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

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

Abstract

One of the main challenges of multimodal learning is combining multiple heterogeneous modalities, e.g., video, audio, and text. Video and audio are obtained at much higher rates than text and are roughly aligned in time. They are often not synchronized with text, which comes as a global context, e.g. a title, or a description. Furthermore, video and audio inputs are of much larger volumes, and grow as the video length increases, which naturally requires more compute dedicated to these modalities, and makes modeling of long-range dependencies harder. We here decouple the multimodal modeling, dividing it into separate autoregressive models, processing the inputs according to the characteristics of the modalities. We propose a multimodal model, consisting of an autoregressive component for the time-synchronized modalities (audio and video), and an autoregressive component for the context modalities which are not necessarily aligned in time but are still sequential. To address the long-sequences of the video-audio inputs, we further partition the video and audio sequences in consecutive snippets and autoregressively process their representations. To that end, we propose a Combiner mechanism, which models the audio-video information jointly, producing compact but expressive representations. This allows us to scale to 512 input video frames without increase in model parameters. Our approach achieves the state-of-the-art on multiple well established multimodal benchmarks. It effectively addresses the high computational demand of media inputs by learning compact representations, controlling the sequence length of the audio-video feature representations, and modeling their dependencies in time.

1. Introduction

Multimodal models aim to combine the signals from multiple varied sources, which makes them both universal and useful for practical applications. However, these modalities have diverse characteristics and are challenging to combine uniformly by a single model. For example, video and text have disparate sampling rates: a video has many frames per second, but text or other similar types of global context, e.g., a description or title, can be provided once per video, or asynchronously to the video. Video also takes a larger portion of the input. At the same time, video and audio are naturally co-occurring and appear (almost) synchronously. They are roughly aligned and complementary. This co-occurrence in time can contribute to their joint learning and serve as a rich self-supervisory learning signal, applied more frequently than global text signals. So, ideally, these modalities need to be processed by differently-synchronized model components, which process more ad-

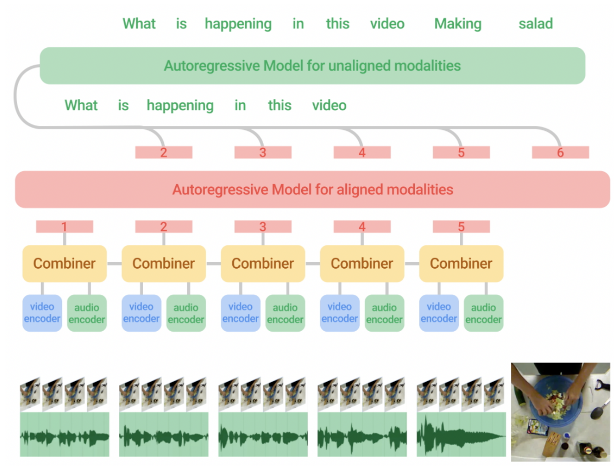

Figure 1. Autoregressive learning of time-aligned video and audio modalities, in time, and decoupling from the autoregressive text modeling allows for more effective multimodal models at smaller sizes and leads to scaling to longer videos.

equately inputs of different frequencies and allocate more parameters to the more abundant modalities.

Following the success of large language models, where text input sequences are processed autoregressively, many recent multimodal models reuse the autoregressive text models, feeding in other modalities, e.g., as embeddings, [2, 7, 24, 26, 33, 47], or by tokenizing the visual inputs to be processed together with the text token sequence [1, 37, 49, 65, 68]). However, the imbalance of the information volume is large and models which are well suited to encode/decode text sequences process only highly compressed image or video features [2, 30]. For example, the Flamingo model [2], subsamples the video features significantly, dedicating only about 1% of the parameters to the image and video inputs, leaving the rest for text processing. Alternatively, methods that process the video, running each frame independently through an encoder or a tokenizer, can process only a limited number of frames [36, 60]. For longer inputs, these representations are insufficient to properly represent the modalities, which inherently limits the ability to model fine-grained or long-range dependencies.

We here propose an audio-video-text multimodal model, where we decouple the autoregressive modeling into a component for time-aligned modalities, e.g., audio and video, which are processed in time, autoregressively, and an autoregressive component for non-time-aligned contextual modalities e.g., text (Fig. 1). Cross-attention weights coordinate the learning between these components. This decoupling allows for better parameter distribution within the model, allocating sufficient capacity for the media modalities (video and audio), and leads to smaller models overall. Furthermore, we partition the time-aligned modalities into time segments, where audio-video representations are jointly learned before modeling their features autoregressively in time. To that end, we introduce a joint feature learning mechanism for audio and video, called the Combiner, which fuses their features and produces a more compact representation. We extract low level spatio-temporal representation from the raw media inputs in order to capture the dynamic nature of videos and combine it with audio features within concurrent timesteps. The Combiner effectively balances the need for efficient audio+video representations and ones which are expressive enough to preserve the media content. It sufficiently represents the events and activities in the videos and other concurrent modalities and can be handled by subsequent autoregressive models, which allows for learning of long-range dependencies. Our model enables consuming multimodal inputs at different rates and scales well with longer videos. Our contributions are:

- An autoregressive multimodal model, subdividing learning into autoregressive modeling for time-aligned media modalities and non-time-aligned contextual modalities.
- Joint feature representation learning via the Combiner to balance the learning of efficient video+audio representations which are also sufficiently expressive.
- We demonstrate learning with 128-512 frames without increase in model parameters. This is in contrast to prior multimodal models that use 8 or 32 frames [13, 60].

Our model outperforms the state-of-the-art on multiple benchmarks, with large margins on audio-video-text datasets and on long video datasets.

2. Related work

Architectures for video-language understanding commonly use a joint transformer, where video inputs are fed in together with text tokens and processed autoregressively [11, 67]). This is often accomplished with tokenizing the visual inputs. Video-text pretraining approaches [23, 27–29, 46, 50] use masked token modeling and reconstruction [11], masking with cross-attention on multimodal inputs [2], or contrastive learning [10, 46, 58, 67]. Visual synthesis models have extensively used autoregressive models, by learning to generate pixel-wise predictions [44], or by learned discrete tokens from images or videos [52, 53, 61]. In other models, encoder-decoder or decoder-only architectures extend an image-text model to a video-text one [20, 36, 47, 60], where video is processed by individual frames which are then combined. Some architectures instead extract full video signals (typically as embeddings) before feeding them to the model [58]. Another option is to attach a projection or re-tokenization layers e.g., as in Perceiver in Flamingo [2], to reduce the amount of visual tokens added to the model. Our approach differs substantially, as the media input features have a specifically designed component to learn them jointly and in time, producing more abstract representations, suitable for modeling long videos.

Multimodal audio-video-text models have also gained popularity [14–16, 41, 68]: UAVM [14] propose joint learning of audio and video by building invariant transformer module which can be reused by either signal. Multimodal Transformer [43] proposes cross-attention mechanisms, for cross-modal learning on all pairs of video-audio-text data, which Pellegrain et al. [31] extends to longer sequences. Zellers et al. [68] demonstrate joint multimodal audio-video-text learning but only aligning text and audio. Gong et al. [15] use contrastive audio-video learning, whereas Huang et al. [16] use masked autoencoder for audio-video learning. Both approaches tokenize the audio video inputs independently in 2D patches which are used for further processing. Contrastive learning for audio-video signals, leveraging the time-alignment between them [19, 38] and audio-video late fusion are also common [32].

Our work is related to long-form video understanding [42, 54]. Long-form videos have been handled by hierarchical feature learning e.g., the Temporal Window Atten-

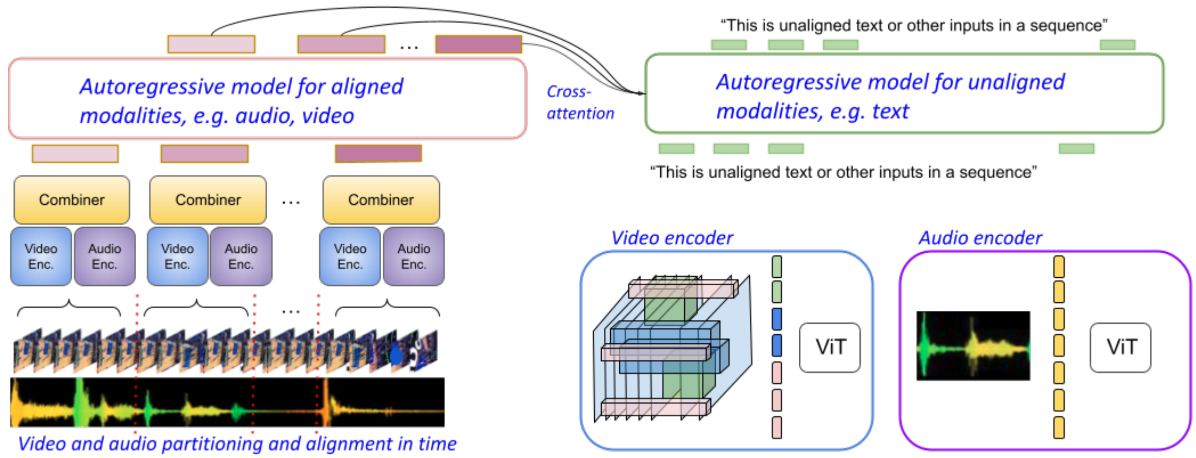

Figure 2. The Mirasol3B model architecture consists of an autoregressive model for the time-aligned modalities, such as audio and video, which are partitioned in chunks (left) and an autoregressive model for the unaligned context modalities, which are still sequential, e.g., text (right). This allows adequate computational capacity to the video/audio time-synchronized inputs, including processing them in time autoregressively, before fusing with the autoregressive decoder for unaligned text (right). Joint feature learning is conducted by the Combiner, balancing the need for compact representations and allowing sufficiently informative features to be processed in time.

tion [42] where dependencies are learned locally and then further propagated to higher level cross-attention modules. Ashutosh et al. [4] propose contrastive learning at different hierarchical levels. Gao et al. [13] segment videos then pool their features into a small representation. Memory-augmented model for long videos [55] accumulate prior context in learnable 'memory', to be referenced at each step of learning. Our work contributes by proposing a balanced approach of locally learning important features, jointly within the modalities.

3. Approach

Autoregressive models are powerful generative models that are well suited for data which appears in a sequence, modeling the probability of the current value, conditioned of previous ones. Video and audio information is sequential but also roughly time-synchronized. At the same time, other modalities e.g., text, might be provided globally per video as context and applied to the full video rather than to specific parts[1]. To address the challenges of modeling diverse multimodal inputs, we propose to subdivide the autoregressive modeling by learning separate autoregressive models: one for the time-aligned modalities (audio-video), Sec. 3.3, and another one for modalities which are not necessarily aligned in time but are still sequential, Sec. 3.4. Learning across these is coordinated by cross-attention mechanisms, where here the audio+video inputs are allocated a lot more parameters and are properly modeled in time. A learning module, called the Combiner (Sec. 3.2), combines the lower-level signals from video/audio snippets. Here information is processed spatio-temporally, extracting features particularly relevant to dynamic changes in the inputs.

Architecture overview. At a high level, the architecture consists of two main learning components (Fig. 2): The first one is an autoregressive component which is designed to process (almost) synchronized multimedia inputs e.g., video+audio and combine their inputs in time (Fig. 3). In order to process the video and audio signals, and to accommodate longer video/audio inputs, they are partitioned into smaller chunks (roughly synchronized in time) for which a joint audio-visual representation is learned via the Combiner as described below (Fig. 4). The second component processes the context, or the signals not aligned in time, such as global text information, which are often still sequential. It is autoregressive as well, and uses the combined latent space as cross-attention inputs.

Model inputs. We have an input video sequence of N frames $\boldsymbol{v} = \{\boldsymbol{v}^f_1, \boldsymbol{v}^f_2, \ldots \boldsymbol{v}^f_N\}$, and audio wave signal of M timesteps $\boldsymbol{a} = \{\boldsymbol{a}^f_1, \boldsymbol{a}^f_2, \ldots \boldsymbol{a}^f_M\}$, where the audio signal is captured during the duration of the video and corresponds to the given video input. Additionally we have an input text sequence $\boldsymbol{t} = \{\boldsymbol{t}^f_1, \boldsymbol{t}^f_2, \ldots \boldsymbol{t}^f_P\}$, which is related to the video/audio and might vary according to the task e.g. it can be a description, a question-answer pair, meta information.

Partitioning of the media inputs. In order to process the video sequence efficiently and to learn the correlation of features in time, we partition the input video into into T non-overlapping segments or chunks, with $\boldsymbol{v}_t$ and $\boldsymbol{a}_t$ denoting the video and audio input per segment (let $K = N/T$). Here each chunk captures all input data between two times-

[1]Text, e.g., ASR, might appear concurrently with audio/video and can contribute to improved understanding of the video content. We leave this to future work.

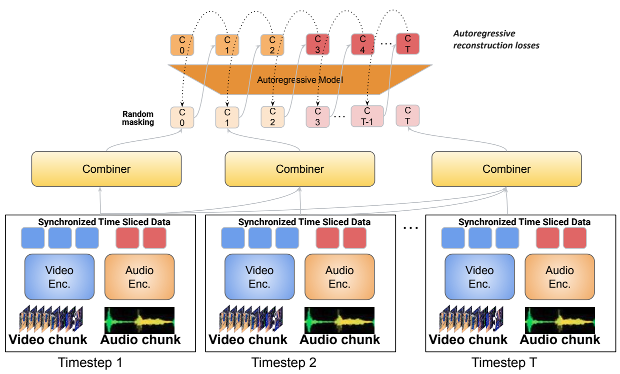

Figure 3. Autoregressive modeling of video and audio in time.

tamps (i.e., video and audio snippets), as follows:

$$\underbrace{\boldsymbol{v}_1^f, \boldsymbol{v}_2^f, \ldots, \boldsymbol{v}_K^f}_{\boldsymbol{v}_1}, \underbrace{\boldsymbol{v}_{K+1}^f, \boldsymbol{v}_{K+2}^f, \ldots, \boldsymbol{v}_{2K}^f}_{\boldsymbol{v}_2}, \ldots \underbrace{\boldsymbol{v}_{(T-1)K+1}^f, \ldots, \boldsymbol{v}_N^f}_{\boldsymbol{v}_T}, \tag{1}$$

Thus the video is represented by its chunks instead, $\boldsymbol{v} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_T\}$, and more specifically latent features will be extracted from each chunk to represent the video (as described in Sec. 3.1). A similar partitioning is done for the audio signals, where they are partitioned in T chunks to be synchronized in time to the video chunks, $\boldsymbol{a} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots \boldsymbol{a}_T\}$. Here too we assume that audio features will be extracted from the raw audio signal, Sec. 3.1.

3.1. Audio/video features

Video features. Prior models captured video information at individual sparsely sampled frames, which lacks the temporal information essential to video understanding and which might miss dynamic events. Alternatively, 3D convolutions [53], sparse 3D tubes [35] and others learn spatio-temporally, which can capture key dynamic changes in the videos. We expand on these ideas and extract sparse 3D tubes [35] from the videos which span all 3 dimensions of the video snippet starting at various locations. The sparse 3D tubes, together with standard 2D patches are processed via a ViT encoder. Rather than applying offsets, the tubes are applied at the beginning of the snippet. Per video chunk t we denote $\hat{\boldsymbol{v}}_t$ as the time-aligned features for this chunk, and thus $\hat{\boldsymbol{v}} = \{\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \ldots, \hat{\boldsymbol{v}}_T)$ are the time-aligned video representations for the whole video.

Audio features. Audio inputs arrive at a predefined frequency and can be processed in various ways. We here represent the audio as a spectrogram. The spectrogram is created so that the time bands match the 25 frames per second used in the videos, and thus can easily be split into snippets aligned with the video. The spectrogram for each snippet is processed by a ViT model, after an audio input projection layer. The ViT backbone is the same as the one used for video features. Reusing the visual component is previously shown to be advantageous [14]. Similarly to above, we denote $\hat{\boldsymbol{a}}_t$ to be the audio feature per chunk t and $\hat{\boldsymbol{a}} = \{\hat{\boldsymbol{a}}_1, \hat{\boldsymbol{a}}_2, \ldots, \hat{\boldsymbol{a}}_T)$ for the full video.

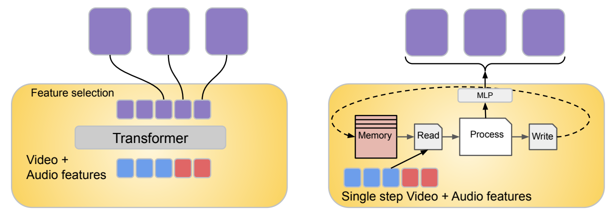

Figure 4. Combiners: Transformer Combiner (left): all features are input to the transformer, a smaller number of m features are selected as combined features. TTM Combiner (right): uses the TTM mechanism to store a memory and compute the m combined features for each time step. This process is repeated for each step.

3.2. Modality Combiner

The task of the Combiner module is two-fold: 1) to combine the video (and audio) features at a specific snippet of time, learning their joint representation and 2) effectively compress the representation from each video/audio snippet, which allows our model to scale to longer videos.

When partitioning the inputs, the features for each modality, video and audio in this case, are (roughly) time-aligned latent features $\hat{\boldsymbol{v}} = \{\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \ldots, \hat{\boldsymbol{v}}_T)$ and $\hat{\boldsymbol{a}} = \{\hat{\boldsymbol{a}}_1, \hat{\boldsymbol{a}}_2, \ldots, \hat{\boldsymbol{a}}_T)$, where the maximum timestamp for any data incorporated into $\hat{\boldsymbol{v}}_t$ or $\hat{\boldsymbol{a}}_t$ is less than the minimum timestamp of any data incorporated into $\hat{\boldsymbol{v}}_{t+1}$ or $\hat{\boldsymbol{a}}_{t+1}$. Explicitly $\hat{\boldsymbol{v}}_t$ is composed of f features of size d giving it a shape of (f, d) and $\hat{\boldsymbol{a}}_t$ is composed of s features also of size d with shape (s, d). The role of the Combiner is to map such time-aligned modal latent features into a smaller set of shared latent features. Specifically let $\boldsymbol{u} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_T\}$ where $\boldsymbol{u}_t = (\hat{\boldsymbol{v}}_t, \hat{\boldsymbol{a}}_t)$ having size (n, d) and $n = f + s$ be the set of all time-aligned features from all modalities. The Combiner then maps $\boldsymbol{u}$ to a lower dimensional latent feature space $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ where $\boldsymbol{x}_t$ has shape (m, d) where $n >> m$.

Since the Combiner is reducing the dimensionality of video+audio features, it can effectively use all features in the sequence, not only the ones per chunk. However, since the features produced by the Combiner are going to be used in the sequential autoregressive modeling of video/audio, it is important for the Combiner to not break causality thus:

$$\boldsymbol{x}_t = \text{Combiner}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_t) \tag{2}$$

We utilize two different architectures for the Combiner, a standard Transformer one and a memory based one, based on the Token Turing Machines [40], to reduce memory.

Causal Transformer Combiner. We explore a straightforward version of the Combiner, which consists of a standard Transformer model, here of R layers (here, $R = 8$). For each step t it maps the original set of features $\boldsymbol{u}_t$ to $\boldsymbol{x}_t$ where $\boldsymbol{x}_t$ is of much lower dimensionality, i.e., effectively reducing the number of tokens (here $m = 32$) (Fig. 4). The inputs to the Combiner are the latent features of the video and audio, which are concatenated before being fed to the Combiner. We here specifically implement a causal version of the transformer as it masks out inputs from future timestamps (i.e., $> t$). The attention mechanism of the transformer is modified to mask features at the time-chunk level as described later in Sec. 3.3.1 (using Eq. (8)), thus all features from $\boldsymbol{u}_t$ and the previous time steps are used to compute each output feature in $\boldsymbol{x}_t$ as in Eq. (2). This effectively applies attention mechanisms to all the modality inputs jointly while respecting causality.

Token Turing Machine Combiner. Token Turing Machine (TTM) [40] is a recurrent sequential model with Transformers and token-based operations. It maintains an external 'memory' M_t as a set of features, and updates it at every time step by reading and writing. Given inputs $\boldsymbol{u}_t$ at each time step, it first 'reads' features to be processed, from input features as well as memory features. Such features, $\boldsymbol{z}_t$, are passed to the 'processor', which is implemented as a standard Transformer, generating a set of intermediate output features $\boldsymbol{o}_t$. These intermediate outputs are then used to update M_t (i.e., memory 'write') as well as to produce the final output $\boldsymbol{x}_t$.

$$\boldsymbol{z}_t = \text{Read}(\boldsymbol{u}_t, M_t) \tag{3}$$
$$\boldsymbol{o}_t = \text{Process}(\boldsymbol{z}_t) \tag{4}$$
$$M_{t+1} = \text{Write}(M_t, \boldsymbol{o}_t, \boldsymbol{u}_t) \tag{5}$$
$$\boldsymbol{x}_t = \text{Output}(\boldsymbol{o}_t) \tag{6}$$

The key idea is to make the TTM processor generate the outputs by utilizing memory M_t instead of the entire history of features $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{t-1}\}$. Once trained, the differentiable read and write operations are optimized to maintain M_t so that it stores important features from the previous time steps $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{t-1}\}$ and updates it, at every step.

We implement TTM as the Combiner module to sequentially combine $\boldsymbol{u}$. The function 'Process' is implemented with a standard Transformer with layers of multi-head self-attention and MLPs. The functions 'Read', 'Write', and 'Output' are implemented with TokenLearner [39] (which is similar to Perceiver [18] and attention pooling [21]). Note that we are able to separately control the number of features in the memory as well as the number of 'Output' function features, allowing efficient Combiner computation and feature generation.

The key advantage of the TTM Combiner is its utilization of memory features to sequentially process $\boldsymbol{u}_t$. The number of such memory features are much smaller than the total number of history features ($\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{t-1}\}$) in general (e.g., 256 vs. $\sim$10k). This not only makes TTM a natural fit for the model, but also reduces the total time complexity of the TTM Combiner to be constant with respect to t, instead of $O(t)$ or $O(t^2)$ in Transformers. We observe that the TTM Combiner saves memory in both training and inference, using about 30% less memory and reduces the runtime by about 18%.

3.3. Time-Aligned Video/Audio Autoregressive Modeling

We describe the autoregressive modeling of time-aligned video and audio. We apply autoregressive modeling strategy where we condition video/audio representations corresponding to a time interval on feature representations from previous time intervals. These representations are learned jointly by the Combiner, as described Sec. 3.2. As mentioned, the video is first partitioned in T smaller video snippets. Each of the snippets itself can be of size 4-64 frames (overlap is possible but currently not used). We extract spatio-temporal information into latent video features $\hat{\boldsymbol{v}}_t$ and audio features $\hat{\boldsymbol{a}}_t$ from the same video partition, apply Combiner to produce x_t. The feature representations per video chunk x_t are then fed sequentially to the autoregressive model, where at each step we reconstruct the features from the previous step, conditioned on the prior inputs and the latent vector $\boldsymbol{h}$ which corresponds to the latent representations learned within the autoregressive model:

$$p(\boldsymbol{v}, \boldsymbol{a}) = \prod_{t=1}^{T} p(\boldsymbol{v}_{t+1}, \boldsymbol{a}_{t+1}|\boldsymbol{h}_t) p(\boldsymbol{h}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{v}_t, \boldsymbol{a}_t) \tag{7}$$

where $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_T\}$, and $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots \boldsymbol{a}_T\}$ are the feature representations from the video and audio, $p(\boldsymbol{x}_t|\boldsymbol{v}_t, \boldsymbol{a}_t)$ is estimated by the Combiner, and $p(\boldsymbol{h}_t|\boldsymbol{x}_t)$ is estimated by the latent causal model, $p(\boldsymbol{v}_{t+1}, \boldsymbol{a}_{t+1}|\boldsymbol{h}_t)$ by the modality reconstruction model (described below). This allows for learning from previous representations in the sequence (in time) and aims to predict the next-step feature representation (Fig. 3). While autoregressive modeling has been used for videos and images, it is often done on a pixel-by-pixel basis [51] which is highly inefficient and captures only short-term dependencies. With our approach, with autoregressive modeling and the Combiner, we address both shortcomings. We note that the Combiner also accumulates information from prior chunks, however, the autoregressive model works at a higher level of abstraction with already learned features from the Combiner. In the ablations, we find that it is most beneficial when both mechanisms work together.

Latent Causal Modeling. The autoregressive latent model estimates: $\prod_{t=1}^{T} p(\boldsymbol{h}_t|\boldsymbol{x}_t)$. This is done by applying an autoregressive transformer to $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$

to produce $\hat{\boldsymbol{h}} = \{\hat{\boldsymbol{h}}_1, \hat{\boldsymbol{h}}_2, \dots, \hat{\boldsymbol{h}}_T\}$ where the target of $\hat{\boldsymbol{h}}_t$ is $\boldsymbol{x}_{t+1}$ so the difference between $\boldsymbol{x}_{2,\dots,T}$ and $\hat{\boldsymbol{h}}_{1,\dots,T-1}$ is used as a loss to control the latent representation of the Combiner output $\hat{\boldsymbol{x}}$. Since we are modeling data autoregressively in time, care must be taken with the attention mechanism during training, the transformer uses a modified attention mechanism as described in Sec. 3.3.1, Eq. (8).

Modality Reconstruction. The autoregressive modality reconstruction models estimate $\prod_{t=1}^{T} p(\boldsymbol{v}_{t+1}, \boldsymbol{a}_{t+1} | \hat{\boldsymbol{h}}_t)$. This is done by applying a separate transformer to $\hat{\boldsymbol{h}}$ to produce reconstructions of the audio and video signals $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{a}}$, which is added as an optional loss below. To save on computation, the video input $\boldsymbol{v}$ is down sampled to $\boldsymbol{v}^{small}$ for the reconstruction target, thus the actual reconstruction is $\hat{\boldsymbol{v}}^{small}$.

3.3.1 Attention mechanisms for Autoregressive modeling

Since the autoreggressive models are trained in time, masking is done to satisfy causality. We note that the attention mechanisms within and across chunks need to be modified when masking. This applies to both the Combiner and the Autoregressive learning (Sections Sec. 3.2 and Sec. 3.3). When masking features for autoregressive modeling, the standard pattern of masking each feature individually would mask features from within the same time-chunk from each other. While this would still satisfy causality, it unnecessarily restricts the model, preventing features from within the same time-chunk from interacting based on position within the time-chunk. To allow features in the same chunk to interact, the autoregressive mask between all features $\boldsymbol{i}$, which fall in a time-chunk t, and another feature $\boldsymbol{j}$ is computed as follows (N is the number of features and T the number of time-chunks):

$$mask_j^i = \begin{cases} 0 & j <= \text{ceil}(t * T/N) * N/T \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

3.4. Combining Aligned and Non-aligned Autoregressive Modeling

Text, or other context information, might not necessarily be aligned in time with the video and audio modalities. It is still sequential. So here it is modeled by a separate autoregressive model, devoted to text representations and to combining the visual-audio information together. Assuming tokenization for the input text $\boldsymbol{t} = \{\boldsymbol{t}_1^f, \boldsymbol{t}_2^f, \dots \boldsymbol{t}_P^f\}$ is provided, obtaining a tokenized text sequence $\boldsymbol{t} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \dots \boldsymbol{w}_L\}$ of length L, we model the text sequentially as conditioned on audio and video. In order to combine the outputs of the video/audio autoregressive model we use cross-attention strategy [2]. Here, unlike prior work, all feature representations $\hat{\boldsymbol{h}} = \{\hat{\boldsymbol{h}}_1, \hat{\boldsymbol{h}}_2, \dots, \hat{\boldsymbol{h}}_T\}$ from the latent causal model are used in the main text model.

$$p(\boldsymbol{w}|\hat{\boldsymbol{h}}) = \prod_{l=1}^{L} p(\boldsymbol{w}_l | \boldsymbol{w}_{l-1}, \hat{\boldsymbol{h}}). \tag{9}$$

The autoregressive text model estimates Eq. (9) by applying a transformer to the input text sequence $\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_L\}$ and using the latent model output $\hat{\boldsymbol{h}}$ as cross-attention to produce $\hat{\boldsymbol{w}}$. The loss is the standard cross-entropy loss between target $\boldsymbol{w}$ and output text sequences $\hat{\boldsymbol{w}}$. This provides further feedback to the Combiner latent representation $\hat{\boldsymbol{h}}$ through the cross-attention layer. Of note is that since all parts of the model are autoregressive, it is naturally applicable to streaming videos.

3.5. Model Losses

We use two main losses, each driving the corresponding autoregressive model:

Latent space reconstruction loss for time-aligned inputs is the difference between $\boldsymbol{x}_{2,\dots,T}$ and $\hat{\boldsymbol{h}}_{1,\dots,T-1}$ in the autoregressive setting such that $\hat{\boldsymbol{h}}_t = \boldsymbol{x}_{t+1}$. We apply a L^2 normalization and then take dot product between the feature vectors as the loss (i.e., cosine similarity).

Unaligned text cross entropy loss is the standard cross-entropy loss between $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$ for the unaligned text output.

Additionally we implement the loss to encourage modality reconstruction as described in Sec. 3.3. More specifically we add a **video reconstruction loss** which is commonly used for video (audio reconstruction loss can also be added). Similar to latent space reconstruction above, the video reconstruction loss approximates the difference between $\hat{\boldsymbol{v}}^{small}$ and $\boldsymbol{v}^{small}$ also in an autoregressive setting such that $\hat{\boldsymbol{v}}_t^{small} = \boldsymbol{v}_{t+1}^{small}$. We use the same distance metric on the video reconstruction as we use on the latent space reconstruction problem. While this loss can be useful, especially for generation tasks, we find that for our model, it is mostly subsumed by the latent space reconstruction loss.

These losses are weighted to compute the final loss.

3.6. Implementation details

Model: Our model has 3B parameters; without audio it is 2.9B. A little over half of the parameters are for the audio+video autoregressive model. Our models work on 128 frames customarily (16 chunks, 8 frames), but can handle more for longer videos (e.g., 512=16 chunks x 32 frames). We use Combiner dimension $m = 32$. We apply random masking to the Combiner output features at a ratio of 0.75% as a form of dropout regularization as we found this stabilizes the causal model latent reconstruction. Due to the design of our model (partitioning and Combiner), adding more frames, or increasing the chunk size, number of chunks, etc. lead to only marginal increase in parameters. Increasing

Method	Accuracy (%)
Just Ask [62]	41.5
ALPRO [23]	42.1
MERLOT [67]	43.1
VIOLETv2 [12]	44.5
VindLU [9]	44.6
VideoOFA [8]	45.4
GIT2 [47]	45.6
Iterative Co-Tok [34]	45.7
VideoCoca [60]	46.3
All-in-one [45]	46.8
UMT-L [27]	47.1
InternVideo [50]	47.1
Flamingo [2]	47.4
M-PLUG2 [58]	48.0
Mirasol3B - TTM	**50.01**
Mirasol3B	**50.42**

Table 1. **Video QA results on MSRVTT-QA.** Results in gray show VideoQA as classification.

Method	Acc %
Just Ask [62]	38.9
MERLOT [67]	41.4
FrozenBiLM [63]	43.2
VideoCoca [60]	56.1
Sing-Temp [22]	44.1
VindLU [9]	44.7
UMT-L [27]	47.9
Mirasol3B - 512 frames TTM	**49.85**
Mirasol3B - 128 frames	**48.25**
Mirasol3B - 512 frames	**51.13**

Table 2. **Long Video QA results on ActivityNet**. Gray is for classification setting.

Method	(Acc %)
CLIP (single frame)	43.7
VQA-T [62]	52.32
AIO [45]	50.60
ATP [5]	54.3
VGT [57]	55.02
MIST - CLIP [13]	57.18
HiTeA [64]	63.1
Mirasol3B - 512 frames TTM	**73.2**
Mirasol3B - 128 frames	**68.2**
Mirasol3B - 512 frames	**72.0**

Table 3. **Long Video QA results on NExT-QA**.

the number of chunks, while not leading to parameter increases, increases memory, which underscores the importance of the Combiner and particularly the TTM. **Model training:** The model is pretrained on the Video-Text Pairs (VTP) dataset which is collected from noisy video-text pairs from the web [2]. We use only about 12% of the data, 3M samples. All losses are given equal weight during pretraining. During finetuning the unaligned text loss is increased 10-fold to better align the training loss with the final evaluation, which we also confirm experimentally.

4. Experiments

We report results on standard Video Question Answering (VideoQA) benchmarks, on long-video VideoQA benchmarks and on Audio+Video benchmarks. We report results using the **open-ended text-generative evaluation**, following [22, 25]. Our model generates a free-form text response which is compared to the target response for an exact match. This is more challenging than a classification setting, as our model might generate a correct answer (e.g. a synonym to the desired answer) but which is not among the target classes. This evaluation is more general and widely applicable.

Video Question Answering. We first report Video Question Answering results on the MSRVTT-QA VideoQA dataset [59], as the most popular Video QA benchmark. The results are shown in Tab. 1 alongside the best state-of-the-art (SOTA) performances. Our method outperforms all prior methods on this challenging dataset, including the ones with classification evaluation which are at an advantage during evaluation. At less than 3B parameters, our model also outperforms the 5B GIT2 [47] by a large margin, and outperforms the very big Flamingo [2] of 80B parameters (the full fine-tuning Flamingo result [2] is reported for direct comparison). This shows the benefit of our model design, where these results can be achieved with much fewer parameters and respectively much less compute needs.

Long Video Question Answering. We further report Video QA results on long video datasets. ActivityNet-QA [66] contains longer videos of about 160 seconds per video. NExT-QA [56] is targeting complex events with long videos of about 44 seconds. We sample up to 512 frames (e.g. 16 chunks of 32 frames each) without increasing the model size. Results are in Tab. 2, Tab. 3, showing we outperform the SOTA with both 128 and 512 frames, where clear improvements are gained from using more frames, and without any increase in model size. We also outperform with either using Transformer Combiner or TTM Combiner.

Audio-Video Results. Tab. 4 shows results on three Audio-Video benchmarks: Kinetics-Sound [3], VGG-Sound [6] and Epic-Sound [17]. Since these datasets are Audio-Video classification, we treat the task as **open-ended generation**: we input the text 'Classify the video audio clip.' and expect the output to be the target class name

Method	Acc. %
MBT [30] (A+V)	85.0
Mirasol3B (Sm, Video)	81.3
Mirasol3B (Sm, A+V)	85.0
Mirasol3B TTM (A+V)	**88.3**
Mirasol3B (A+V)	**90.1**

(a) **Kinetics-Sound**.

Method	Acc. %
UAVM [14]	65.8
MMT [69]	66.2
MAViL [16]	67.1
ONE-PEACE [48]	68.2
Mirasol3B TTM (A+V)	66.4
Mirasol3B (A+V)	**69.8**

(b) **VGG-Sound**.

Method	Acc. %
SSAST[17]	53.47
ASF[17]	53.75
Mirasol3B (Audio)	62.4
Mirasol3B (Video)	72.4
Mirasol3B TTM (A+V)	**79.4**
Mirasol3B (A+V)	**78.2**

(c) **Epic-Sound.**.

Table 4. **Audio-Video results on Kinetics-Sound, VGG-Sound, and Epic-Sound.**

Model	Frames/Chunks	Acc.
Baseline	32/4	41.5
+ AR	32/4	43.2
+ Combiner	32/4	42.1
+ AR + Combiner	32/4	44.7
+ Pretraining	32/4	45.2
+ AR + Comb. + PT	32/4	47.9

(a) **Effects of proposed components**.

Combiner type	Fr./Ch.	Acc.
Perceiver	32/4	43.1
Transf.+CLS	32/4	43.7
Ours-Transf.	32/4	44.2
Ours-TTM	32/4	44.8

(b) **Combiner types**.

Model	Frames/Chunks	Acc.
Baseline	64/1	41.8
Ours-Autoreg.	64/8	45.1
Ours + BD	64/8	45.1
Ours-Autoreg.	128/8	45.8

(c) **Autoregressive model more frames**.

Model	Fr./Ch.	Dim	Acc.
Ours-8	32/4	8	42.53
Ours-16	32/4	16	43.36
Ours-32	32/4	32	44.20
Ours-64	32/4	64	44.22

(d) **Combiner dimension**.

Table 5. **Ablation studies on the MSRVTT-QA dataset.**

e.g., 'playing drums', where only an exact match is counted as accurate answer. Across all datasets, we outperform the SOTA with large margins, despite the more challenging open-text generation evaluation, as opposed to classification used in all prior works.

4.1. Ablations

The ablations (Tab. 5), are conducted with the video and text model in order to understand the main behaviors of this architecture. We use a smaller model and configuration and 2x fewer pretraining steps with the same batch size to save compute (see the supp. material for more details).

Main model components: We first study the effect of each component (Tab. 5a). We find that on top of a baseline model, adding each part, the autoregressive (AR) model, the Combiner, and pretraining, each individually help and the combination of all three further help.

Combiner type ablations: We compare Transformer-based (ours, CLS and Perceiver [2]) and TTM Combiners. The CLS-token Combiner appends m learnable features to the end of the sequence and takes their values as the combined features after passing the whole sequence through the transformer. Our main Combiners are shown in Fig. 4. We use the same settings for direct comparison. Tab. 5b shows that our proposed Combiners perform best.

Autoregressive modeling in time: We ablate the Autoregressive part of the model. Tab. 5c shows that processing the video in chunks autoregressively in time is more advantageous than learning from the full video at once, with a large jump in performance (first two rows). Not only is our autoregressive model feasible for longer videos but it is also more beneficial for the same size inputs. More frames per chunk contribute to the improvements (rows two and four). We also compare to a bidirectional (BD) model, finding that the performance is the same as the autoregressive portion.

Combiner size ablations. We compare the number of features output by the Combiner per timestep. We noticed a trend for larger Combiner outputs giving better results, lines 3-4 (Tab. 5d). We chose 32 as a trade-off between sufficiently compact feature length and sufficiently expressive.

5. Conclusions

We propose a multimodal autoregressive model which decouples the autoregressive modeling into a component, devoted to time-aligned modalities (video, audio) and another one for the non-aligned, contextual modalities (text). To address long video/audio inputs we partition the media inputs and learn from them jointly by a Combiner, which allows to control the sequence lengths. The model can handle 512 frames, without increasing its size. Our approach not only enables working with long videos effectively but also outperforms SOTA, achieving gains over previous models.

References

[1] Armen Aghajanyan, Bernie Huang, Candace Ross, Vlad Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. CM3: A causal masked multimodal model of the internet. In *ArXiv:2201.07520*, 2022. 2

[2] 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, 6, 7, 8

[3] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. 7

[4] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In *CVPR*, 2023. 3

[5] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2917–2927, 2022. 7

[6] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. VGG-Sound: A large-scale audio-visual dataset. In *ICASSP*, 2020. 7

[7] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, , Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model. In *ICLR*, 2023. 2

[8] Xilun Chen, Lili Yu, Wenhan Xiong, Barlas Oguz, Yashar Mehdad, and Wen-Tau Yih. Videoofa: Two-stage pre-training for video-to-text generation. In *arXiv:abs/2305.03204*, 2023. 7

[9] Feng Cheng, Xizi Wang, Jie Lei1 David Crandall, Mohit Bansal, and Gedas Bertasius. Vindlu : A recipe for effective video-and-language pretraining. *arXiv preprint arXiv:2212.05051*, 2022. 7

[10] Sixun Dong, Huazhang Hu, Dongze Lian, Weixin Luo, Yicheng Qian, and Shenghua Gao. Weakly supervised video representation learning with unaligned text for sequential videos. In *CVPR*, 2023. 2

[11] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. In *arXiv:2111.1268*, 2021. 2

[12] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. An empirical study of end-to-end video-language transformers with masked visual modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7

[13] Difei Gao, Luowei Zhou, Lei Ji, Linchao Zhu, Yi Yang, and Mike Zheng Shou. Mist: Multi-modal iterative spatial-temporal transformer for long-form video question answering. In *CVPR*, pages 14773–14783, 2023. 2, 3, 7

[14] Yuan Gong, Alexander H. Liu, Andrew Rouditchenko, and James Glass. Uavm: Towards unifying audio and visual models. In *IEEE SIGNAL PROCESSING LETTERS*, 2022. 2, 4, 8

[15] Yuan Gong, Andrew Rouditchenko, Alexander H. Liu, David Harwath, Leonid Karlinsky, Hilde Kuehne, and James Glass. Contrastive audio-visual masked autoencoder. In *ICLR*, 2023. 2

[16] Po-Yao Huang, Vasu Sharma, Hu Xu, Chaitanya Ryali, Haoqi Fan, Yanghao Li, Shang-Wen Li, Gargi Ghosh, Jitendra Malik, and Christoph Feichtenhofer. Mavil: Masked audio-video learners. In *ArXiv:2212.08071*. 2, 8

[17] 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. 7, 8

[18] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention, 2021. 5

[19] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. 2018. 2

[20] 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

[21] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. 2019. 5

[22] Jie Lei, Tamara L. Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learnin. *ArXiv:abs/2206.03428*, 2022. 7

[23] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *CVPR*, 2022. 2, 7

[24] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *NeurIPS*, 2021. 2

[25] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh D. Gotmare, Shafiq Joty, Caiming Xiong, and Steven C.H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 7

[26] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. 2

[27] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *ICCV*, 2023. 2, 7

[28] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[29] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 2

[30] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. 2021. 2, 8

[31] Victor Pellegrain, Myriam Tami, Michel Batteux, and Céline Hudelot. Streamult: Streaming multimodal transformer for heterogeneous and arbitrary long sequential data. In *ArXiv:2110.08021*, 2021. 2

[32] Lionel Pibre, Francisco Madrigal, Cyrille Equoy, Frédéric Lerasle, Thomas Pellegrini, Julien Pinquier, and Isabelle Ferrané. Audio-video fusion strategies for active speaker detection in meetings. In *ArXiv:2206.10411*, 2022. 2

[33] AJ Piergiovanni, Wei Li, Weicheng Kuo, Mohammad Saffar, Fred Bertsch, and Anelia Angelova. Answer-me: Multi-task open-vocabulary visual question answering. *arXiv preprint arXiv:2205.00949*, 2022. 2

[34] AJ Piergiovanni, Kairo Morton, Weicheng Kuo, Michael Ryoo, and Anelia Angelova. Video question answering with iterative video-text co-tokenization. *ECCV*, 2022. 7

[35] AJ Piergiovanni, Weicheng Kuo, and Anelia Angelova. Rethinking video vits: Sparse video tubes for joint image and video learning. *CVPR*, 2023. 4

[36] AJ Piergiovanni, Weicheng Kuo, Wei Li, and Anelia Angelova. Dynamic pretraining of vision-language models. In *First workshop on Multimodal Representation Learning, International Conference on Learning Representations (ICLR)*, 2023. 2

[37] Scott Reed, Konrad Żołna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. In *ArXiv:2205.06175*, 2022. 2

[38] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, and James Glass. Avlnet: Learning audio-visual language representations from instructional videos. In *INTERSPEECH*, 2021. 2

[39] Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. 2021. 5

[40] Michael S. Ryoo, Keerthana Gopalakrishnan, Kumara Kahatapitiya, Ted Xiao, Kanishka Rao, Austin Stone, Yao Lu, Julian Ibarz, and Anurag Arnab. Token turing machines. In *CVPR*, 2023. 4, 5

[41] Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Fine-grained audio-visual joint representations for multimodal large language models. In *ArXiv:2310.05863*, 2023. 2

[42] Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu. Long-form video-language pre-training with multimodal temporal contrastive learning. 2022. 2, 3

[43] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *ArXiv:2110.08021*, 2021. 2

[44] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, and Koray Kavukcuoglu Alex Graves. Conditional image generation with pixelcnn decoders. In *arXiv:1606.05328*, 2016. 2

[45] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Ge Yuying, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022. 7

[46] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. Omnivl:one foundation model for image-language and video-language tasks. 2022. 2

[47] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 2, 7

[48] Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, and Chang Zhou. One-peace: Exploring one general representation model toward unlimited modalities. In *ArXiv:2305.11172*, 2023. 8

[49] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 2

[50] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He1, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning. In *arXiv preprint arXiv:2212.03191*, 2022. 2, 7

[51] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*, 2020. 5

[52] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. In *ArXiV:2104.14806*, 2021. 2

[53] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nuwa: Visual synthesis pre-training for neural visual world creation. In *ArXiv:2111.12417*, 2022. 2, 4

[54] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *CVPR*, 2021. 2

[55] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, and Jitendra Malik amd Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *ArXiv:2201.08383*, 2023. 3

[56] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa:next phase of question-answering to explaining temporal actions. *CVPR*, 2021. 7

[57] Junbin Xiao, Pan Zhou, Tat-Seng Chua, and Shuicheng Yan. Video graph transformer for video question answering. In *European Conference on Computer Vision*, pages 39–58. Springer, 2022. 7

[58] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, Guohai Xu, Ji Zhang, Songfang Huang, Fei Huang, and Jingren Zhou. mplug-2: A modularized multi-modal foundation model across text, image and video. In *arXiv preprint arXiv:2302.00402*, 2023. 2, 7

[59] Jun Xu, Tao Mei, Ting Yao, , and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 7

[60] Shen Yan, Tao Zhu, ZiRui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Videococa: Video-text modeling with zero-shot transfer from contrastive captioners. In *ArXiV:2212.04979*, 2022. 2, 7

[61] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *ArXiv:abs/2104.10157*, 2021. 2

[62] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1686–1697, 2021. 7

[63] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *ArXiv:abs/2206.08155*, 2022. 7

[64] Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. Hitea: Hierarchical temporal-aware video-language pre-training. *arXiv preprint arXiv:2212.14546*, 2022. 7

[65] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, Candace Ross, Adam Polyak, Russell Howes, Vasu Sharma, Puxin Xu, Hovhannes Tamoyan1, Oron Ashual, Uriel Singer, Shang-Wen Li, Susan Zhang Gargi Ghosh, Yaniv Taigman, Maryam Fazel-Zarandi, Asli Celikyilmaz, Luke Zettlemoyer, and Armen Aghajanyan. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. In *https://ai.meta.com/research/publications/scaling-autoregressive-multi-modal-models-pretraining-and-instruction-tuning/*, 2023. 2

[66] 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. 7

[67] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. 2021. 2, 7

[68] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *CVPR*, 2022. 2

[69] Wentao Zhu, Jingru Yi, Xiaohang Sun, Xiang Hao, Linda Liu, and Mohamed Omar. Multiscale multimodal transformer for multimodal action recognition. In *ICLR*, 2022. 8