model CAV-MAE-Scale++ and add a randomly initialized MLP classifier on top with 32 classes. The resulting fine-tuned model is treated as the source model for experiments on test-time adaptation. For input normalization, we set the dataset mean to -5.081 and the standard deviation to 4.4849, following [32]. We use a learning rate of 1e-4, a batch size of 48, and train for 10 epochs.

During pretraining, **EquiAV** processes 10 s video clips, where the visual stream involves sampling frames and applying spatial augmentations, while the audio stream is converted into spectrograms and undergoes temporal augmentations. We use the AudioSet-2M pretrained model as the backbone and add newly initialized layer for the fine-tuning stage. We fine-tune it on the Kinetics-Sounds training set with both audio and visual data under multi-modal mode. The model is trained for a maximum of 50 epochs with a learning rate of 1e-4. For input normalization, we use the same dataset mean and standard deviation as in CAV-MAE.

In **UAVM**, audio features are extracted using an AudioSet-2M pretrained ConvNeXt-Base, while visual features are obtained using the official ImageNet-pretrained ConvNeXt-Base. UAVM model consists of three modality-specific Transformer layers, followed by three shared Transformer layers. For each input sample, a separate forward pass is performed for the audio and visual modalities, and the predictions from the two passes are averaged to produce the final fused prediction. We use a learning rate of 1e-4, a batch size of 144, and train for 10 epochs. During training, each iteration uses only one modality, with a 50% chance of selecting either audio or video.

## A.2.3 Online Test-Time Adaptation (TTA) settings

**Source [13]** Following the audio-visual TTA protocol set by [32], we use CAV-MAE [13] as the source model. For experiments on VGG SOUND-2C and KINETICS-2C, as the initialization, we use pre-trained weights from VGG Sound and Kinetics-Sounds (as mentioned in A.2.2), respectively, and do a direct inference on a test-batch.

**TENT [29]** Following [29, 32], all the LayerNorm parameters of the CAV-MAE audio, visual, and joint encoder are updated by minimizing the Shannon entropy [25] of model predictions. For VGG SOUND-2C, we use a batch size of 32 and USE with an Adam optimizer using a learning rate of 1e-4. For KINETICS-2C, we use a batch size of 16 instead.

**RPL [24]** The LayerNorm parameters of the CAV-MAE model are updated using the generalized cross-entropy loss. We use batch sizes of 32 and 16 for VGG SOUND-2C and KINETICS-2C, respectively, with the Adam optimizer and a learning rate of 1e-4.

**EATA [22]** On VGG SOUND-2C and KINETICS-2C, we use a batch size of 16 for an Adam optimizer with a learning rate of 1e-4. The entropy threshold is set  $0.4 \times \log(C)$ , where  $C$  refers to the number of

<sup>1</sup><https://github.com/yuangongnd/cav-mae>

class labels. Since the source data is unavailable, we do not use the Fisher regularization to minimize forgetting of source domain knowledge.

**SAR [23]** We use batch sizes of 32 and 16 for VGG SOUND-2C and KINETICS-2C, respectively. The LayerNorm parameters of CAV-MAE are updated using the Adam optimizer with a fixed learning rate of  $1e-4$ . For stable entropy minimization, we adopt the same confidence threshold as in EATA [22], while a threshold of 0.2 is used for model recovery. The exponential moving average (EMA) coefficient for model predictions is set to 0.9.

**READ [32]** Following their original implementation, the QKV parameters of CAV-MAE’s joint encoder are self-adapted based on a confidence-aware and balancing loss function. A confidence threshold of  $\frac{1}{e}$  is used. We use the Adam optimizer with a learning rate of  $1e-4$ , and batch sizes of 64 and 16 for VGG SOUND-2C and KINETICS-2C, respectively.

**SuMi [15]** LayerNorm parameters are updated based on Adam using learning rates of  $1e-4$  and  $1e-5$  for KINETICS-2C and VGG SOUND-2C respectively. We use a batch size of 16 for both. The multimodal threshold [22] is set to  $0.4 \times \log(C)$ . As per their recommendation, a mutual information loss is applied for every half iteration. All other dataset-specific hyperparameters are set based on the original work.

### A.3 AV2C - Our proposed TTA method

Our proposed online TTA framework, AV2C, consists of two key areas. The first area emphasizes efficient audio-visual cross-modal fusion at test-time. Let  $f_a$  and  $f_v$  represent the audio and visual embeddings, respectively, obtained from the modality-specific encoders of the CAV-MAE source model [13]. To enable fine-grained integration across modalities at the token level, we concatenate these embeddings to form a joint representation:  $f_{av} = [f_a; f_v]$

Inspired by READ [32], a simple proposal for good, reliable, and on-the-fly fusion is to modulate the attention parameters of the joint-encoder, which dynamically re-weights modality contributions, enabling more robust integration under distribution shifts. Formally speaking, let  $w_q$ ,  $w_k$ , and  $w_v$  denote the weight matrices of the query, key, and value parameters of the attention block in the joint encoder. And, let  $b_q$ ,  $b_k$ , and  $b_v$  be their corresponding biases. So, the attention matrices are,

$$\mathcal{Q} = f_{av}W_q + b_q \quad (2)$$

$$\mathcal{K} = f_{av}W_k + b_k \quad (3)$$

$$\mathcal{V} = f_{av}W_v + b_v \quad (4)$$

Throughout,  $\mathcal{Q}$ ,  $\mathcal{K}$ , and  $\mathcal{V}$  are adapted at test-time with all other model parameters being frozen and fixed to the default source weights. With  $\mathcal{Q}$ ,  $\mathcal{K}$ , and  $\mathcal{V}$  being adaptive, both self- and cross-attention are computed at the token level to dynamically capture and integrate modality-specific and modality-shared information, enabling robust fusion under distribution shifts.

However, under simultaneous distributional shifts in both modalities, the input token quality may degrade significantly, resulting in unreliable attention computations and elevated model uncertainty. To address this, in the second area, we adapt the attention parameters  $\mathcal{Q}$ ,  $\mathcal{K}$ , and  $\mathcal{V}$  at test time by selectively updating them based on confident predictions. Inspired by the loss formulation in [22], we apply entropy minimization but only on low-entropy (i.e., high-confidence) samples. This selective update strategy ensures stable and reliable adaptation without propagating noise from uncertain predictions.

We minimize a weighted Shannon entropy [25] of model predictions, as an unsupervised objective. That is, the optimization based on the entropy  $H(x)$  at time-step  $t$  is,

$$\operatorname{argmin}_{\theta} - \widehat{\eta(x)} H(x) \quad (5)$$

$$= \operatorname{argmin}_{\theta} - \widehat{\eta(x)} \sum_{c \in C} p(y_t = c | x) \log p(y_t = c | x) \quad (6)$$

where,  $C$  is the complete set of classes and  $p(y_t|x)$  is the probability of output logits.  $\widehat{\eta(x)}$  is a penalty on the entropy of model predictions and  $\theta$  refers to the set of  $\mathcal{Q}$ ,  $\mathcal{K}$ , and  $\mathcal{V}$  as the model parameters.

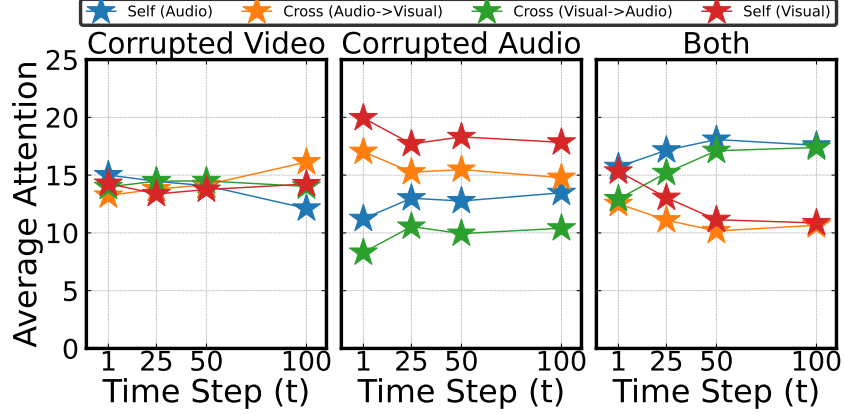


Figure 2: Over time steps ( $t$ ) during online TTA, AV2C begins to mitigate the modality bias. VGGSound [3] is audio-dominant, i.e., audio has task-specific information. AV2C begins to put more self-attention and cross-attention weights on audio. Average attention weights are computed across 12 heads from 1 block of CAV-MAE’s joint encoder for a batch size of 64. The numbers indicate averaged attention, scaled by 10,000. We show *Gaussian* on VGGSound-2C.

226 To penalize high-entropy samples, we first compute an entropy-based threshold  $\eta_{ent}(x)$  as,

$$\eta_{ent}(x) = \frac{1}{\exp(H(x) - H_1)} \cdot \mathbb{1}_{\{H(x) < H_1\}} \quad (7)$$

227 With  $H(x)$  being the entropy and  $H_1$  being a fixed threshold, we see that high-entropy samples are  
 228 penalized more and excluded from the adaptation process. Essentially, samples with low-entropy  
 229 predictions are more reliable and contribute more effectively to the audio-visual test-time adaptation  
 230 process. However, simply using all low-entropy samples may introduce redundancy, as similar inputs  
 231 often yield similar gradients, which could hurt adaptation. To promote diversity among the selected  
 232 samples, we introduce a filtering mechanism. Specifically, we maintain a running exponential moving  
 233 average of the model’s predicted class probabilities across recent batches, denoted as  $\hat{k}$ , up to the  
 234 current time step  $t$ . For each incoming test sample, we compute the cosine similarity between its  
 235 prediction and  $\hat{k}$  to assess redundancy and retain only sufficiently dissimilar (i.e., diverse) low-entropy  
 236 samples for adaptation. That is,

$$\eta_d(x) = \mathbb{1}_{\{\text{sim}(p(y_t|x), \hat{k}) < \rho\}}(x) \quad (8)$$

237 where,  $\text{sim}$  refers to the cosine similarity and  $\rho$  is a threshold. Overall,  $\widehat{\eta}(x) = \eta_{ent}(x) \cdot \eta_d(x)$ .

238 Overall, our proposed TTA method AV2C, is a simple audio-visual TTA approach, inspired by  
 239 [32, 22]. It focuses on performing on-the-fly cross-modal fusion with the  $\mathcal{Q}$ ,  $\mathcal{K}$ , and  $\mathcal{V}$  weights being  
 240 updated based on reliable multimodal samples. Our goal is to push and give new directions for  
 241 using AVROBUSTBENCH to understand and to inspire the development of more robust adaptation  
 242 strategies in real-world settings. In our experiments, we set  $H_1$  to  $0.4 \times \log(C)$ , following EATA [22],  
 243 and  $\rho$  is set to 0.05. We update the attention parameters with a learning rate of  $1e-4$  using the Adam  
 244 update rule, with a batch size of 32.

245 • **AV2C minimizes modality-bias.** From Figure 2, we observe that on VGGSound-2C, which  
 246 contains dominant task-specific audio cues [3] and with both modalities corrupted, the model  
 247 gradually increases its attention to the audio tokens to perform better recognition.

## 248 A.4 Additional Results

### 249 A.4.1 Corruption specific results on AUDIOSET-2C, VGGSound-2C, KINETICS-2C, and 250 EPICKITCHENS-2C

251 In Tables 1 and 2, we report the direct inference results of supervised (UAVM [12], CAV-MAE [13],  
 252 EquiAV [20], TBN [18], and TIM [2]) and self-supervised models (AudioCLIP [16], ImageBind [9],  
 253 and WavCLIP [31]) at a corruption-specific level/task.

Table 1: Metrics of audio-visual models evaluated on AUDIOSET-2C, VGGSound-2C, and KINETICS-2C at a severity level of 5. For AUDIOSET-2C, we report the mean of  $MAP$ , while for VGGSound-2C and KINETICS-2C, we report the accuracy ( $Acc$ ).

Model		Gaussian	Impulse	Shot	Speckle	Compression	Snow	Frost	Spatter	Wind	Rain	Underwater	Concert	Smoke	Crowd	Interference	Mean
AUDIOSET-2C	UAVM [11]	27.95	28.19	28.67	30.71	23.57	26.46	35.06	35.58	33.96	27.54	31.63	37.22	30.36	39.02	42.86	31.91
	CAV-MAE [13]	27.39	28.55	27.23	27.85	13.81	26.98	35.92	37.06	38.70	28.23	33.10	38.24	31.76	40.06	44.68	31.97
	EquiAV [20]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	AudioCLIP [16]	10.27	9.71	7.14	7.48	8.98	8.88	12.14	13.13	18.13	11.86	14.63	8.97	9.05	16.93	23.61	12.06
	ImageBind [9]	6.34	6.79	6.75	9.24	7.85	8.73	10.45	12.21	10.91	9.51	9.58	12.08	9.58	12.84	16.59	9.96
	WavCLIP [31]	0.89	0.91	0.92	1.62	1.58	1.17	1.61	1.85	2.29	1.05	1.69	1.55	1.25	3.38	4.40	1.74
VGGSound-2C	UAVM [11]	13.77	24.53	14.96	24.20	8.74	15.35	29.67	40.42	36.16	23.01	28.96	38.84	24.74	40.82	46.91	27.41
	CAV-MAE [13]	20.16	15.31	19.28	25.48	20.24	31.24	41.38	44.59	47.15	32.69	32.40	44.44	33.78	47.46	51.11	33.78
	EquiAV [20]	42.42	46.26	64.23	71.10	45.58	68.50	73.66	76.75	78.27	71.02	85.28	57.24	59.54	60.59	61.85	64.15
	AudioCLIP [16]	6.71	6.11	7.25	8.61	8.25	9.29	10.91	12.55	13.48	12.27	11.46	17.34	8.88	17.90	16.16	11.14
	ImageBind [9]	8.95	10.54	8.98	10.32	1.75	4.96	11.74	14.90	12.60	3.97	7.25	9.91	12.78	10.07	24.84	10.24
	WavCLIP [31]	0.56	0.59	0.59	3.92	3.13	1.89	3.92	5.74	7.38	1.17	4.72	4.96	3.63	12.86	19.82	4.99
KINETICS-2C	UAVM [11]	37.15	33.24	35.62	37.94	34.56	31.23	54.10	60.73	62.56	48.07	46.71	58.29	41.37	69.21	70.08	48.06
	CAV-MAE [13]	51.34	48.82	51.27	46.90	44.88	47.88	59.97	63.16	68.76	58.54	61.51	66.80	48.15	74.81	79.44	58.15
	EquiAV [20]	55.29	55.26	50.91	55.29	55.03	58.15	64.22	69.24	71.81	67.05	62.36	72.36	59.47	79.30	80.23	63.73
	AudioCLIP [16]	13.82	12.66	16.55	21.05	19.25	19.99	22.66	27.07	26.33	25.65	23.98	34.97	19.70	36.87	33.04	23.57
	ImageBind [9]	26.97	29.93	27.32	30.95	6.81	14.43	22.37	37.03	33.46	13.05	23.88	25.91	32.37	29.73	48.22	26.82
	WavCLIP [31]	4.50	4.37	4.85	15.59	12.21	8.52	17.49	21.99	25.72	7.68	19.09	20.28	12.83	38.09	45.55	17.25

Table 2: Metrics of TBN [18] and TIM [2] evaluated on EPICKITCHENS-2C at a severity level of 5.

Model	Gaussian	Impulse	Shot	Speckle	Compression	Snow	Frost	Spatter	Wind	Rain	Underwater	Concert	Smoke	Crowd	Interference	Mean
TBN [18](Noun)	23.32	22.26	23.46	16.30	19.91	21.68	28.05	25.67	36.05	26.16	20.42	27.95	17.22	35.40	41.32	25.68
TBN [18](Verb)	54.45	53.92	53.88	42.80	38.95	51.26	55.75	54.77	59.75	52.85	50.22	53.03	44.89	58.80	60.34	52.38
TIM [2](Noun)	42.58	50.77	53.50	58.08	50.02	45.81	43.34	54.85	41.16	50.61	43.73	61.59	45.35	44.30	54.71	49.36
TIM [2](Verb)	65.13	68.72	70.57	72.76	68.89	63.02	60.37	71.62	62.62	65.71	65.74	74.13	60.63	63.00	65.37	66.55

#### 254 A.4.2 AUDIOSET-2C AND KINETICS-2C - Relative robustness for different severities

255 In Figures 3 and 4, we illustrate the effect of corruption severity on relative robustness on  
 256 AUDIOSET-2C and KINETICS-2C. Our findings remain the same-model robustness declines with an  
 257 increase in corruption severity.

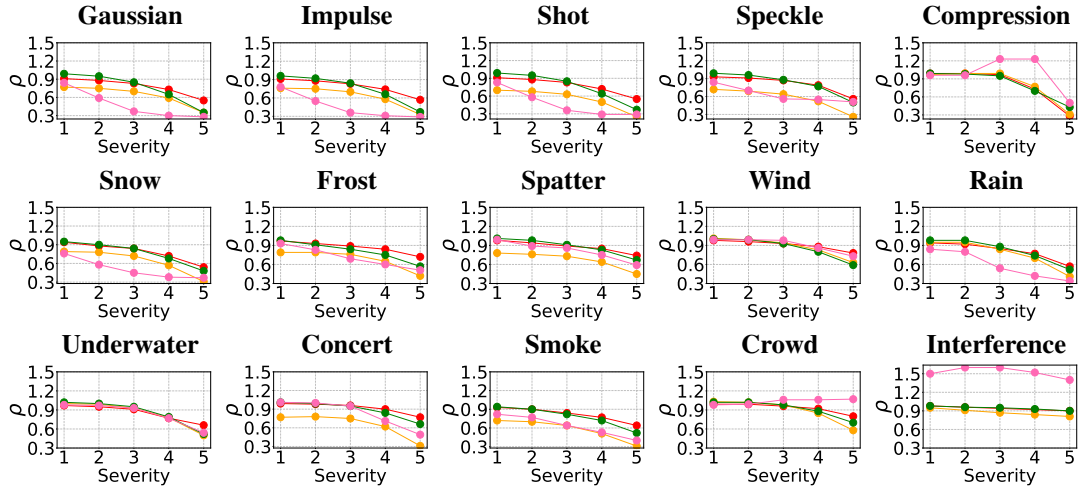


Figure 3: Relative robustness ( $\rho$ ) on AUDIOSET-2C. We show the performance of **CAV-MAE**, **AudioCLIP**, **ImageBind**, and **Wav2CLIP**. The x-axis denotes corruption severity, and the y-axis denotes  $\rho$ .



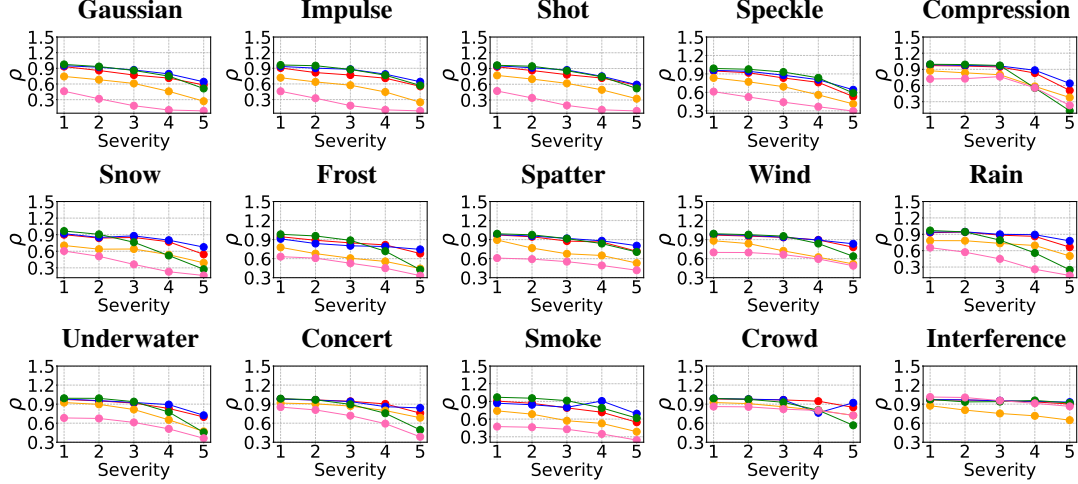


Figure 4: Relative robustness ( $\rho$ ) on KINETICS-2C. We show the performance of **CAV-MAE**, **EquiAV**, **AudioCLIP**, **ImageBind**, and **Wav2CLIP**. The x-axis denotes corruption severity, and the y-axis denotes  $\rho$ .

#### 258 A.4.3 EPICKITCHENS-2C - Relative robustness for different severities

259 On similar lines, we illustrate the effect of corruption severity on the relative robustness of TBN [18]  
 260 and TIM [2] on EPICKITCHENS-2C in Figure 5.

Table 3: *In a continual setup, with no model reset, the performance gap between mean accuracy by TTA baselines and the source model’s accuracy on VGGSound (65.50%) and Kinetics-Sounds (88.10%), widens drastically. CAV-MAE [13] is the source model initialized by VGGSound/Kinetics-Sounds weights’. We report mean accuracy (%) on VGGSound-2C (top) and KINETICS-2C (bottom) at a severity of 5. Source denotes the direct inference of CAV-MAE.*

TTA Method	Gaussian	Impulse	Shot	Speckle	Compression	Snow	Frost	Spatter	Wind	Rain	Underwater	Concert	Smoke	Crowd	Interference	Mean
VGGSound-2C																
Source [13]	20.39	23.73	20.72	25.34	17.26	25.07	46.82	48.46	50.17	29.89	42.19	47.61	32.93	47.71	54.88	35.54
TENT [29]	1.83	0.34	0.33	0.39	0.63	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.45
READ [32]	1.76	0.33	0.33	0.38	17.59	3.09	4.30	43.39	45.33	1.03	26.54	32.06	23.56	28.50	36.98	17.68
SuMi [15]	22.24	22.90	21.83	23.29	16.79	12.97	44.99	47.30	48.70	15.74	40.68	46.46	28.57	46.57	54.51	32.90
AV2C	38.27	38.80	40.33	34.89	21.77	41.66	49.23	50.11	50.88	44.85	46.15	49.30	46.22	51.80	52.67	43.80
KINETICS-2C																
Source [13]	51.34	48.82	51.27	46.90	44.88	47.88	59.97	63.16	68.76	58.54	61.51	66.80	48.15	74.81	79.44	58.15
TENT [29]	42.45	11.31	5.65	4.26	3.60	3.33	3.59	3.14	3.17	3.14	3.18	3.14	3.14	3.18	3.17	6.88
READ [32]	53.18	53.01	54.39	44.25	37.36	35.21	43.02	37.18	33.77	14.27	20.08	20.05	13.84	23.88	24.66	33.85
SuMi [15]	49.07	46.46	40.06	32.71	34.27	22.99	11.44	4.27	3.69	3.17	3.24	3.17	3.17	3.17	3.3	17.79
AV2C	51.71	53.50	54.45	52.87	35.61	52.25	65.75	61.13	69.92	61.21	61.04	68.00	55.00	75.12	78.64	59.75

#### 261 A.4.4 Continual Online Test-Time Adaptation

262 Online TTA focuses on adapting a pre-trained source model to a single target domain at a time.  
 263 However, this assumption is often unrealistic in dynamic, real-world settings where models encounter  
 264 sequences of non-stationary and evolving target domains with rapid shifts in test distributions and no  
 265 knowledge of task boundaries [30]. In such a case, there are two potential challenges. The first is  
 266 *catastrophic forgetting* [14]. Due to continual model parameter updates on long sequences of tasks  
 267 involving unlabeled data of different distributions, there is a long-term loss of the model’s source  
 268 knowledge. The second challenge is *error accumulation*. As updates happen on noisy test data, errors  
 269 in early adaptation steps can propagate and compound over time, leading to significant degradation in  
 270 performance.



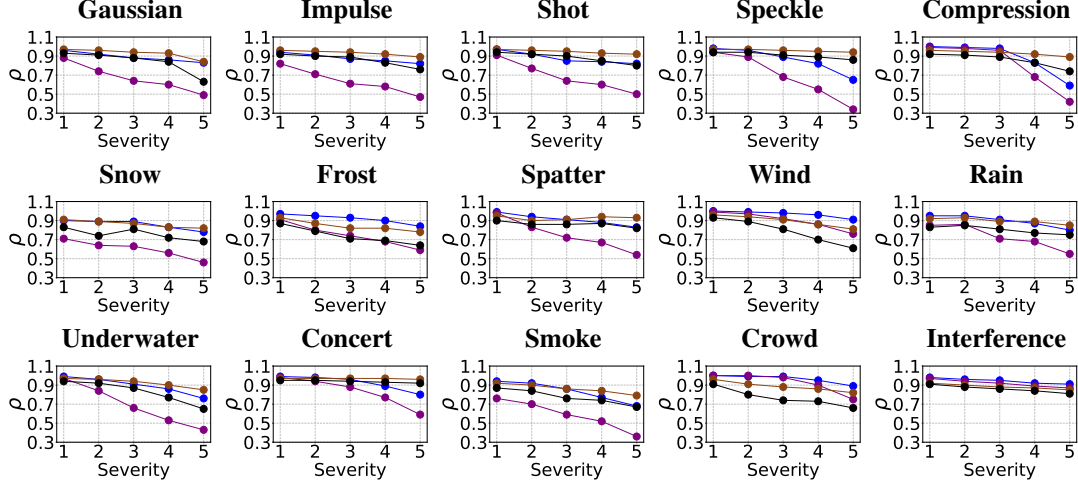


Figure 5: Relative robustness ( $\rho$ ) on Epic-Kitchens-2C (EPICKITCHENS-2C). We show **TBN (Noun)**, **TBN (Verb)**, **TIM (Noun)**, and **TIM (Verb)**. The x-axis denotes corruption severity; the y-axis denotes  $\rho$ .

271 In this section, we extend our study of online TTA to a continual setting where the CAV-MAE source  
 272 model is not reset at any point in time or after any domain, and being continually fine-tuned to  
 273 the tasks. We present the results in Table 3. Experiments are performed on VGG SOUND-2C and  
 274 KINETICS-2C.

## 275 A.5 Subjective Evaluations - Humans are very effective in recognizing corrupted audios and 276 videos

277 **Motivation.** Geirhos et al. [6] demonstrate a notable gap between human and model robustness  
 278 on noisy images. From an audio-visual standpoint, humans naturally integrate cross-modal cues  
 279 to interpret and learn from their surroundings [26]. We bridge the two ideas to study this from a  
 280 subjective point of view.

281 **Setup and Participants.** We recruited 30 volunteers from diverse backgrounds, with most participants  
 282 falling within the 18–50 age range. This recognition study aimed to evaluate the effectiveness of  
 283 AVROBUSTBENCH and investigate human performance under severe audio and visual distributional  
 284 shifts. The central question we sought to answer was: Can humans still reliably recognize when both  
 285 modalities are corrupted? If so, this underscores the importance of developing models that are not  
 286 only robust but also adaptive to the challenges of an open and dynamic world.

287 To begin our experiment, we designed a user-friendly interface. From the VGG SOUND-2C dataset,  
 288 we manually sampled 30 challenging videos with each featuring overlaid corrupted audio and drawn  
 289 from any of the 15 proposed audio-visual corruptions at a severity level 5. We choose a small subset  
 290 to not overburden a participant. For each instance, each participant was shown a corrupted video  
 291 alongside a set of 20 plausible labels and asked to select the one that best described the action  
 292 depicted. We did not show all the 309 labels since that would have made it even more challenging.  
 293 Each participant, on average, took about 20–30 seconds to identify the action in a video.

294 **Results.** We discuss the results here. Averaged across all the participants, the reported human  
 295 accuracy was  $\sim 89\%$ . Qualitatively, we observed that participants found *Digital*-ly corrupted videos  
 296 slightly difficult to recognize. The participants did mention that, depending on the video, they relied  
 297 on the audio or visual cues to identify an action. Likely, since there is a pixel-level and frequency-level  
 298 disturbance, this hindered human recognition. Similar to our findings, videos with *Human-related*  
 299 corruptions were very easy to identify. These corruption types are often observed and are familiar by  
 300 humans. The major takeaway from these experiments is that human perception remains robust under  
 301 many real-world corruptions. This highlights the importance of designing audio-visual models that  
 302 can similarly adapt to and withstand such conditions in open-world environments.

## 303 A.6 Audio-Visual Retrieval - Cross-modal correspondence is hampered drastically

Table 4: *Models struggle to maintain cross-modal correspondence under AV corruptions at test-time.* Numbers report retrieval recalls (R@1, R@5, R@10) for Visual→Audio (left) and Audio→Visual (right) on AUDIOSET-2C and VGG SOUND-2C subsets. We report the mean metrics across the proposed 15 tasks, computed at a severity of 5. "Clean" refers to the original test sets. We use CAV-MAE [13] as the pre-trained model.

Visual→Audio	AUDIOSET-2C			VGG SOUND-2C		
	R@1	R@5	R@10	R@1	R@5	R@10
Clean	16.63	35.71	45.15	12.62	28.48	37.00
Across 15 tasks	0.97	2.94	3.33	1.51	4.57	6.60

Audio→Visual	AUDIOSET-2C			VGG SOUND-2C		
	R@1	R@5	R@10	R@1	R@5	R@10
Clean	13.41	29.42	38.36	12.76	28.43	36.36
Across 15 tasks	0.91	2.69	4.06	2.24	6.72	10.24

304 While CAV-MAE [13] claims to learn rich joint audio-visual representations, we now investigate  
305 whether such a supervised pre-trained model can effectively capture audio-visual correspondences  
306 under real-world distributional shifts at test-time. Here, we study audio-visual retrieval, which relies  
307 on semantic alignment between audio and visual content for cross-modal search. Following the setup  
308 from CAV-MAE, we uniformly sample pairs from AUDIOSET-2C and VGG SOUND-2C, creating  
309 subsets of 1,725 and 1,545 samples, respectively. Retrieval performance is evaluated using cosine  
310 similarity between the modality representations and reported as retrieval recall at ranks 1, 5, and  
311 10. The results of audio→visual and visual→audio are reported in Table 4. Given a corrupted  
312 query modality, we retrieve the other modality. We report metrics on a clean subset, which may  
313 slightly differ from the original CAV-MAE due to variations in the test subsets and YouTube URL  
314 availability. However, the main takeaway lies in the large recall gap between clean subsets and the  
315 average performance. On AUDIOSET-2C and VGG SOUND-2C, R@1 drops by 15.66% and 12.5%  
316 respectively.

Table 5: Metrics of audio-visual LLMs evaluated on VGG SOUND-2C and KINETICS-2C at a severity level of 5 for action recognition. We report the accuracy (*Acc*) on each task. For comparison, we also provide the performances on the clean/original test sets.

Model		Gaussian	Impulse	Shot	Speckle	Compression	Snow	Frost	Splatter	Wind	Rain	Underwater	Concert	Smoke	Crowd	Interference	Clean
VGG SOUND-2C	VideoLLaMA-2 [4]	20.73	38.49	14.81	42.70	39.91	18.78	28.93	38.89	39.62	24.55	30.03	43.52	25.41	50.01	50.99	55.80
	PandaGPT [27]	3.90	7.00	3.09	7.71	6.85	2.04	5.49	5.39	4.91	3.18	2.68	5.61	3.61	7.50	9.13	11.87
KINETICS-2C	VideoLLaMA-2 [4]	21.67	24.94	24.76	42.91	53.84	48.41	46.77	60.95	42.78	55.35	44.81	66.44	18.00	66.83	69.78	76.37
	PandaGPT [27]	6.94	7.68	7.3	10.67	9.19	6.36	8.16	10.67	9.1	6.27	6.72	15.30	7.20	13.28	16.19	22.24

## 317 A.7 Robustness of Audio-Visual LLMs

318 Given the success of Multimodal Large Language Models (MLLMs) [33] in various understanding  
319 tasks, we touch upon and explore their robustness on our proposed audio-visual datasets. Specifically,  
320 we use Audio-Visual LLMs (AVLLMs) i.e. VideoLLaMA-2 [4] and PandaGPT [27] for the audio-  
321 visual recognition task on VGG SOUND-2C and KINETICS-2C.

322 The evaluation approach of these multimodal LLMs differs slightly from the *supervised* and *self-*  
323 *supervised* models discussed in the main paper. These MLLMs take audio-visual and a text query  
324 input. For example, we prompt the model with : "Which class of VGG Sound does this video belong  
325 to?" The model generates a textual output, which we compare against class labels using cosine  
326 similarity. To do this, we encode both the predicted output and the class labels using the CLIP text  
327 encoder and compute similarity scores. The highest similarity label is considered the predicted label  
328 for this specific audio-visual pair.

329 Since KINETICS-2C has 32 labels, we use the following prompt during inference with MLLMs:  
330 "Which of the following actions best describe the content of this video? Choose one from the list  
331 below: [labels]. " The model generate a textual response as output. Similarly, we use the text encoder  
332 in CLIP to compute the cosine similarity between predicted and ground-truth labels in the embedding  
333 space to calculate the accuracy.

334 The results are shown in Table 5. Consistent with findings from supervised and self-supervised models,  
 335 we observe AVLLMs show large performance degradation under audio and visual distributional shifts.  
 336 While effective prompting techniques or other strategies can be explored, we leave that for future  
 337 work.

## 338 References

- 339 [1] Arandjelovic, R., Zisserman, A.: Look, listen and learn. In: Proceedings of the IEEE interna-  
 340 tional conference on computer vision. pp. 609–617 (2017)
- 341 [2] Chalk, J., Huh, J., Kazakos, E., Zisserman, A., Damen, D.: Tim: A time interval machine for  
 342 audio-visual action recognition. In: Proceedings of the IEEE/CVF Conference on Computer  
 343 Vision and Pattern Recognition. pp. 18153–18163 (2024)
- 344 [3] Chen, H., Xie, W., Vedaldi, A., Zisserman, A.: Vggsound: A large-scale audio-visual dataset. In:  
 345 ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing  
 346 (ICASSP). pp. 721–725. IEEE (2020)
- 347 [4] Cheng, Z., Leng, S., Zhang, H., Xin, Y., Li, X., Chen, G., Zhu, Y., Zhang, W., Luo, Z., Zhao,  
 348 D., Bing, L.: Videollama 2: Advancing spatial-temporal modeling and audio understanding in  
 349 video-llms. arXiv preprint arXiv:2406.07476 (2024), <https://arxiv.org/abs/2406.07476>
- 350 [5] Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D.,  
 351 Munro, J., Perrett, T., Price, W., Wray, M.: Scaling egocentric vision: The epic-kitchens dataset.  
 352 In: European Conference on Computer Vision (ECCV) (2018)
- 353 [6] Geirhos, R., Janssen, D.H., Schütt, H.H., Rauber, J., Bethge, M., Wichmann, F.A.: Comparing  
 354 deep neural networks against humans: object recognition when the signal gets weaker. arXiv  
 355 preprint arXiv:1706.06969 (2017)
- 356 [7] Geirhos, R., Temme, C.R., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation  
 357 in humans and deep neural networks. *Advances in neural information processing systems* **31**  
 358 (2018)
- 359 [8] Gemmeke, J.F., Ellis, D.P., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M.,  
 360 Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events. In: 2017 IEEE  
 361 international conference on acoustics, speech and signal processing (ICASSP). pp. 776–780.  
 362 IEEE (2017)
- 363 [9] Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K.V., Joulin, A., Misra, I.: Imagebind:  
 364 One embedding space to bind them all. In: Proceedings of the IEEE/CVF conference on  
 365 computer vision and pattern recognition. pp. 15180–15190 (2023)
- 366 [10] Girdhar, R., Singh, M., Ravi, N., Van Der Maaten, L., Joulin, A., Misra, I.: Omnivore: A single  
 367 model for many visual modalities. In: Proceedings of the IEEE/CVF conference on computer  
 368 vision and pattern recognition. pp. 16102–16112 (2022)
- 369 [11] Gong, Y., Liu, A.H., Rouditchenko, A., Glass, J.: Uavm: Towards unifying  
 370 audio and visual models. *IEEE Signal Processing Letters* **29**, 2437–2441 (2022).  
 371 <https://doi.org/10.1109/LSP.2022.3224688>
- 372 [12] Gong, Y., Liu, A.H., Rouditchenko, A., Glass, J.: Uavm: Towards unifying audio and visual  
 373 models. *IEEE Signal Processing Letters* **29**, 2437–2441 (2022)
- 374 [13] Gong, Y., Rouditchenko, A., Liu, A.H., Harwath, D., Karlinsky, L., Kuehne, H., Glass, J.R.:  
 375 Contrastive audio-visual masked autoencoder. In: The Eleventh International Conference on  
 376 Learning Representations (2023), <https://openreview.net/forum?id=QPtMRyk5rb>
- 377 [14] Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation  
 378 of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6211  
 379 (2013)

- [15] Guo, Z., Jin, T.: Smoothing the shift: Towards stable test-time adaptation under complex multimodal noises. In: The Thirteenth International Conference on Learning Representations (2025)
- [16] Guzhov, A., Raue, F., Hees, J., Dengel, A.: Audioclip: Extending clip to image, text and audio. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 976–980. IEEE (2022)
- [17] Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2019)
- [18] Kazakos, E., Nagrani, A., Zisserman, A., Damen, D.: Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5492–5501 (2019)
- [19] Kazakos, E., Nagrani, A., Zisserman, A., Damen, D.: Slow-fast auditory streams for audio recognition. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 855–859. IEEE (2021)
- [20] Kim, J., Lee, H., Rho, K., Kim, J., Chung, J.S.: Equiav: leveraging equivariance for audio-visual contrastive learning. arXiv preprint arXiv:2403.09502 (2024)
- [21] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)
- [22] Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., Tan, M.: Efficient test-time model adaptation without forgetting. In: International conference on machine learning. pp. 16888–16905. PMLR (2022)
- [23] Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., Tan, M.: Towards stable test-time adaptation in dynamic wild world. arXiv preprint arXiv:2302.12400 (2023)
- [24] Rusak, E., Schneider, S., Pachitariu, G., Eck, L., Gehler, P., Bringmann, O., Brendel, W., Bethge, M.: If your data distribution shifts, use self-learning. arXiv preprint arXiv:2104.12928 (2021)
- [25] Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal **27**(3), 379–423 (1948)
- [26] Smith, L., Gasser, M.: The development of embodied cognition: Six lessons from babies. Artificial life **11**(1-2), 13–29 (2005)
- [27] Su, Y., Lan, T., Li, H., Xu, J., Wang, Y., Cai, D.: Pandagpt: One model to instruction-follow them all. arXiv preprint arXiv:2305.16355 (2023)
- [28] Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in neural information processing systems **35**, 10078–10093 (2022)
- [29] Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: International Conference on Learning Representations (2021)
- [30] Wang, Q., Fink, O., Van Gool, L., Dai, D.: Continual test-time domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7201–7211 (2022)
- [31] Wu, H.H., Seetharaman, P., Kumar, K., Bello, J.P.: Wav2clip: Learning robust audio representations from clip. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2022)
- [32] Yang, M., Li, Y., Zhang, C., Hu, P., Peng, X.: Test-time adaptation against multi-modal reliability bias. In: The Twelfth International Conference on Learning Representations (2024)
- [33] Zhang, D., Yu, Y., Dong, J., Li, C., Su, D., Chu, C., Yu, D.: Mm-llms: Recent advances in multimodal large language models. arXiv preprint arXiv:2401.13601 (2024)