Supplementary Material: Audiovisual Masked Autoencoders

Mariana-Iuliana Georgescu[1, 2*†] Eduardo Fonseca[1] Radu Tudor Ionescu[2]
Mario Lucic[1] Cordelia Schmid[1] Anurag Arnab[1*]
[1]Google Research [2]University of Bucharest

In this supplementary material, we include additional experimental details and results. We include additional ablation studies and evaluation in Sec. A, details about our experimental hyperparameters in Sec. B and qualitative visualisations in Sec. C.

A. Additional Experiments and Ablation Studies

This section presents additional experiments and ablation studies and evaluations of our model. Unless otherwise stated, the experiments are performed using a ViT-Base backbone, pretrained for 400 epochs on VGGSound, using the "Separate" encoding and "Shared" decoding strategies.

A.1. Audiovisual event localisation

In Sec. 4.3 of the main paper, using our learned representations we obtain state-of-the-art results on three downstream classification tasks. To show the capabilities of our audiovisual representations in a different downstream task, in Tab. A1 we evaluate on the "Supervised Event Localisation" task proposed by [17] using a ViT-Base backbone.

We consider two models, one pretrained on VGGSound for 800 epochs, and another pretraind on AudioSet for 80 epochs. These two models are pretrained for approximately the same number of iterations as AudioSet is about 10 times larger than VGGSound.

To our knowledge, we outperform the best method (concurrent work) on this task, when pretraining on either VGGSound or AudioSet. We observe that pretraining on VGGSound learns better audiovisual representations overall for this dataset.

A.2. Pretraining methods for MBT

Our main contribution is an audiovisual, self-supervised pretraining method. To show the benefit of our pretraining, for downstream finetuning we used the same model architecture as the current SOTA (MBT [13]).

Table A1: Supervised event localisation accuracy on the AVE dataset [17]. We outperform prior work when using a ViViT-Base model, and pretraining on either VGGSound or AudioSet for the equivalent number of itrations (since AudioSet is approximately 10 times larger than VGGSound.

	Audio-only	Video-only	Audiovisual
Senocak *et al.* [15]	79.1	76.1	87.8
Ours (Audioset, 80 epochs)	**82.3**	77.6	88.6
Ours (VGGSound, 800 epochs)	81.3	**78.2**	**90.2**

Table A2: Comparison of pretraining methods according to model size. Our self-supervised pretraining scales with the model size, unlike supervised pretraining on ImageNet-21K, as used by MBT [13]. We report audiovisual finetuning accuracy for VGGsound and mAP for AudioSet.

Model size	Pretraining	VGGSound	AudioSet
Base (172×10^6 params)	Scratch	51.0	39.9
	Supervised, ImageNet-21K	64.1	49.6
	Self-supervised, ours	**64.2**	**50.0**
Large (611×10^6 params)	Scratch	41.6	21.5
	Supervised, ImageNet-21K	61.4	48.2
	Self-supervised, ours	**65.0**	**51.8**

Table A2 shows audiovisual recognition performance when training MBT on the target datasets VGGSound and AudioSet for three different pretraining strategies: (1) from scratch (i.e., no pretraining), (2) initializing MBT from a ViT pretrained with supervised image-classification labels on ImageNet-21K (as done in [13]), (3) using our proposed mask-based self-supervised pretraining on each target dataset.

Our proposed pretraining, using *only* the target datasets without labels, outperforms the original MBT setup. Notice that the original MBT setup [13] is based on pretraining on an *external* dataset different from the target ones, and using expensive labels for millions of examples. If we train MBT using data from only VGGSound / Audioset, as our method, we must train it from scratch, and the results are significantly worse.

Table A2 also shows that our proposed pretraining scales better with model size than traditional supervised pretraining, in line with results reported in the original MAE paper [11] on ImageNet.

*Equal contribution. Correspondence to aarnab@google.com.
†Work done during an internship at Google.

Table A3: Additional baseline for Audiovisual MAE on VGGSound. We report the audiovisual finetuning accuracy. Note that joint modelling and pretraining by our proposed Audiovisual MAE model outperforms the baseline of pretraining two separate, unimodal MAE models.

Method	AV accuracy
Separate AudioMAE and VideoMAE	63.3
Audiovisual MAE	64.2

Table A4: Ablation study of different mask ratios. We use a ViT-Base backbone, "Separate" encoding and "Shared" decoding, architecture pretrained for 400 epochs with the "Joint Reconstruction" objective. The table shows audiovisual finetuning accuracy on VGGSound.

Video \ Audio	0.3	0.5	0.7	0.8
0.7	62.4	63.4	62.2	61.6
0.9	63.3	63.0	63.5	62.3
0.95	63.0	63.0	63.0	62.8

A.3. Additional baseline

Table A3 reports an additional baseline for our proposed Audiovisual MAE model.

Here, we train two separate MAE models on audio-only and video-only on VGGSound for 800 epochs, and use this to initialise an MBT model which we then finetune on VGGSound. This corresponds to a "Separate" encoding and decoding strategy, and thus two separate MAEs pretrained in parallel. We compare this to our proposed Audiovisual MAE model.

As shown in Tab. A3, our Audiovisual MAE outperforms this baseline, showing the benefits of joint modelling of both audio and video.

A.4. Masking ratio

Tables A4, A5 and A6 ablate the effect of the masking ratio in the case of audiovisual, audio-only and video-only pretraining respectively.

In all cases, we pretrain for 400 epochs with ViT-Base on VGGSound. We use the "Separate" encoding and "Shared" decoding architecture and the "Joint Reconstruction" objective.

We observe that the optimal masking ratios for unimodal and multimodal pretraining are correlated. However, the best masking ratio for video-only for example is 0.95 (Tab. A6), but this is not the best value for audiovisual pretraining as shown in Tab. A4.

Table A5: Ablation study of mask ratios when pretraining and finetuning on audio-only on VGGSound.

Mask ratio for audio	Accuracy
0.3	55.1
0.5	55.7
0.7	55.5
0.8	55.3

Table A6: Ablation study of mask ratios when pretraining and finetuning on video-only on VGGSound.

Mask ratio for video	Accuracy
0.7	49.1
0.9	49.3
0.95	49.5

Table A7: Effect of the finetuning architecture. For audiovisual finetuning, we can either finetune using the original encoder architecture, or we can initialise an MBT [13] model instead. We consistently find that finetuning with an MBT architecture is better, regardless of the original pretraining architecture.

Pretraining		Finetuning	
Encoder	Decoder	Pretraining encoder	MBT
Early fusion	Shared	59.4	62.2
Early fusion	Separate	58.1	61.1
Separate	Shared	60.4	63.0
Shared	Separate	58.7	61.3

A.5. Ablation of audiovisual finetuning architecture

As mentioned in Sec. 4.1 of the main paper, for audiovisual finetuning, we can either finetune using the original pretraining encoder architecture. Or, we can instead initialise an MBT [13] model. As shown in Tab. A7, we consistently find that finetuning with an MBT model is better, regardless of the original pretraining architecture.

A.6. Modality inpainting

As mentioned in Sec. 4.2 of the main paper, we found that the "Modality inpainting" model is difficult to optimise, and requires learning rate tuning in order to train in a stable manner. This is shown in Fig. A1: The "Joint reconstruction" objective is stable across three different learning rate values. The "Modality inpainting" objective, on the other hand, only trains well for one of these learning rates. At a higher learning rate of 10^{-3}, the loss diverges, which is why we stopped training.

A.7. Mid-fusion layer hyperparameter

For our mid-fusion architecture (Sec. 3.2 of the main paper), we have an additional hyperparameter S, which de-

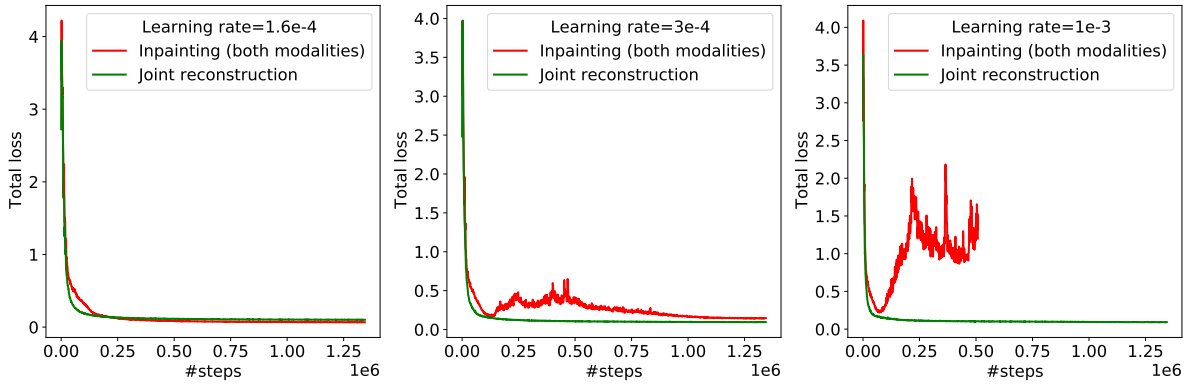

Figure A1: Learning curves for the "Joint Reconstruction" and "Modality Inpainting" objectives. Observe how "Joint Reconstruction" is stable across a wide range of learning rates. "Modality Inpainting", on the other hand, only performs well for a learning rate of 1.6×10^{-4}, and is unstable at higher values. These pretraining experiments were performed on VGGSound for 400 epochs with a batch size of 512.

Table A8: Ablation of S, the hyperparameter denoting the number of shared layers when using the "Mid-fusion" encoding strategy. The experiment is performed on ViT-Base, where there are a total of 12 layers. We report audiovisual finetuning accuracy on VGGSound.

S	Accuracy
$S = 1$	63.4
$S = 2$	63.5
$S = 3$	63.2
$S = 4$	63.1

notes the number of shared layers. Table A8 ablates this hyperparameter for a ViT-Base model with a total of 12 layers. As with the other ablation experiments, it was performed on VGGSound whilst pretraining for 400 epochs.

A.8. Mid-fusion vs Separate encoders on AudioSet

In Sec. 4.2 of the main paper, we show that the "Mid-fusion" encoding strategy slightly outperforms other encoding strategies on audiovisual classification using VGGSound. Here we compare the "Mid-fusion" strategy vs the "Separate" encoders strategy on AudioSet, using our best setup consisting of a ViT-Large backbone pre-trained for 120 epochs. Results in Tab. A9 confirm that "Mid-fusion" also exhibits slightly better performance on AudioSet.

As noted in the main paper, "Early fusion" uses the same model parameters for all modalities, and thus does not allow modality-specific modelling. The late fusion provided by "Separate" encoders, in contrast, does not allocate many parameters to model interactions between modalities. "Mid-fusion" is a middle-ground, featuring both modality-specific parameters, and sufficient layers to model inter-modality relations. The benefits of mid-fusion have also been observed empirically by MBT [13] in a supervised setting.

Table A9: Encoder architecture comparison on AudioSet. Large backbone pretrained for 120 epochs, using a "Shared" decoder.

	A	V	AV
Separate encoders	46.5	30.3	51.4
Mid-fusion	**46.6**	**31.1**	**51.8**

Table A10: Pretraining for the same number of iterations on different subsets of VGGSound produces similar finetuning results.

	A	V	AV
VGGSound-50% for 800 epochs	55.5	48.5	63.4
VGGSound-100% for 400 epochs	55.8	48.5	63.5

A.9. Pretraining for the same number of iterations on different subsets of VGGSound

In Sec. 4.2 of the main paper, we saw that pretraining on VGGSound leads to better performance on Epic Kitchens than pretraining on the substantially larger AudioSet, when using 10x epochs for VGGSound in order to keep the number of training iterations roughly constant. This suggests that the number of iterations of pretraining are more important than the actual size of the pretraining dataset, in line with some of the observations made by [18].

For an additional comparison, in Tab. A10, we conduct a similar experiment now utilising different subsets of VGGSound. In particular, we compare pretraining a ViT-Base backbone on the full VGGSound for 400 epochs, with pretraining on half of VGGSound for 800 epochs, thus keeping the number of training iterations constant. The similar finetuning results of Tab. A10 support the hypothesis posed in Sec. 4.2 that the number of pretraining iterations is more critical than the size of the pretraining dataset. El-Nouby *et al*. [7] and Tong *et al*. [18] have also observed self-supervised pretraining performing well on smaller datasets. We aim to study exactly how much pretraining data is needed further in future work.

Table A11: Comparison of different pretraining architectures. We show audio-only downstream evaluation on VGGSound.

Encoder	Decoder	Linear probing	Full finetuning
Early fusion	Shared	26.2	55.5
Shared	Shared	27.6	55.5
Separate	Shared	27.6	55.4
Mid-fusion	Shared	**27.8**	**55.8**

Table A12: Pretraining time analysis on VGGSound, using identical hardware. We also report audiovisual finetuning accuracy.

Pretraining	Epochs / Iterations	Total time (hours)	AV Accuracy
AudioMAE	800 / 268K	59.0	58.3
VideoMAE	800 / 268K	84.4	62.1
Separate Audio & Video MAEs	800 / 268K	143.4	63.3
Audiovisual MAE (ours)	800 / 268K	89.2	64.2

A.10. Audio-only linear evaluation of different encoder architectures

In Table 1, we saw that different encoder architectures perform similarly for audio-only finetuning. We analysed this effect further in Table A11 by doing linear probing instead. "Early-fusion" performs markedly worse in this case, but the other encoder architectures perform similarly. This suggests that "early-fusion" learns different audio representations, but the effect is concealed by fully finetuning the network. Mid- and late-fusion seem to learn similar representations though.

A.11. Computational cost

Table A12 compares the wallclock training time of our proposed Audiovisual MAE to separately training audio-only and/or video-only MAEs. Audiovisual pretraining is only marginally more expensive than video-only pretraining, and provides substantial accuracy gains. Moreover, we showed in Table 3 that audiovisual pretraining is just as effective for unimodal downstream tasks. We also significantly outperform the baseline of training separate audio-only and video-only MAEs.

B. Experimental Details

In this section, we provide exhaustive details of our experimental setup. We will also release pretraining code and models, and also finetuning code and models upon acceptance. Our models are trained using 32 GPU (Nvidia V100) or Cloud TPU v3 accelerators, using the JAX [3] and Scenic [6] libraries.

B.1. Pretraining hyperparameters

Table A13 details our hyperparameters for pretraining Audiovisual MAE models. Note that we use the same pretraining hyperparameters for different datasets. And we only vary the number of epochs according to the dataset. Our hyperparameters are based on those of [8, 11, 18]. Note that we linearly scale our learning rate with the batch size [10], and we show the learning rate for the reported batch size. Additionally, we can use a larger batch size during pretraining due to the high masking ratio for Audiovisual MAE pretraining. As for data normalization, for RGB frames, we followed ViViT [1] and zero-centered inputs, from the interval $[0, 255]$ to $[-1, 1]$. For audio, we followed MBT [13], and did not normalise the log-mel spectrograms.

Table A13: Pretraining hyperparameters

Configuration	Value
Optimizer	Adam
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
Weight decay	0
Base learning rate	3×10^{-4}
Learning rate schedule	cosine decay
Warm-up epochs	40
Augmentation	None
Batch size	512

Table A14: Hyperparameters of our decoder used during pretraining. We change the size of our decoder based on the size of the encoder, and use hyperparameters following [8, 11, 18]

	Base	Large
Hidden dimension	384	512
Number of layers	4	4
Number of heads	6	8
MLP dimension	1536	2048

Table A14 also lists the configuration of the decoders that we use whilst pretraining. These were set following [8, 11, 18].

B.2. Finetuning hyperparameters

Tables A15, A17 and A16 show our finetuning hyperparameters for the VGGSound, AudioSet and Epic Kitchens datasets respectively. We typically use the same hyperparameters across different datasets. However, we found that audio-only finetuning sometimes required greater regularisation (also noted earlier by [19]), which is why we used a higher Mixup coefficient for it.

For audio, we use two modality-specific regularisers. Firstly, we apply SpecAugment [14] following the settings used in previous works [9, 13]. We also apply random time shifting on the spectrogram, which involves circularly shifting the audio spectrogram by a time offset sampled from a uniform distribution. As mentioned in Sec. 4.2 of the main paper, we are not adopting any dataset balancing techniques

Table A15: VGGSound finetuning hyperparameters

Configuration	A	V	AV
Number of video frames	–	32	32
Spectrogram audio length (seconds)	8	–	8
Optimizer		SGD	
Optimizer momentum		0.9	
Layerwise decay [2, 5]		0.75	
Base learning rate		0.8	
Learning rate schedule		cosine decay	
Gradient clipping		1.0	
Warm-up epochs		2.5	
Epochs		50	
Batch size		64	
SpecAugment [14]	✓	–	✓
Mixup α [20]		0.5	
Stochastic depth [12]		0.3	
Label smoothing [16]		0.3	

Table A16: Epic Kitchens finetuning hyperparameters

Configuration	A	V	AV
Number of video frames	–	32	32
Spectrogram audio length (seconds)	8	–	8
Optimizer		SGD	
Optimizer momentum		0.9	
Layerwise decay [2, 5]		0.75	
Base learning rate		1.2	
Learning rate schedule		cosine decay	
Gradient clipping		1.0	
Warm-up epochs		2.5	
Epochs		50	
Batch size		64	
Random time shifting	✓	–	✓
SpecAugment [14]	✓	–	✓
Mixup α [20]	1.25	0.5	0.5
Stochastic depth [12]		0.3	
Label smoothing [16]		0.3	

for AudioSet. Instead, we finetuned on the AS500K training subset, which is slightly more balanced than the full AS2M (and also smaller, hence faster to process). We also use a larger batch size for AudioSet since it is a larger dataset.

Note that prior work that we compare to, such as MBT [13], used the same regularisers as we do (stochastic depth, mixup, label smoothing). Also following standard practice [1, 4, 13], we process multiple views of the input video, averaging the results of 4 views for every evaluation example.

Table A17: AudioSet finetuning hyperparameters

Configuration	A	V	AV
Number of video frames	–	32	32
Spectrogram audio length (seconds)	10	–	10
Optimizer		SGD	
Optimizer momentum		0.9	
Layerwise decay [2, 5]		0.75	
Base learning rate		1.6	
Learning rate schedule		cosine decay	
Gradient clipping		1.0	
Warm-up epochs		2.5	
Epochs		50	
Batch size		128	
Random time shifting	✓	-	✓
SpecAugment [14]	✓	-	✓
Mixup α [20]	1.25	0.5	0.5
Stochastic depth [12]		0.3	
Label smoothing [16]		0.3	

C. Qualitative Results

Figure A2 shows examples of reconstructions of our model trained with the "Joint reconstruction" objective on the AudioSet dataset.

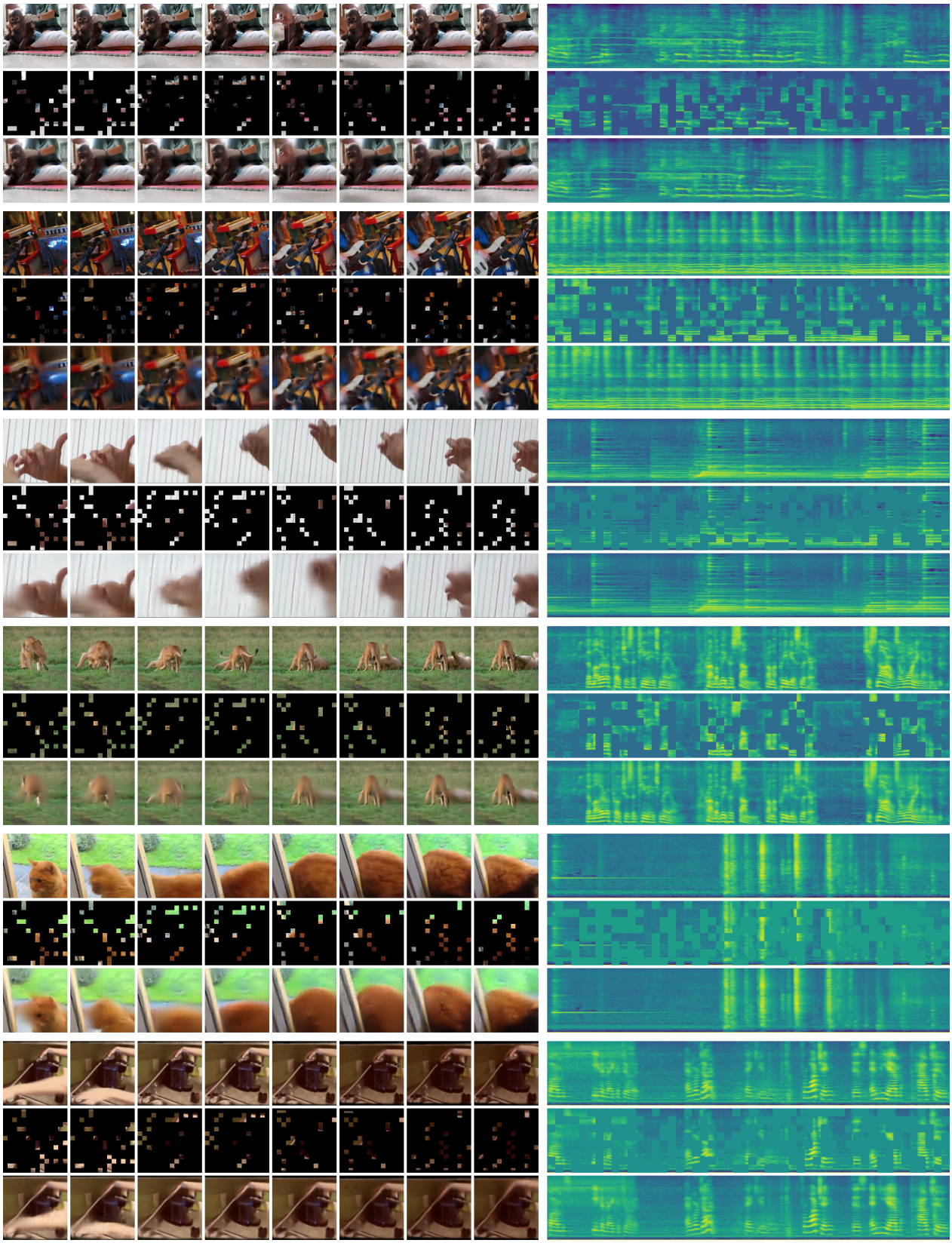

Figure A2: Examples of reconstructions of our model, trained with the "Joint reconstruction" objective on AudioSet. We show video frames on the left, and audio spectrograms on the right. The first row shows the original input, the second the input after masking, and the final row shows the reconstruction produced by the model. For the unmasked patches in the reconstruction, we show the original input. Note that the model is pretrained with 16 video frames, and we show 8 here for clarity. This figure is best viewed on screen, zoomed in.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *ICCV*, 2021. 4, 5

[2] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. 5

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 4

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 5

[5] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. 5

[6] Mostafa Dehghani, Alexey Gritsenko, Anurag Arnab, Matthias Minderer, and Yi Tay. Scenic: A JAX library for computer vision research and beyond. In *CVPR Demo*, 2022. 4

[7] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? In *arXiv preprint arXiv:2112.10740*, 2021. 3

[8] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *arXiv preprint arXiv:2205.09113*, 2022. 4

[9] Yuan Gong, Yu-An Chung, and James Glass. AST: Audio spectrogram transformer. In *Interspeech*, 2021. 4

[10] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. In *arXiv preprint arXiv:1706.02677*, 2017. 4

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 4

[12] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 5

[13] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In *NeurIPS*, 2021. 1, 2, 3, 4, 5

[14] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. SpecAugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech 2019*, pages 2613–2617, 2019. 4, 5

[15] Arda Senocak, Junsik Kim, Tae-Hyun Oh, Dingzeyu Li, and In So Kweon. Event-specific audio-visual fusion layers: A simple and new perspective on video understanding. In *WACV*, 2023. 1

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 5

[17] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018. 1

[18] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *arXiv preprint arXiv:2203.12602*, 2022. 3, 4

[19] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *CVPR*, 2020. 4

[20] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 5