Mirasol3B: A Multimodal Autoregressive Model for Time-Aligned and Contextual Modalities
Supplemental materials

AJ Piergiovanni
Google DeepMind

Isaac Noble
Google Research

Dahun Kim
Google DeepMind

Michael S. Ryoo
Google DeepMind

Victor Gomes
Google Research

Anelia Angelova
Google DeepMind

1. Datasets details

The following datasets have been used for evaluation in the paper:

MSRVTT-QA [13] is a popular Video QA dataset of about 10K video clips and 243K question-answer pairs. It is derived from the MSRVTT dataet by automatic question-answer pairs and contains a certain level of noise. Videos are about 14 seconds in length, on average.

ActivityNet-QA [14] is a commonly used benchmark for understanding of longer videos. It contains 5,800 videos and 58,000 question-answer pairs. It has much longer videos which entail longer and more complex scenes. The video length is about 160 seconds per video on average.

NExT-QA [12] dataset is also addressing long video understanding. It contains 5,440 videos and about 52K manually annotated question-answer pairs. The average length of the videos is 44 seconds. Apart from questions related to descriptions and in the video, NExT-QA focuses on questions related to events and sequence of events within the video, e.g., causal ('Why' and 'How' questions), and temporal - questions related to order of events, or related to concurrent activities and others.

VGG-Sound [3] is a large-scale audio-video dataset, featuring over 200,000 videos accompanied by audio sounds. The data is formulated as classification tasks with 300 audio classes.

Epic-Sound [6] is an audio-video dataset based on the Epic-Kitchens dataset. It has 78.4k examples and 44 target classes.

Kinetics-Sound [2] is a dataset derived from the popular Kinetics-400 video recognition dataset. Kinetics-Sound includes audio inputs sampled together with the video and has 36 classes.

All the abovementioned audio-video datasets used in the paper, have been formulated as datasets for classification tasks. Here we use the class outputs (which are typically short phrases describing an activity, instrument or type of sound e.g 'Knocking on a door') and treat them as open-ended text generation tasks and thus they are now audio-video-text datasets.

2. Additional ablations

Tab. 1 shows additional ablations. This is conducted by a model trained on only 1/2 of the epochs to save compute. All experiments within each ablation table are ran for the same steps.

Autoregressive ablations, equalizing total dimensions. In Tab. 1a we evaluate the autoregressive model vs non-autoregressive one, by equalizing the total number of Combiner dimensions. More specifically, if the full video is ran on T chunks, each of Combiner dimension K, then we compare to a non-autoregressive model of total $T * K$ dimensions, in order to be maximally fair for both models. We see that, when equalizing the total dimensions, an autoregressive model is also more advantageous. More frames are beneficial, as expected, also confirming findings in the paper. We further see that allocating more dimensions, all other things being equal, is slightly beneficial.

Loss ablations: We compare using different loss weights when training (Tab. 1b). We see that increasing the weight for the text generative loss is overall beneficial. This is done only during fine-tuning. This ablation informed out decision to finetune the larger model using a larger unaligned text loss weight of 10.0.

3. Combiner Visualizations.

In Figure Fig. 1, we visualize the different combiners we explored. The Transformer combiner, CLS combiner and Perceiver combiner are all based on transformers taking input of all the video + audio features and reducing them to

Model	Frames	Chunks	Dim	Total Dim	Acc.
Baseline	32	1	256	256	40.4
Baseline	128	1	256	256	44.8
Autoreg.	128	16	16	256	45.5

(a) **Autoregressive model**.

Model	Causal	Video	Text	Acc.
Main	1.0	1.0	1.0	45.0
Text Low	1.0	1.0	0.1	44.6
Text High	1.0	1.0	10.0	45.4

(b) **Loss weights**.

Table 1. **Additional ablation studies.**

m combined features. We found our main combiner to outperform the other two in Table 5 of the main paper. We note that the Perceiver combiner is an adaptation of our combiner by applying Perceiver resampling [7]. The TTM combiner is conceptually different: rather than taking all the previous features as input, it takes only the current timestep features as input and uses a memory mechanism with read and write operations to update it. It then uses a MLP to produce the m combined output features. This reduces memory and compute use and sometimes reduces accuracy.

4. Additional Model and Implementation details

Model Details. The autoregressive text model contains about 1.3B parameters, 400M are for cross-attention weights and 400M for the vocab embeddings and following specifications: layers=18, model dims=1536, hidden dims=12288, heads=12, and head dims=128. About 100M parameters are for the additional weights associated with audio. The remaining parameters are for the video input processor, combiner, causal latent model and video reconstruction model (a bit over 1.5B parameters in total). The combiner, causal latent model and video reconstruction model are transformers with 128M parameters and the following specifications: layers=8, model dims=1024, hidden dims=4096, heads=16, and head dims=64. The video chunk processor has roughly 630M parameters, following ViT-Huge. The convolutional tubes have 1.5M parameters and the transformer has 630M parameters and following specifications: layers=32, model dims=1280, hidden dims=5120, heads=16, and head dims=80. The total parameter size is 3B parameters.

The smaller model used for ablations keeps the same combiner, causal latent model, and video reconstruction model as the main model. However the autoregressive text model is reduced to 128M parameters with the same settings as the combiner, and has 20M cross-attention weights and 260M parameters for the vocab embedding. The audio parameters are held roughly the same. The video input processor is reduced to ViT-Large which has 300M parameters and the following specifications: layers=24, model dims=1024, hidden dims=4096, heads=16, and head dims=80. The total parameter size is 1.15B parameters.

The TTM Combiner, as mentioned is implemented by a TokenLearner [11] function and a transformer. The output dimension $K = 32$ is the same as the output dimension for the standard Transformer Combiner. The output dimensions for the 'Read' and 'Write' functions are 512 and 256, respectfully. These two parameters can be controlled independently to allow more or less capacity to the TTM Combiner. The transformer used within the 'Process' function is of 2 layers, 128 hidden dimension and 12 heads. These are fixed throughout the paper.

Model Pretraining. The pretraining data is the Video-Text Pairs (VTP) dataset which is collected from noisy video-text pairs from the web [1]. The main pretraining is done for the autoregressive, combiner, and the learning components processing the low-level video features (e.g., video tubes convolutions). The text backbone is frozen during pretraining while the other components including the cross attention weights are unfrozen. The model's image and text backbones and cross attention layers are initialized from a contrastively image-text pretrained MaMMUT model [9]. More specifically, MaMMUT is trained jointly with contrastive and text generative objectives, where the latter is not of significant importance, and contrastive-only training is also possible. Pre-training is done on the Align dataset [8]. The audio backbone is also reusing the same pre-trained image backbone. During pretraining, the combiner model, causal latent reconstruction model and video reconstruction model and video tubes are all randomly initialized. All losses are given equal weight during pretraining. For pretraining, we used a learning rate of 1×10^{-5}, batch size of 32, image resolution of 224×224, 128 frames.

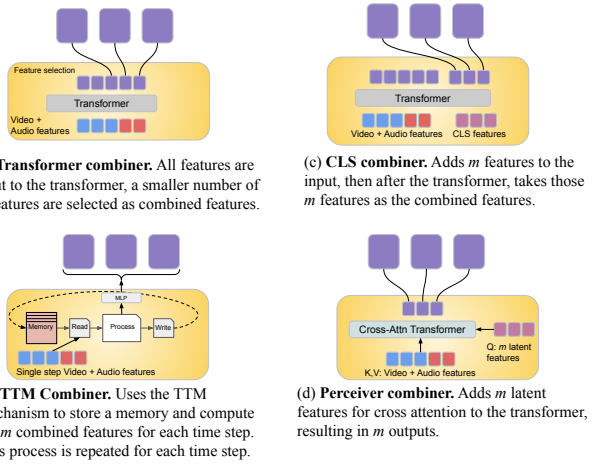
Fine-tuning. During finetuning all parameters are unfrozen. In addition the unaligned text loss is given extra weight and increased 10-fold to better align the training loss

with the final evaluation, since the latent space and video reconstruction are not evaluated. The model is trained for 10 epochs for the MSRVTT-QA dataset and for 80 epochs on ActivityNet-QA and 20 epochs on NExT-QA. For these datasets, we finetune with a learning rate of 5×10^{-6}, weight decay of 0.01, image resolution of 448×448, batch size of 32. We use 128 frames for the main experiments, except for the long video benchmarks where we also report performance with 512. Sampling more frames from the other benchmarks is not porductive as they contain relatively short videos. We used dropout of 0.1, label smoothing of 0.2

Video-Audio Implementation Details. Since the model is pretrained on VTP data, where most videos lack audio, we add a further audio pretraining step here. We use AudioSet-2M [4] and train the model to output the text of the class names. In this step, we freeze the weights of the model, other than the audio weights, allowing the model to learn to handle the spectrogram inputs. During finetuning on the eval datasets, we fully train the model. During finetuning, we also use Mixup [15], specaugment [10], dropout and label smoothing, following the settings of previous works (e.g., [5]). We use a learning rate of 1×10^{-5}, with the Adam optimizer (default settings), weight decay of 0.0001, cosine learning rate decay. We use an image resolution of 448×448, batch size of 32, and 128 frames.

Ablation experiments details. The ablation experiments in Tab. 5a, 5b, 5c, 5d of the main paper are conducted with our small model. The Baseline in Tab. 5a of the main paper uses partitioning, as the rest of the approaches tested in the table, and concatenation of the features to be maximally comparable to others.

The baselines in Tab. 1a use a single time chunk which turn off aligned autoregressive modeling. The different chunks and dimensions explore the relationship between the number of frames, and output size of the combiner (dim). None of the single time chunk settings achieve the same performance as including a an autoregressive representation even at the same total dimensionality.



(a) **Transformer combiner.** All features are input to the transformer, a smaller number of *m* features are selected as combined features.

(c) **CLS combiner.** Adds *m* features to the input, then after the transformer, takes those *m* features as the combined features.

(b) **TTM Combiner.** Uses the TTM mechanism to store a memory and compute the *m* combined features for each time step. This process is repeated for each time step.

(d) **Perceiver combiner.** Adds *m* latent features for cross attention to the transformer, resulting in *m* outputs.

Figure 1. Visualization of the different combiners we explored in this paper. The Transformer combiner, which is the main one we used, simply takes the last m features of the output to represent the combined inputs. We found this to work well. The CLS combiner and Perceiver combiner we found both underperformed the base combiner. The TTM combiner is different, it uses a memory to store the previous representations and has read, process and write operations. We found this method saved memory with some tradeoff for accuracy for some datasets.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. 2

[2] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. 1

[3] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. VGG-Sound: A large-scale audio-visual dataset. In *ICASSP*, 2020. 1

[4] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE ICASSP*, 2017. 3

[5] Mariana-Iuliana Georgescu, Eduardo Fonseca, Radu Tudor Ionescu, Mario Lucic, Cordelia Schmid, and Anurag Arnab. Audiovisual masked autoencoders. *arXiv preprint arXiv:2212.05922*, 2022. 3

[6] Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, and Andrew Zisserman. EPIC-SOUNDS: A Large-Scale Dataset of Actions that Sound. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023. 1

[7] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention, 2021. 2

[8] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 2

[9] Weicheng Kuo, AJ Piergiovanni, Dahun Kim, Xiyang Luo, Ben Caine, Wei Li, Abhijit Ogale, Andrew Dai Luowei Zhou, Zhifeng Chen, Claire Cui, and Anelia Angelova. MaMMUT: A simple architecture for joint learning for multimodal tasks. In *Transactions on Machine Learning Research*, 2023. 2

[10] 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. *arXiv preprint arXiv:1904.08779*, 2019. 3

[11] Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. 2021. 2

[12] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa:next phase of question-answering to explaining temporal actions. *CVPR*, 2021. 1

[13] Jun Xu, Tao Mei, Ting Yao, , and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 1

[14] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-QA: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. 1

[15] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3