Attention Bottlenecks for Multimodal Fusion

Anonymous Author(s)
Affiliation
Address
email

Abstract

Humans perceive the world by concurrently processing and fusing high-dimensional inputs from multiple modalities such as vision and audio. Machine perception models, in stark contrast, are typically modality-specific and optimised for unimodal benchmarks. A common approach for building multimodal models is to simply combine multiple of these modality-specific architectures using late-stage fusion of final representations or predictions (*'late-fusion'*). Instead, we introduce a novel transformer based architecture that uses 'attention bottlenecks' for modality fusion at multiple layers. Compared to traditional pairwise self-attention, these bottlenecks force information between different modalities to pass through a small number of 'bottleneck' latent units, requiring the model to collate and condense the most relevant information in each modality and only share what is necessary. We find that such a strategy improves fusion performance, at the same time reducing computational cost. We conduct thorough ablation studies, and achieve state-of-the-art results on multiple audio-visual classification benchmarks including Audioset, Epic-Kitchens and VGGSound. All code and models will be released.

1 Introduction

Overlapping multimodal sensations are a crucial enabler of human perceptual learning [42]. For artificial learning systems, however, designing a unified model for modality fusion is challenging due to a number of factors: (i) variations in learning dynamics between modalities [48], (ii) different noise topologies, with some modality streams containing more information for the task at hand than others, as well as (iii) specialised input representations. The difference in input representations between audio and vision is particularly stark – many state of the art audio classification methods rely on short term Fourier analysis to produce log-mel spectrograms, often using them as inputs to CNN architectures designed for images [22, 41]. These time-frequency representations have different distributions to images – multiple acoustic objects can have energy at the same frequency, and the translation invariances of CNNs may no longer be a desired property (while an acoustic object can be shifted in time, a shift in frequency could alter the meaning entirely). In contrast, the visual stream in a video is three-dimensional (two spatial and one temporal), and while different spatial regions of the image correspond to different objects, there is the unique challenge of high redundancy across multiple frames. Hence input representations, and consequently neural network architectures and benchmarks tend to vary wildly for different modalities. For simplicity, the dominant paradigm for multimodal fusion therefore often consists of an ad-hoc scheme that involves integrating separate audio and visual networks via their output representations or scores i.e. 'late-fusion' [19, 38].

In this work, we present a new transformer based model for audiovisual fusion in video. Despite originally being proposed for NLP tasks, there has been recent interest in transformers [46] as universal perceptual models [25], due to their ability to model dense correlations between tokens, at the same time making few assumptions about their inputs (and because continuous perceptual inputs can be tokenised). By dividing dense continuous signals into patches and rasterising them

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

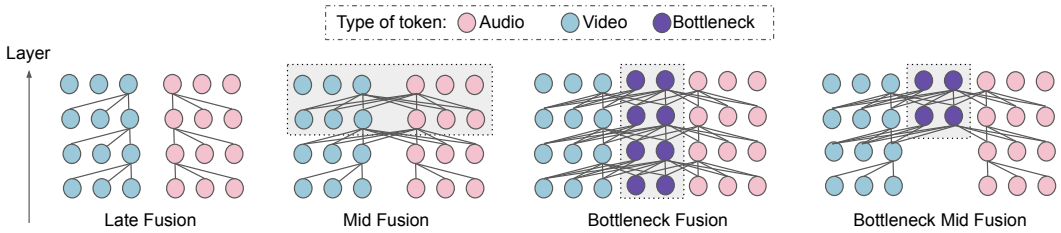

Figure 1: **Cross-modal Fusion**. Unlike late fusion (left), where no cross-modal information is exchanged in the model until after the classifier, we investigate two pathways for the exchange of cross-modal information. The first is via standard pairwise self attention across all hidden units in a layer, but applied only to later layers in the model – mid fusion (middle, left). We also propose the use of 'fusion bottlenecks' (middle, right) that restrict attention flow within a layer through tight latent units. Both forms of restriction can be applied in conjunction (Bottleneck Mid Fusion) for optimal performance (right). We show $B = 2$ bottleneck units and 3 hidden units per modality. Grey boxes indicate tokens that receive attention flow from both audio and video tokens.

to 1D tokens, transformers have been shown to perform competitively for image (ViT [12]) and video classification (ViViT [5]), and more recently, audio classification (AST [20]). Because these models are able to elegantly handle variable length sequences, a natural first extension would be to feed in a sequence of both visual and auditory patches to a transformer, with minimal changes to the architecture. This 'early fusion' model allows free attention flow between different spatial and temporal regions in the image, as well as across frequency and time in the audio spectrogram. While theoretically appealing, we hypothesise that full pairwise attention at all layers of the model is not necessary because audio and visual inputs contain dense, fine-grained information, much of which is redundant. This is the case particularly for video, as shown by the performance of 'factorised' versions of [5]. Such a model would also not scale well to longer videos due to the quadratic complexity of pairwise attention with the sequence length of tokens, which would increase as more modalities are introduced to the model. To mitigate this, we propose two methods to restrict the flow of attention in our model. The first follows from a common paradigm in multimodal learning, which is to restrict cross-modal flow to later layers of the network, allowing early layers to specialise in learning and extracting unimodal patterns. Henceforth this is is referred to as 'mid fusion' (Fig. 1, middle left), where the layer at which cross-modal interactions are introduced is called the 'fusion layer'. The two extreme versions of this are 'early fusion' (all layers are cross-modal) and 'late fusion' (all are unimodal) which we compare to as a baselines. Our second method (and main contribution), is to restrict cross-modal attention flow between tokens *within* a layer. We do this by allowing free attention flow within a modality, but force our model to collate and 'condense' information from each modality before sharing it with the other. The core idea is to introduce a small set of latent *fusion* units that form an 'attention bottleneck', through which cross-modal interactions within a layer must pass. We demonstrate that this 'bottlenecked' version, which we name Multimodal Bottleneck Transformer (*MBT*), outperforms or matches its unrestricted counterpart, but with lower computational cost.

Concretely, we make the following contributions: (i) We propose a new architecture (*MBT*) for audiovisual fusion. Our model restricts the flow of cross-modal information between latent units through tight fusion 'bottlenecks', that force the model to collect and 'condense' the most relevant inputs in each modality (and therefore share only that which is necessary with the other modality). This avoids the quadratic scaling cost of full pairwise attention, and leads to performance gains with less compute; (ii) We apply MBT to image and spectogram patches (Fig. 2), and explore a number of ablations related to restricting fusion to later layers of the network, the sampling of inputs and data size; and finally (iii) We set the new state-of-the-art for video classification across a number of popular audio-visual benchmarks, including AudioSet [18], Epic-Kitchens100 [9] and VGGSound [7].

2 Related Work

Audiovisual Learning: Audiovisual multimodal learning has a rich history, both before and during the deep learning era [40]. Given the limited available data and computational resources, early work focused on relatively simple early-stage (e.g. stacking hand-designed features) and late-stage (e.g.

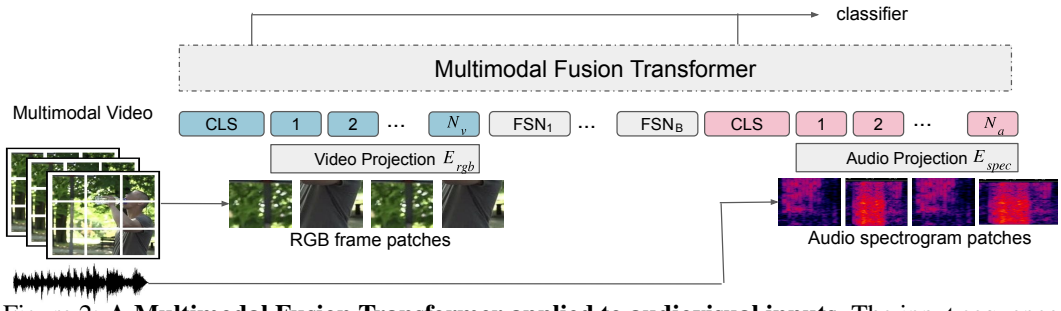

Figure 2: **A Multimodal Fusion Transformer applied to audiovisual inputs.** The input sequence consists of image and spectrogram patches. Our transformer encoder then uses self and cross-attention operations to model both unimodal and cross-modal information at multiple layers in the network, without imposing any modality specific priors.

score fusion) techniques [8]. Deep learning has provided a platform for more sophisticated strategies in which powerful modality-specific or joint latent representations are implicitly learned to mediate the fusion. The result has enabled major advances in a range of downstream supervised audiovisual tasks [37, 29, 13]. In the supervised setting, multiple modality-specific convolution networks can be jointly trained, whose intermediate activations are then combined by summation [27] or via 'lateral connections' [49]. These methods often assume the activations to be combined are strictly aligned in time. In the unsupervised setting, a popular pretraining task is to learn to synchronise signals from different modalities via a contrastive loss [3, 4, 6, 38, 26, 1, 2], however each modality is usually encoded separately under this setup.

Multimodal Transformers: The self attention mechanism of transformers provides a natural bridge to connect multimodal signals. Applications include audio enhancement [13, 45], speech recognition [21], image segmentation [50, 45], cross-modal sequence generation [17, 34, 32], video retrieval [16] and image/video captioning/classification [36, 44, 43, 31, 24]. For many works, the inputs to transformers are the output representations of single modality CNNs [30, 16] – unlike these works we use transformer blocks throughout, using only a single convolutional layer to rasterise 2D patches. The tokens from different modalities are usually combined directly as inputs to the transformers [33], for example, the recently released Perceiver model [25] introduces an iterative attention mechanism which takes concatenated raw multimodal signals as inputs, which corresponds to our 'early fusion' baseline. In contrast, we carefully examine the impact of different modality fusion strategies, including limiting cross-modal attention flow to later layers of our model, and 'channeling' cross-modal connections through bottlenecks in our proposed Multimodal Bottleneck Transformer (MBT).

3 Multimodal Fusion Transformers

In this section we describe our proposed Multimodal Bottleneck Transformer (MBT). We begin by summarising the recently proposed Vision Transformer (ViT) [12] and Audio Spectrogram Transformer (AST) [20], developed for image and audio classification respectively, in Sec. 3.1. We then describe our extension to the audio-visual case. We discuss three different token fusion strategies (Sec. 3.2), and finally discuss the fusion pathway in the entire model (Sec. 3.3), which involves restricting multimodal fusion to certain layers of the model.

3.1 The ViT and AST architectures

Vision Transformer (ViT) [12] (and a recent extension to audio – Audio Spectrogram Transformer (AST) [20]) adapts the Transformer architecture [46], originally designed for natural language processing, to process 2D inputs with minimal changes. The key insight is to extract N non-overlapping patches from the RGB image (or the audio spectrogram), $x_i \in \mathbb{R}^{h \times w}$, and convert them into a series of 1D tokens $z_i \in \mathbb{R}^d$, as follows:

$$\mathbf{z} = g(\mathbf{x}; \mathbf{E}, z_{\text{cls}}) = [z_{\text{cls}}, \mathbf{E}x_1, \mathbf{E}x_2, ..., \mathbf{E}x_N] + \mathbf{p}. \tag{1}$$

Here, $\mathbf{E}$ is a linear projection mapping each token to $\mathbb{R}^d$, z_{cls} is a special token prepended to this sequence so that its representation at the final layer can be passed to a classifier for classification tasks [11], and $\mathbf{p} \in \mathbb{R}^{(N+1)\times d}$ is a learned positional embedding added to the tokens to retain positional information (as all subsequent self-attention operations are permutation invariant).

The tokens are then passed through an encoder consisting of a sequence of L transformer layers. Each transformer layer consists of Multi-Headed Self-Attention (MSA), Layer Normalisation (LN) and Multilayer Perceptron (MLP) blocks applied using residual connections. We denote a transformer layer, $\mathbf{z}^{l+1} = \text{Transformer}(\mathbf{z}^l)$ as

$$\mathbf{y}^l = \text{MSA}(\text{LN}(\mathbf{z}^l)) + \mathbf{z}^l \tag{2}$$

$$\mathbf{z}^{l+1} = \text{MLP}(\text{LN}(\mathbf{y}^l)) + \mathbf{y}^l. \tag{3}$$

Here, the MSA operation [46] computes dot-product attention [46] where the queries, keys and values are all linear projections of the same tensor, $\text{MSA}(\mathbf{X}) = \text{Attention}(\mathbf{W}^Q\mathbf{X}, \mathbf{W}^K\mathbf{X}, \mathbf{W}^V\mathbf{X})$. We further define cross-attention between two tensors, $\mathbf{X}$ and $\mathbf{Y}$, where $\mathbf{X}$ forms the query and $\mathbf{Y}$ forms the keys and values which are used to reweight the query as $\text{MCA}(\mathbf{X}, \mathbf{Y}) = \text{Attention}(\mathbf{W}^Q\mathbf{X}, \mathbf{W}^K\mathbf{Y}, \mathbf{W}^V\mathbf{Y})$. This will be used in our multimodal case, as described next.

3.2 Multimodal Transformer

We now describe our extension to the multimodal case. We begin by discussing three different token fusion strategies.

3.2.1 Fusion via Vanilla Self-Attention

We begin by describing a 'vanilla' fusion model, which simply consists of the regular transformer applied to multimodal inputs. Our method of tokenising video is straightforward – given a video clip of length t seconds, we uniformly sample F RGB frames and convert the audio waveform into a single spectrogram. We then embed each frame and the spectrogram independently following the encoding proposed in ViT [12], and concatenate all tokens together into a single sequence.

Formally, if we have extracted a total of N_v RGB patches from all F sampled frames, $\mathbf{x}_{\text{rgb}} \in \mathbb{R}^{N_v \times d}$, and N_a spectrogram patches, $\mathbf{x}_{\text{spec}} \in \mathbb{R}^{N_a \times d}$, our sequence of tokens is

$$\mathbf{z} = [\mathbf{z}_{\text{rgb}} || \mathbf{z}_{\text{spec}}] \quad \text{where} \quad \mathbf{z}_{\text{rgb}} = g(\mathbf{x}_{\text{rgb}}; \mathbf{E}_{\text{rgb}}, z_{\text{cls-rgb}}) \quad \text{and} \quad \mathbf{z}_{\text{spec}} = g(\mathbf{x}_{\text{spec}}; \mathbf{E}_{\text{spec}}, z_{\text{cls-spec}}). \tag{4}$$

Here, $[\mathbf{z}_{\text{rgb}} || \mathbf{z}_{\text{spec}}]$ denotes the concatenation of the tokens for each modality. We use different projections $\mathbf{E}_{\text{rgb}}$ and $\mathbf{E}_{\text{spec}}$ for RGB and spectrogram patches respectively, and prepend a separate classification token for each modality.

Our multimodal encoder then applies a series of transformer layers in the same manner as above. Attention is allowed to flow freely through the network, i.e. each RGB token can attend to all other RGB and spectrogram tokens since $\mathbf{z}^{l+1} = \text{Transformer}(\mathbf{z}^l; \theta)$ with model parameters θ.

3.2.2 Fusion with Modality-specific Parameters

We can generalise this model by allowing each modality to have its own dedicated parameters θ_{rgb} and θ_{spec}, but still exchange information via the attention mechanism. For this purpose, we define a Cross-Transformer layer:

$$\begin{aligned} \mathbf{z}^{l+1}_{\text{rgb}} &= \text{Cross-Transformer}(\mathbf{z}^l_{\text{rgb}}, \mathbf{z}^l; \theta_{\text{rgb}}) \\ \mathbf{z}^{l+1}_{\text{spec}} &= \text{Cross-Transformer}(\mathbf{z}^l_{\text{spec}}, \mathbf{z}^l; \theta_{\text{spec}}) \end{aligned} \tag{5}$$

Where the Cross-Transformer employs the generalised cross-attention operation that takes two sets of inputs $\mathbf{z}_1$ and $\mathbf{z}_2$ that are not necessarily overlapping. This layer follows the original transformer layer with the difference being that Eq. 2 becomes

$$\mathbf{y}^l = \text{MCA}(\text{LN}(\mathbf{z}^l_1), \text{LN}(\mathbf{z}^l_2)) + \mathbf{z}^l_1. \tag{6}$$

Finally, note that we have explicitly defined the parameters, θ_{rgb} and θ_{spec} of the cross-transformer layers in Eq. 5 as they are different for each modality. However, when θ_{rgb} and θ_{spec} are equal, ($\theta_{\text{rgb}} = \theta_{\text{spec}} = \theta$), the computation defined in Eq. 5 is equivalent to Sec. 3.2.1.

3.2.3 Fusion via Attention Bottlenecks

In order to tame the quadratic complexity of pairwise attention, we next introduce a small set of B fusion bottleneck tokens $\mathbf{z}_{\text{fsn}} = [z^1_{\text{fsn}}, z^2_{\text{fsn}}, \ldots, z^B_{\text{fsn}}]$ to our input sequence (see Fig. 2). The input sequence is now

$$\mathbf{z} = [\mathbf{z}_{\text{rgb}} || \mathbf{z}_{\text{fsn}} || \mathbf{z}_{\text{spec}}]. \tag{7}$$

We then restrict all cross-modal attention flow in our model to be via these bottleneck tokens. More formally for layer l, we compute token representations as follows:

$$[\mathbf{z}^{l+1}_{\text{rgb}} || \hat{\mathbf{z}}^{l+1}_{\text{fsn}}] = \text{Transformer}([\mathbf{z}^{l}_{\text{rgb}} || \mathbf{z}^{l}_{\text{fsn}}]; \theta_{\text{rgb}}) \tag{8}$$

$$[\mathbf{z}^{l+1}_{\text{spec}} || \mathbf{z}^{l+1}_{\text{fsn}}] = \text{Transformer}([\mathbf{z}^{l}_{\text{spec}} || \hat{\mathbf{z}}^{l+1}_{\text{fsn}}]; \theta_{\text{spec}}) \tag{9}$$

Here $\mathbf{z}_{\text{rgb}}$ and $\mathbf{z}_{\text{spec}}$ can only exchange information via the bottleneck $\mathbf{z}_{\text{fsn}}$ within a transformer layer. The bottleneck tokens $\mathbf{z}_{\text{fsn}}$ are updated twice, first with visual information (Equation 8), and then with audio information (Equation 9). We keep the number of bottleneck tokens in the network to be much smaller than the total number of latent units per modality ($B \ll N_v$ and $B \ll N_a$). Because all cross-modal attention flow must pass through these units, these tight 'fusion' bottlenecks force the model to condense information from each modality and share only that which is necessary. As we show in the experiments, this increases or maintains performance for multimodal fusion, at the same time reducing computational complexity.

3.3 Where to Fuse: Early, Mid and Late

The above strategies discuss fusion within a layer, and in most transformer architectures (such as ViT), every layer consists of an identical set of operations. A common paradigm in multimodal learning, however, is to restrict early layers of a network to focus on unimodal processing, and only introduce cross-modal connections at later layers. This is conceptually intuitive if we believe lower layers are involved in processing low level features, while higher layers are focused on learning semantic concepts – low-level visual features such as edges and corners in images might not have a particular sound signature, and therefore might not benefit from early fusion with audio [49].

This can be implemented with our model as follows: We initially perform vanilla self-attention among tokens from a single modality for L_f layers. Thereafter, we concatenate all latent tokens together, $\mathbf{z}^{L_f} = [\mathbf{z}^{L_f}_{\text{rgb}} || \mathbf{z}^{L_f}_{\text{spec}}]$ and pass them through the remaining $L - L_f$ layers where the tokens are fused according to Sec. 3.2. Here, $L_f = 0$ corresponds to an 'early-fusion' model, $L_f = L$ a 'late-fusion' model, and $0 < L_f < L$ a 'mid-fusion' one. More formally, this can be denoted as

$$\begin{aligned} \mathbf{z}^{l+1}_{\text{rgb}} &= \text{Transformer}(\mathbf{z}^{l}_{\text{rgb}}; \theta_{\text{rgb}}), \mathbf{z}^{l+1}_{\text{spec}} = \text{Transformer}(\mathbf{z}^{l}_{\text{spec}}; \theta_{\text{spec}}) && \text{if } l < L_f \\ \mathbf{z}^{l} &= [\mathbf{z}^{l}_{\text{rgb}} || \mathbf{z}^{l}_{\text{spec}}], \qquad \mathbf{z}^{l+1} = \text{Multimodal-Transformer}(\mathbf{z}^{l}; \theta_{\text{spec}}, \theta_{\text{rgb}}) && \text{otherwise} \end{aligned}$$

where $\text{Multimodal-Transformer}(\cdot)$ can refer to either of the 3 fusion strategies described in Sec 3.2.

3.4 Classification

For all model variants described above, we pass output representations of the CLS tokens $z^L_{\text{cls-rgb}}$ and $z^L_{\text{cls-spec}}$ to the same linear classifier and average the pre-softmax logits.

4 Experiments

In this section we first describe the datasets and their respective evaluation protocols (Sec. 4.1), then discuss implementation details (Sec. 4.2). We then ablate the key design choices in our model (Sec. 4.3), before finally comparing our model to the state of the art (Sec. 4.5).

4.1 Datasets and Evaluation Protocol

AudioSet [18] consists of almost 2 million 10-second video clips from YouTube videos, annotated with 527 class labels. We only use the clips still available on YouTube, giving us 20,361 clips for the balanced train set (henceforth referred to as mini-AudioSet or miniAS) and 18,589 clips for the

test set. We note that while we are unable to obtain the full test set, ours is exactly the same as recent works we compare to, including Perceiver [25], the current state of the art for audiovisual classification on this dataset. While the unbalanced training set is large (almost 2 million samples), we train on a (slightly more) balanced subset consisting of 500K samples (henceforth referred to as AS-500K). More details are provided in the supplementary materials. Because each sample has multiple labels, we train with a binary cross-entropy (BCE) loss and report mean average precision (mAP) over all classes, as following standard practice.

VGGSound [7] contains almost 200K video clips of length 10s, annotated with 309 sound classes consisting of human actions, sound-emitting objects and human-object interactions. Unlike AudioSet, the sound source for each clip is 'visually present' in the video. This was ensured during dataset creation through the use of image classifiers. After filtering clips that are no longer available on YouTube, we end up with 172,427 training and 14,448 test clips. We train with a standard cross-entropy loss for classification and report Top-1 and Top-5 classification accuracy.

Epic-Kitchens 100 [9] consists of egocentric videos capturing daily kitchen activities. The dataset consists of 90,000 variable length clips spanning 100 hours. We report results for action recognition following standard protocol [9] - each action label is a combination of a verb and noun, and we predict both using a single network with two 'heads', both trained with a cross-entropy loss. The top scoring verb and action pair predicted by the network are used, and Top-1 action accuracy is the primary metric. Actions are mainly short-term (average length is 2.6s with minimum length 0.25s).

4.2 Implementation Details

Our backbone architecture follows that of ViT [12] identically, specifically we use ViT-Base (ViT-B, $L = 12$, $N_H = 12$, $d = 3072$)[1] initialised from ImageNet-21K [10], however we note that our method is agnostic to backbone and can be applied to any other transformer backbone. Unless otherwise specialised, we use $B = 4$ bottleneck tokens for all experiments with bottleneck fusion. We randomly sample clips of t seconds for training. RGB frames for all datasets are extracted at 25 fps. For AudioSet and VGGSound we sample 8 RGB frames over the sampling window of length t with a uniform stride of length $(t \times 25)/8$. We extract 16×16 patches from each frame of size 224×224, giving us a total of $8 \times 14 \times 14 = 1568$ patches per video. For Epic-Kitchens (because the segments are shorter), we sample 32 frames with stride 1. Audio for all datasets is sampled at 16kHz and converted to mono channel. Similar to [20], we extract log mel spectrograms with a frequency dimension of 128 computed using a 25ms Hamming window with hop length 10ms. This gives us an input of size $128 \times 100t$ for t seconds of audio. Spectrogram patches are extracted with size 16×16, giving us $80 \times 5 = 400$ patches for 8 seconds of audio. For images we apply the standard data augmentations used in [5] (random crop, flip, colour jitter), and for spectrograms we use SpecAugment [39] with a max time mask length of 192 frames and max frequency mask length of 48 bins following AST [20]. We also apply Mixup [51] with $\alpha = 0.3$ and stochastic depth regularisation [23] with probability $p = 0.3$. All models (across datasets) are trained with a batch size of 64, synchronous SGD with momentum of 0.9, and a cosine learning rate schedule with warmup of 2.5 epochs on TPU accelerators. We set the base learning rate to 0.5 and train for 50 epochs.

Inference: Following standard practice, we uniformly sample multiple temporal crops from the clip and average per-view logits to obtain the final result. The number of test crops is set to 3.

4.3 Ablation Analysis

In this section we investigate the impact of the different architectural choices in MBT. Unless otherwise specified, we use mini-AudioSet for training and AudioSet test for evaluation.

4.3.1 Fusion Strategies

We implement all the three fusion strategies described in Sec. 3.2: (i) **Vanilla self-attention** – Unrestricted pairwise attention between all latent units within a layer; (ii) **Vanilla cross-attention with separate weights:** Same as above, but we now have separate weights for each modality. The latent units are updated via pairwise attention with all other latent units from both modalities; and finally (iii) **Bottleneck fusion:** In this case all cross-modal attention must pass through fixed bottleneck FSN latent units. Note that these only describes attention flow between tokens within

[1] L is the number of transformer layers, N_H is the number of self-attention heads with hidden dimension d.

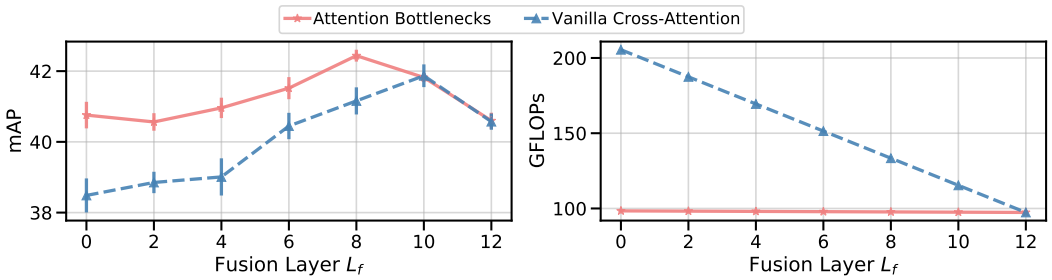

Figure 3: The impact of using attention bottlenecks for fusion on performance (left) and compute (right) at different fusion layers L_f on AudioSet, using clip span $t = 4$ and $B = 4$ bottleneck tokens.

a layer. For strategies (ii) and (iii), we also conduct experiments showing the impact of restricting cross-modal attention to layers after a fixed fusion layer L_f. We investigate models with different fusion layers, $L_f = 0, 2, 4, 6, 8, 10, 12$, and present the results in Fig. 3.[2]

Sharing Weights for both Modalities: We first investigate the impact of sharing the encoder weights for both modalities (strategy (i) vs (ii)). We find that performance is boosted slightly by using different weights for the audio and video encoders (see supplementary material for details), and so we use separate modality weights for further experiments.

Fusion Layer: We then investigate the impact of varying the fusion layer L_f, for the latter two strategies: (ii) Vanilla Cross-Attention and (iii) Bottleneck Fusion. We conduct experiments with $L_f = 0, 2, 4, 6, 8, 10, 12$. We fix the input span t to 4s and the number of bottleneck tokens B to 4. We conduct 3 runs for each experiment and report mean and std deviation. As can be seen from Fig. 3 (left), 'mid fusion' outperforms both early ($L_f = 0$) and late fusion ($L_f = 12$), with optimal performance obtained by using fusion layer $L_f = 10$ for vanilla cross-attention and $L_f = 8$ for bottleneck attention. This suggests that the model benefits from restricting cross-modal connections to later layers, allowing earlier layers to specialise to learning unimodal features.

Attention Bottlenecks: In Fig. 3, we also examine the effect of bottleneck attention vs vanilla cross-attention for multimodal fusion. We find that for all values of L_f restricting flow to bottlenecks improves or maintains performance, with improvements more prominent at lower values of L_f. At $L_f = 10$, both perform similarly, note that at this stage we only have 3 fusion layers in the model. Our best performing model uses attention bottlenecks with $L_f = 8$, and we fix this for all further experiments. We also compare the amount of computation, measured in GFLOPs, for both fusion strategies (Fig. 3, right). Using a small number of bottleneck tokens (in our experiments $B = 4$) adds negligible extra computation over a late fusion model, with computation remaining largely constant with varying fusion layer L_f. This is in contrast to vanilla cross-fusion, which has a non-negligible computational cost for every layer it is applied to. We note that for early fusion ($L_f = 0$), bottleneck fusion outperforms vanilla cross-attention by over 2 mAP, with less than half the computational cost.

Number of bottleneck tokens B: We experiment with $B = 4, 36, 64, 256$ and 1024, and find that performance is relatively consistent (all within 0.5 mAP). We hence fix the number of tokens to $B = 4$ for all experiments. It is interesting that with such a small number of cross-modal connections through only 4 hidden units ($B = 4$) at each cross-modal layer, we get large performance gains over late fusion (Fig. 3), highlighting the importance of allowing cross-modal information to flow at multiple layers of the model.

4.3.2 Input Sampling and Dataset Size

In this section we investigate the impact of different modality sampling strategies. We also compare to single modality baselines – the visual-only and audio-only baselines consist of a vanilla transformer model applied to only the RGB or spectrogram patches respectively.

Sampling window size t: An advantage of our transformer based model is that we can easily input variable length token sequences. We experiment with varying the sampling window t with the following values $t = 2, 4, 6$ and 8 seconds (note that all videos in AudioSet are 10s), and show results in Fig. 4[3]. At inference, we uniformly sample multiple windows covering the entire video. While

[2]Note that $L_f = 12$ refers to late fusion, where scores are only aggregated after the classifier, and neither strategy (ii) nor (iii) is applied, but we show results on the same plot for convenience.

[3]Averaged over 3 runs. Because error bars are small in the plot we provide them in suppl. materials as well.

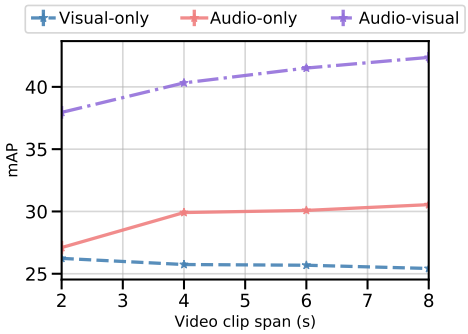

Figure 4: The effect of varying input clip span t on the AudioSet test set.

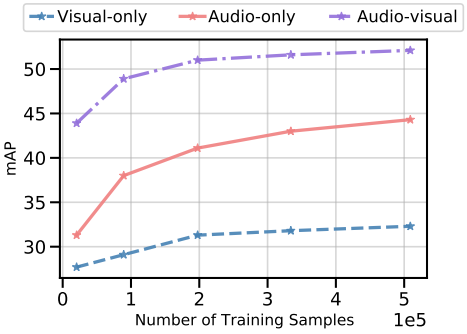

Figure 5: The effect of training data size on the AudioSet test set.

the number of spectrogram patches N_a changes with t, we keep the number of RGB patches N_v fixed by changing the stride of frames (to avoid running out of memory). Our results indicate that the performance of both the audio and audio-visual fusion model increases with input span, however the performance of the visual-only model slightly decreases (we hypothesize that this is due to the increased fixed stride, meaning fewer frames are randomly sampled during training). We fix $t = 8s$ in all further experiments.

Synchronous vs Asynchronous sampling: Given that auditory and visual events may not always be perfected aligned in videos [27], we also investigate asynchronous sampling of different modalities. Here input windows are sampled independently from the entire video clip for each modality. We found asynchronous sampling to slightly hurt performance (details in supplementary material), and so we use synchronised sampling for all further experiments.

Modality MixUp: While applying Mixup regularization [51] to training, we note that there are two different ways to apply it for multimodal inputs – the standard approach is to sample one set of mixup weights using the probability parameter α, and use it to generate all virtual modality-label pairs [51]. In addition, we also explore a modified version which we call *modality mixup*, which samples an independent weight for each modality. Modality mixup imposes stronger augmentation than standard mixup, leading to a slight performance improvement (from 42.6 mAP to 43.9 mAP) on AudioSet.

Impact of Dataset Size: We show the impact of varying the number of training samples in Fig. 5, and find a monotonic increase with dataset size (more steeply for audio-only than visual-only).

4.4 Comparison to single modality performance

We compare MBT to visual-only and audio-only baselines on AudioSet (Table 1), Epic-Kitchens (Table 2a) and VGGSound (Table 2b). Note we use the best parameters obtained via the ablations above, i.e. bottleneck fusion with $t = 8$, $B = 4$, $F_l = 8$ and modality mixup. For all datasets, multimodal fusion outperforms the higher-performing single modality baseline, demonstrating the value of complementary information. We note that the relative importance of modalities for the classification labels varies (audio-only has higher relative performance for AudioSet and lower for Epic-Kitchens, while both audio and visual baselines are equally strong for VGGSound). This is (unsurprisingly) largely a function of the dataset annotation procedure and positions VGGSound as a uniquely suitable dataset for fusion.

4.5 Comparison to state of the art

We compare MBT to previous fusion methods on AudioSet in Table 1. For audio-visual fusion, we outperform all previous works (even though we only train on 500K samples which is a quarter of the data used by the others), including Attn Audio-Visual [14] which uses self-attention fusion on top of individual modality CNNs, and even outperforming Perceiver by almost +5 mAP (10.1% relative improvement). We compare to previous video classification methods on Epic-Kitchens in Table 2a, and note that our model outperforms all previous works that use vision only, as well as TBN [27] which uses three modalities - RGB, audio and optical flow. Given VGGSound is a relatively

Model	Training Set	A only	V only	AV Fusion
GBlend [48]	MiniAS	29.1	22.1	37.8
GBlend [48]	FullAS-2M	32.4	18.8	41.8
Attn Audio-Visual [14]	FullAS-2M	38.4	25.7	46.2
Perceiver [25]	FullAS-2M	44.9	38.0	47.3
MBT	MiniAS	31.3	27.7	43.9
MBT	AS-500K	44.3	32.3	**52.1**

Table 1: **Comparison to the state of the art on AudioSet [18].** We report mean average precision (mAP). For audio-visual fusion, our method outperforms others that use the entire AudioSet training set (almost 2M samples), while we train on only 500K.

Model	Mods	Verb	Noun	Action
Damen et al. [9]	A	42.1	21.5	14.8
AudioSlowFast [28]†	A	46.5	22.78	15.4
TSN [47]	V, F	60.2	46.0	33.2
TRN [52]	V, F	65.9	45.4	35.3
TBN [27]	A, V, F	66.0	47.2	36.7
TSM [35]	V, F	**67.9**	49.0	38.3
SlowFast [15]	V	65.6	50.0	38.5
MBT	A	44.3	22.4	13.0
MBT	V	62.0	56.4	40.7
MBT	A, V	64.8	**58.0**	**43.4**

(a) Epic Kitchens 100 [9]

Model	Mods	Top-1 Acc	Top-5 Acc
Chen et al‡ [7]	A	48.8	76.5
AudioSlowFast‡ [28]	A	50.1	77.9
MBT	A	52.3	78.1
MBT	V	51.2	72.6
MBT	A,V	**64.1**	**85.6**

(b) VGGSound [7].

Table 2: **Comparison to the state of the art**. We report Top-1 accuracy on Epic Kitchens 100 and Top-1 and Top-5 accuracy on VGGSound. Modalities (Mods) are **A:** Audio, **V:** Visual, **F:** Optical flow. † Uses pretraining on VGGSound. ‡ We calculate metrics on our test set for a fair comparison using the scores provided by the authors.

new dataset, we compare to two existing audio-only works[4] (Table 2b), and set the first audiovisual benchmark (that we are aware of) on this dataset.

5 Conclusion

We propose a new transformer architecture (*MBT*) for audiovisual fusion, and explore a number of different fusion strategies using cross-attention between latent tokens. We propose a novel strategy to restrict cross-modal attention via a small set of fusion 'bottlenecks', and demonstrate that this improves performance over vanilla cross-attention at lower computational cost, achieving state of the art results on a number of benchmarks.

Limitations: Our study is limited to audio-visual learning for video classification. We note that the fusion strategies investigated in MBT are agnostic to the inputs and may be applied to other modalities such as text or optical flow. Our bottlenecks can also be used with more than 2 modalities in the same model. We also only explore fully supervised fusion, and future work will tackle extensions to a self-supervised learning framework.

Broader Impact: Multimodal fusion strategies are important for machine learning, as fusing complementary information from different modalities can increase robustness when applied to real world applications. We also note that transformers are in general compute-heavy, which can have adverse environmental effects. We propose a token fusion method via bottlenecks that helps reduce computational complexity when applying transformers for multimodal fusion. Finally, we observe that training datasets contain biases that may render models trained on them unsuitable for certain applications. It is thus possible that people use classification models (intentionally or not) to make decisions that impact different groups in society differently, and it is important to keep this in mind when deploying, analysing and building upon these models.

[4] To fairly compare to these works, we obtain the scores on the full VGGSound test set from the authors, and compute accuracy metrics on our slightly smaller test set as described in Sec. 4.1.

References

[1] Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text, 2021.

[2] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020.

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

[4] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *ECCV*, 2018.

[5] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021.

[6] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016.

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

[8] Tsuhan Chen and Ram R Rao. Audio-visual integration in multimodal communication. *Proceedings of the IEEE*, 86(5):837–852, 1998.

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018.

[14] Haytham M Fayek and Anurag Kumar. Large scale audiovisual learning of sounds with weakly labeled data. *arXiv preprint arXiv:2006.01595*, 2020.

[15] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[16] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision (ECCV)*, volume 5. Springer, 2020.

[17] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. *arXiv preprint arXiv:2007.10984*, 4(6):7, 2020.

[18] Jort F Gemmeke, Daniel PW 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017.

[19] Bernard Ghanem, Juan Carlos Niebles, Cees Snoek, Fabian Caba Heilbron, Humam Alwassel, Victor Escorcia, Ranjay Krishna, Shyamal Buch, and Cuong Duc Dao. The activitynet large-scale activity recognition challenge 2018 summary. *arXiv preprint arXiv:1808.03766*, 2018.

[20] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[21] David Harwath, Antonio Torralba, and James R Glass. Unsupervised learning of spoken language with visual context. 2017.

[22] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[23] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[24] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 958–959, 2020.

[25] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021.

[26] Aren Jansen, Daniel PW Ellis, Shawn Hershey, R Channing Moore, Manoj Plakal, Ashok C Popat, and Rif A Saurous. Coincidence, categorization, and consolidation: Learning to recognize sounds with minimal supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125. IEEE, 2020.

[27] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501, 2019.

[28] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Slow-fast auditory streams for audio recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 855–859. IEEE, 2021.

[29] Yelin Kim, Honglak Lee, and Emily Mower Provost. Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 3687–3691. IEEE, 2013.

[30] Sangho Lee, Youngjae Yu, Gunhee Kim, Thomas Breuel, Jan Kautz, and Yale Song. Parameter efficient multimodal transformers for video representation learning. *arXiv preprint arXiv:2012.04124*, 2020.

[31] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8928–8937, 2019.

[32] Jiaman Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li. Learning to generate diverse dance motions with transformer. *arXiv preprint arXiv:2008.08171*, 2020.

[33] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[34] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation. *arXiv preprint arXiv:2101.08779*, 2021.

[35] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. 2019 ieee. In *CVF International Conference on Computer Vision (ICCV)*, pages 7082–7092, 2019.

[36] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[37] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *ICML*, 2011.

[38] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.

[39] 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.

[40] Dhanesh Ramachandram and Graham W Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.

[41] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

[42] Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29, 2005.

[43] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019.

[44] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019.

[45] Efthymios Tzinis, Scott Wisdom, Aren Jansen, Shawn Hershey, Tal Remez, Daniel PW Ellis, and John R Hershey. Into the wild with audioscope: Unsupervised audio-visual separation of on-screen sounds. *arXiv preprint arXiv:2011.01143*, 2020.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[47] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[48] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705, 2020.

[49] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

[50] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10502–10511, 2019.

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

[52] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] All data is public, and we plan to release code and pretrained models in the near future.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.2.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used three, publicly available datasets that are common in the community and cited the original dataset authors, as described in Sec. 4.1. We have not collected any new data in this paper.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]