Audiovisual SlowFast Networks for Video Recognition

Fanyi Xiao[1,2*] Yong Jae Lee[1] Kristen Grauman[2] Jitendra Malik[2] Christoph Feichtenhofer[2]

[1]University of California, Davis [2]Facebook AI Research (FAIR)

Abstract

We present Audiovisual SlowFast Networks, an architecture for integrated audiovisual perception. AVSlowFast extends SlowFast Networks [13] with a Faster Audio pathway that is deeply integrated with its visual counterparts, by fusing audio and visual features at multiple layers, enabling audio to contribute to the formation of hierarchical audiovisual concepts. To overcome training difficulties that arise from different learning dynamics for audio and visual modalities, we employ DropPathway that randomly drops the Audio pathway during training as a simple and effective regularization technique. Inspired by prior studies in neuroscience, we perform hierarchical audiovisual synchronization and show that it leads to better audiovisual features. We report state-of-the-art results on four video action classification and detection datasets, perform detailed ablation studies, and show the generalization of AVSlowFast to self-supervised tasks, where it is improves over prior work without bells and whistles.

1. Introduction

Joint audiovisual learning is core to human perception. However, most contemporary models for video analysis exploit only the visual signal and ignore the audio signal. For many video understanding tasks, it is obvious that audio could be very helpful. Consider the action "playing saxophone". One would expect that the unique sound signature would significantly facilitate recognizing the class. Furthermore, visually subtle classes such as "whistling", where the action itself can be difficult to see in video frames, can be much easier to recognize with the aid of audio signals.

This line of thinking is supported by perceptual and neuroscience studies suggesting interesting ways in which visual and audio signals are combined in the brain. A classic example is the McGurk effect [46][1] – when one is listening to an audio clip (e.g., sounding "ba-ba"), alongside watching a video of fabricated lip movements (indicating "va-va"), the sound one perceives *changes* (in this case from "ba-ba" to "va-va").

*Work done during an internship at Facebook AI Research.

[1]https://www.youtube.com/watch?v=G-lN8vWm3m0

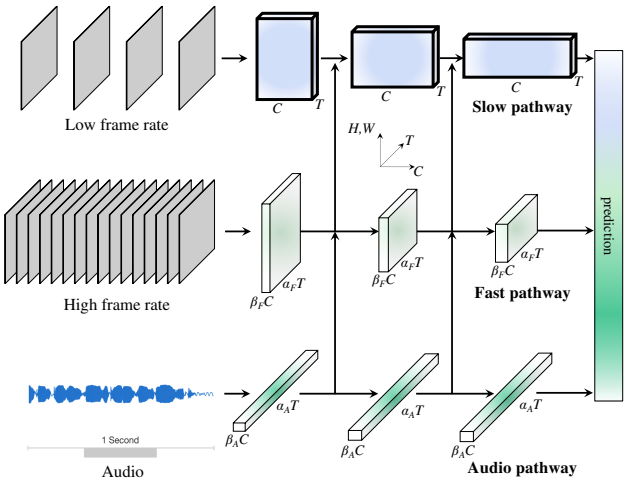

Figure 1. **Audiovisual SlowFast Networks** extend SlowFast (top two paths) with an Audio pathway (bottom). The network performs *integrated audiovisual perception* with hierarchical fusion.

This effect demonstrates that there is tight entanglement between audio and visual signals (known as the *multisensory integration process*) [45, 37, 65, 64]. Importantly, research has suggested this fusion between audio and visual signals happens at a fairly early stage [57, 49].

Given its high potential in facilitating video understanding, researchers have attempted to utilize audio in videos [36, 19, 2, 5, 50, 51, 3, 58, 18]. However, there are a few challenges in making effective use of audio. First, audio does not always correspond to visual frames (e.g., in a "dunking basketball" video, there can be class-unrelated background music playing). Conversely, audio does not always contain information that can help understand the video (e.g., "shaking hands" does not have a particular sound signature). There are also challenges from a technical perspective. Specifically, we identify the incompatibility of "learning dynamics" between the visual and audio pathways – audio pathways generally train much faster than visual ones, which can lead to generalization issues during joint audiovisual training. Due in part to these various difficulties, a principled approach for audiovisual modeling is currently lacking. Many previous methods adopt an ad-hoc scheme that consists of a separate audio network that is integrated with the visual pathway via "late-fusion" [19, 2, 50].

The objective of this paper is to build an architecture for *integrated audiovisual perception*. We aim to go beyond previous work that performs "late-fusion" of independent audio and visual pathways, to instead learn *hierarchies* of integrated audiovisual features, enabling unified audiovisual perception. We propose a new architecture, *Audiovisual SlowFast* Networks (AVSlowFast), to perform fusion at multiple levels (Fig. 1). AVSlowFast Networks build on previous work of SlowFast Networks [13], a class of architectures that has two pathways, of which one (Slow) is designed to capture more static but semantic-rich information whereas the other (Fast) is tasked to capture fast motion. Our architecture extends the Slow and Fast pathways of [13] with a Faster audio pathway, as audio has a much higher sampling rate, that is hierarchically integrated in AVSlowFast. Our key contributions are:

(i) We propose to fuse audio and visual information *at multiple levels* in the network hierarchy (*i.e.*, hierarchical fusion) so that audio can contribute to the formation of visual concepts at different levels of abstraction. In contrast to late-fusion, this enables the audio signal to participate in the process of forming visual features.

(ii) Unlike many previous approaches which directly take existing image-desgned CNNs (*e.g.*, VGG [62], ResNet [27]) to process audio [19, 28, 8, 78], we employ a ResNet based audio pathway design that is more efficient.

(iii) To overcome the incompatibility of learning dynamics between the visual and audio pathways, we propose *DropPathway* that randomly drops the Audio pathway during training as a simple and effective regularization technique to tune the pace of the learning process. This enables us to train our joint audiovisual model with hierarchical fusion connections across modalities.

(iv) Inspired by prior work in neuroscience [37], which suggests that there exist *audiovisual mirror neurons* in monkey brains that respond to "any evidence of the action, be it auditory or visual", we propose to perform audio visual synchronization (AVS) [50, 39, 2, 5] at multiple network layers to learn features that generalize across modalities.

We conduct extensive experiments on multiple action recognition dataset, Kinetics [35], EPIC-kitchen [11] and Charades [60] for action classification, as well as AVA [23] for action detection. We report state-of-the-art results on all datasets and demonstrate the benefits of joint audiovisual recognition. In addition, we show the generalization of AVSlowFast to self-supervised tasks, where it improves over prior work without bells and whistles. Finally, we provide detailed ablation studies to dissect the contribution of various components in AVSlowFast networks.

2. Related Work

Video recognition. Significant progress has been made in video recognition in recent years, some notable directions are two-stream networks in which one stream processes RGB frames and the other processes optical flow [61, 14, 77], 3D ConvNets as an extension of 2D networks to the spatiotemporal domain [72, 54, 84], and recent SlowFast Networks that have two pathways to process videos at different temporal frequency [13]. Despite all these efforts on harnessing temporal information in videos, research is relatively lacking when it comes to another important information source – audio in video.

Audiovisual activity recognition. Joint modeling of audio and visual signals has been largely conducted in a "late-fusion" manner in video recognition literature [36, 19]. For example, all the entries that utilize audio in the 2018 ActivityNet challenge report [19] have adopted this paradigm – meaning that there are networks processing visual and audio inputs separately, and then they either concatenate the output features or average the final class scores across modalities. Recently, an interesting audiovisual fusion approach has been proposed [36] using flexible binding windows when fusing audio and visual features. With three identical network streams, this approach fuses audio features with the features from a single RGB frame and optical flow after global average pooling, at the end of the network streams. In contrast, we fuse hierarchically in an *integrated architecture*, which we show to be beneficial. In addition, unlike methods that directly take existing visual CNNs (e.g., Inception [71], ResNet [27], Inception-ResNet [70], *etc.*) to process audio, we propose a dedicated audio pathway which we will show to be more effective in experiments.

Other audiovisual tasks. Audio has also been extensively utilized outside of video recognition, *e.g.* for learning audiovisual representations in a self-supervised manner [2, 5, 50, 51, 39] by exploiting audio-visual correspondence. While related, the goal is different to ours (learning representations *vs.* video recognition). Further, as methods discussed above, these approaches typically apply late-fusion on audio and visual features. Other audiovisual tasks that have been studied include audio-visual speech recognition [53, 48], sound-source localization [3, 58], audio-visual source separation [50, 18], and audiovisual question answering [1].

Multi-modal learning. Researchers have long been interested in developing models that can learn from multiple modalities (e.g., audio, vision, language, *etc.*). Beyond audio and visual modalities, extensive research has been conducted in various other instantiations of multi-modal learning, including vision and motion [61, 15, 77, 9], vision and language [4, 17], and learning from physiological data [42]. Recently, Wang et al. discussed the difficulty of audiovisual joint training in a learning context [78]. Unlike [78], which requires carefully balancing audio and visual loss terms, we propose DropPathway and a hierarchical AV synchronization strategy to jointly train AVSlowFast from scratch.

3. Audiovisual SlowFast Networks

Inspired by prior research [7], which suggests that audio and visual signals fuse at multiple cognitive levels, we propose to fuse audio and visual features at multiple stages, from intermediate-level features to high-level semantic concepts. This way, audio can participate in the formation of visual concepts at different levels. Audiovisual SlowFast Networks are conceptually simple. SlowFast has Slow and Fast pathways to process visual inputs (Sec. 3.1); to this we add a third Audio pathway that processes audio (Sec. 3.2).

3.1. SlowFast pathways

We begin by briefly reviewing the SlowFast architecture. The Slow pathway (Fig. 1, top row) is a convolutional network that processes videos with a large temporal stride (*i.e.*, it samples one frame out of τ frames). The primary goal of Slow pathway is to produce features that capture semantic contents of the video, which has a low refreshing rate (semantics do not change all of a sudden). The Fast pathway (Fig. 1, middle row) is another convolutional model with three key properties. First, it has an α_F times higher frame rate (*i.e.*, with temporal stride τ/α_F, $\alpha_F > 1$) so that it can capture fast motion information. Second, it preserves fine temporal resolution by avoiding any temporal downsampling. Third, it has a lower channel capacity (β_F times the Slow pathway channels, where $\beta_F < 1$) as it is demonstrated to be a desired trade-off [13]. We refer readers to [13] for more details.

3.2. Audio pathway

A key property of the Audio pathway is that it has an even finer temporal structure than the Slow and Fast pathways (with waveform sampling rate on the order of kHz). As standard processing, we take log-mel-spectrogram (2-D representation in time and frequency of audio) as input and set the temporal stride to τ/α_A frames, where α_A can be much larger than α_F (*e.g.*, 32 *vs.* 8). In a sense, it serves as a "Faster" pathway with respect to Slow and Fast pathways. Another notable property of the Audio pathway is its low computation cost. Due to the 1-D nature of audio signals, they are cheap to process. To control this, we set the channels of Audio pathway to $\beta_A \times$ Slow pathway channels. By default, we set β_A to $1/2$. Depending on the specific instantiation, the Audio pathway typically only requires 10% to 20% of the overall computation of AVSlowFast Networks.

3.3. Lateral connections

In addition to the lateral connections between Slow and Fast pathways in [13], we add lateral connections between the Audio, Slow & Fast pathways to fuse audio and visual features. Following [13], lateral connections are added after ResNet "stages" (*e.g.*, pool_1, res_2, res_3, res_4 and pool_5).

stage	*Slow* pathway	*Fast* pathway	*Audio* pathway
raw clip	$3\times64\times224^2$	$3\times64\times224^2$	80×128 (*freq.*$\times$*time*)
data layer	stride 16, 1^2	stride 2, 1^2	-
conv_1	1×7^2, 64 stride 1, 2^2	5×7^2, 8 stride 1, 2^2	[9×1, 1×9], 32 stride 1, 1
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	-
res_2	$\begin{bmatrix}1\times1^2, 64\\1\times3^2, 64\\1\times1^2, 256\end{bmatrix}\times3$	$\begin{bmatrix}3\times1^2, 8\\1\times3^2, 8\\1\times1^2, 32\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 32\\[3×1, 1×3], 32\\1\times1, 128\end{bmatrix}\times3$
res_3	$\begin{bmatrix}1\times1^2, 128\\1\times3^2, 128\\1\times1^2, 512\end{bmatrix}\times4$	$\begin{bmatrix}3\times1^2, 16\\1\times3^2, 16\\1\times1^2, 64\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 64\\[3×1, 1×3], 64\\1\times1, 256\end{bmatrix}\times4$
res_4	$\begin{bmatrix}3\times1^2, 256\\1\times3^2, 256\\1\times1^2, 1024\end{bmatrix}\times6$	$\begin{bmatrix}3\times1^2, 32\\1\times3^2, 32\\1\times1^2, 128\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times6$
res_5	$\begin{bmatrix}3\times1^2, 512\\1\times3^2, 512\\1\times1^2, 2048\end{bmatrix}\times3$	$\begin{bmatrix}3\times1^2, 64\\1\times3^2, 64\\1\times1^2, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times3$
global average pool, concate, fc			

Table 1. **An example instantiation of the AVSlowFast network**. For Slow & Fast pathways, the dimensions of kernels are denoted by $\{T\times S^2, C\}$ for temporal, spatial, and channel sizes. For the Audio pathway, kernels are denoted with $\{F\times T, C\}$, where F and T are frequency and time. Strides are denoted with {temporal stride, spatial stride^2} and {frequency stride, time stride} for SlowFast and Audio pathways, respectively. In this example, the speed ratios are $\alpha_F = 8$, $\alpha_A = 32$ and the channel ratios are $\beta_F = 1/8$, $\beta_A = 1/2$ and $\tau = 16$. The backbone is ResNet-50.

However, unlike [13], which has lateral connections after each stage, we found that it is most beneficial to have lateral connections between audio and visual features starting from intermediate levels (we ablate this in Sec. 4.2). Next, we discuss several concrete instantiations of AVSlowFast.

3.4. Instantiations

AVSlowFast Networks define a generic class of models that follow same design principles. In this section, we exemplify a specific instantiation in Table 1. We denote spatiotemporal size by $T\times S^2$ for Slow/Fast pathways and $F\times T$ for Audio pathway, where T is the temporal length, S is the height and width of a square spatial crop and F is frequency bins for audio.

SlowFast pathways. For Slow and Fast pathways, we follow the basis instantiation of SlowFast 4×16, R50 model defined in [13]. It has a Slow pathway that samples $T = 4$ frames out from a 64-frame raw clip with a temporal stride $\tau = 16$. There is no temporal downsampling in the Slow pathway, since input stride is large. Also, it only applies non-degenerate temporal convolutions (temporal stride > 1) in res_4 and res_5 (see Table 1), as this is more effective.

For the Fast pathway, it has a higher frame rate ($\alpha_F = 8$) and a lower channel capacity ($\beta_F = 1/8$), such that it can better capture motion while trading off spatial modeling capability. The Fast pathway adopts non-degenerate temporal convolutions in every block to preserve fine temporal mod-

eling capability. Spatial downsampling is performed with stride 2^2 convolution in the center ("bottleneck") filter of the first residual block in each stage.

Audio pathway. The Audio pathway takes as input the log-mel-spectrogram representation, which is a 2-D representation with one axis being time and the other one denoting frequency bins. In the instantiation shown in Table 1, we use 128 spectrogram frames (corresponding to 2 seconds of audio) with 80 frequency bins.

Similar to Slow/Fast pathways, the Audio pathway is also based on a ResNet, but with specific design to better fit the audio inputs. First, it does not perform pooling after the initial convolutioal filter (*i.e*. there is no downsampling layer at stage pool_1) to preserve information along both temporal and frequency axis. Downsampling in time-frequency space is performed by stride 2^2 convolution in the center ("bottleneck") filter of the first residual block in each stage from res_2 to res_5.

Second, we decompose the 3×3 convolution filters in res_2 and res_3 into 1×3 filters for frequency and 3×1 filters for time. This not only reduces computation, but it also allows the network to treat time and frequency differently (as opposed to 3×3 filters which implies both axis are equivalent) in early stages. While for spatial filters it is reasonable to perform filtering in x and y dimensions symmetrically, this might not be optimal for early filtering in time and frequency dimensions, as the statistics of spectrograms are different from natural images which are approximately isotropic and shift-invariant [56, 30].

Lateral connections. There are many options on how to fuse audio features into the visual pathways. Here we describe several instantiations and the motivation behind them. Note that here we are discussing the lateral connections between Audio pathway and SlowFast pathways. For the fusion connection between the two visual pathways (Slow and Fast), we adopt the temporal strided convolution as it is demonstrated to be most effective in [13].

(i) *A→F→S*: In this approach (Fig. 2 left), Audio pathway (A) is first fused to the Fast pathway (F), and then fused to the Slow pathway (S). Specifically, audio features are subsampled to the temporal length of the Fast pathway and then fused into the Fast pathway with a *sum* operation. After that, the resulting features are further subsampled (e.g., $4\times$ subsample) and fused with the Slow pathway (as is done in SlowFast). The key property of this fusion method is that it enforces *strong temporal alignment* between audio and visual features, as audio features are fused into Fast pathway which preserves fine temporal resolution.

(ii) *A→FS*: An alternative way is to fuse the Audio pathway into the output of the SlowFast fusion (Fig. 2 center), which is coarser in temporal resolution. This method imposes a less stringent requirement on temporal alignment between audio and visual features. Note that similar ideas of relaxing the alignment requirement is also explored in [36], but in the context of late fusion of RGB, flow and audio streams.

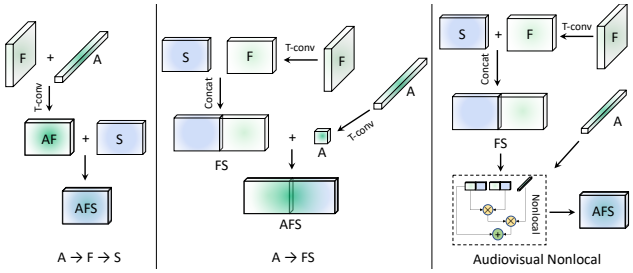

Figure 2. **Fusion connections for AVSlowFast**. *Left*: A→F→S enforces strong temporal alignment between audio and RGB frames, as audio is fused into the Fast pathway with fine temporal resolution. *Center*: A→FS has higher tolerance on temporal misalignment as audio is fused into the temporally downsampled output of SlowFast fusion. *Right*: Audiovisual Nonlocal fuses through a Nonlocal block [79], such that audio features are used to select visual features that are deemed important by audio.

(iii) *Audiovisual Nonlocal*: One might also be interested in using audio as a *modulating signal* to visual features. Specifically, instead of directly summing or concatenating audio features into the visual stream, one might expect audio to play a more subtle role of modulating, through attention mechanisms such as Non-Local (NL) blocks [79], the visual concepts. One example would be audio serving as a probing signal indicating where the interesting event is happening in the video, both spatially and temporally, and then focus the attention of visual pathways on those locations. To materialize this, we adapt NL blocks to take both audio and visual features as inputs (Fig. 2 right). Audio features are then matched to different locations within visual features (along H, W and T axis), and the affinity is used to generate a new visual feature that combines information from locations deemed important by audio features.

3.5. Joint audiovisual training

Unlike in SlowFast, AVSlowFast trains with multiple modalities. As noted in Sec. 1, this leads to challenging training dynamics (*i.e*., different training speed of audio and visual pathways). To tackle this, we propose two training strategies that enable joint training of AVSlowFast.

DropPathway. We discuss a possible reason for why many previous video classification approaches employ audio in an ad-hoc manner (*i.e*., late fusion). By analyzing the model training dynamics we observe the following. First, audio and visual pathways are very different in terms of their "learning speed". Taking the curves in Fig. 3 as an example, the green curve is for training a visual-only SlowFast model, whereas the red curve is for training an Audio-only model. It shows that the Audio-only model requires fewer training iterations before it starts to overfit (at ~70 epochs, which is $\sim 1/3$ of visual model training epochs).

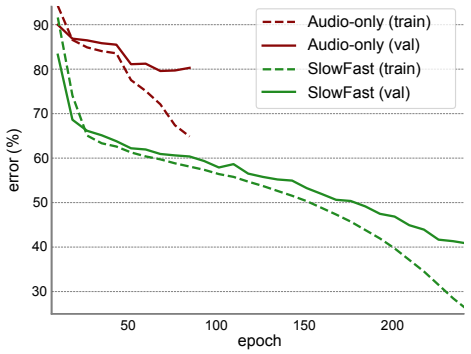

Figure 3. Training procedure on Kinetics for Audio-only (red) *vs*. SlowFast (green) networks. We show the top-1 training error (dash) and validation error (solid). The curves show single-crop *errors*; the video *accuracy* is 24.8% *vs*. 75.6%. The audio network converges with much less iterations compared to the visual one.

As we will show by experiments, this discrepancy on learning pace leads to strong overfitting if we naively train both modalities jointly. To unlock the potential of joint training, we adopt a simple strategy of *randomly dropping the Audio pathway during training* (referred to as *DropPathway*). Specifically, at each training iteration, we would drop the Audio pathway altogether with probability P_d. This way, we *slow down* the learning of the Audio pathway and make its learning dynamics more compatible with its visual counterpart. When dropping the audio pathway, we simply feed zero tensors into visual pathways (we also explored feeding running average of audio features, and found similar results). Our ablation studies in the next section will show the effect of DropPathway, demonstrating that this simple strategy provides good generalization and is essential for jointly training AVSlowFast models.

Hierarchical audiovisual synchronization. As noted in Sec. 2, temporal synchronization (that comes for free) between audio and visual sources has been explored as self-supervisory signal to learn feature representations [51, 5, 2, 10, 50, 39, 24]. In this work, we use audiovisual synchronization to encourage the network to produce feature representations that are generalizable across modalities (inspired by the *audiovisual mirror neurons* in primate vision [37]). Specifically, we add an auxiliary task to classify whether a pair of audio and visual frames are *in-sync* or not [39, 50] and adopt a curriculum schedule used in [39] that starts with easy negatives (audio and visual frames come from different clips), and transition into a mix of easy and hard (audio and visual frames are from the same clip, but with a temporal shift) after 50% of training epochs.

One notable difference of our approach to previous work is that unlike previous work which has a single synchronization loss at the network output (since these works adopt a "late-fusion" strategy), we add multiple losses to each of the fusion junctions, since AVSlowFast has fusion connections at multiple levels. As we will show in our experiments, this leads to better audiovisual features learned by AVSlowFast.

4. Experiments: Action Classification

We evaluate our approach on four video recognition datasets using standard evaluation protocols. For the action classification experiments in this section we use the Kinetics-400 [35], EPIC-Kitchens [11] and Charades [60]. For action detection experiments, we use the challenging AVA dataset [23], which will be covered in Sec. 5.

Datasets. Kinetics-400 [35] is a large-scale video dataset of ~240k training videos and 20k validation videos in 400 action categories. Results on Kinetics are reported as top-1 and top-5 classification accuracy (%).

The EPIC-Kitchens dataset [11] consists of egocentric videos of daily activities recorded in various kitchen environments. It has 39k segments in 432 videos. For each segment, the task is to predict a verb (*e.g.*, "turn-on"), a noun (*e.g.*, "switch"), and an action by combining the two ("turn on switch"). Performance is measured as top-1 and top-5 accuracy. We use the train/val split following [6]. Test set results are obtained by submitting to the evaluation server.

Charades [60] is a dataset of ~9.8k training videos and 1.8k validation videos in 157 classes. Each video has multiple labels of activities spanning ~30 seconds. Performance is measured in mean Average Precision (mAP).

Audio pathway. Following previous work [2, 3, 39, 40], we extract log-mel-spectrograms from the raw audio waveform to serve as the input to Audio pathway. Specifically, we sample audio data with 16 kHz sampling rate, then compute a spectrogram with window size of 32ms and step size of 16ms. The length of audio input is exactly matched to the duration spanned by RGB frames. For example, under 30 FPS, for AVSlowFast with $T\times\tau = 8\times8$ frames (2 secs) input, we sample 128 frames (2 secs) in log-mel-spectrogram.

Training. We train our AVSlowFast models on Kinetics from scratch without any pre-training. We use synchronous SGD optimizer and follow the training recipe (e.g., learning rate, weight decay, warm-up, etc) used in [13]. Given a training video, we randomly sample T frames with stride τ and extract the corresponding log-mel-spectrogram. For video frames, we randomly crop 224×224 pixels from a video, applying horizontal flip at random, and resize it to a shorter side sampled in [256, 320] pixels [62, 79].

Inference. Following previous work [79, 13], we uniformly sample 10 clips from a video along its temporal axis. For each clip, we resize the shorter spatial side to 256 pixels and take 3 crops of 256×256 along the longer side to cover the spatial dimensions. Video-level predictions are computed by averaging softmax scores. We report the actual *inference-time* computation as in [13], by listing the FLOPs per spacetime "view" of spatial size 256^2 (temporal clip with spatial crop) at inference *and* the number of views (*i.e.* 30 for 10 temporal clips each with 3 spatial crops).

Full training and inference details for Kinetics, EPIC and Charades are in appendices A.4, A.5, and A.6, respectively.

model	inputs	pretrain	top-1	top-5	KS	GFLOPs×views
I3D [9]	V	✓	72.1	90.3	-	108 × N/A
Nonlocal [79], R101	V	✓	77.7	93.3	-	359 × 30
R(2+1)D [74]	V	-	72.0	90.0	-	152 × 115
R(2+1)D [74]	V+F	-	73.9	90.9	-	304 × 115
I3D [9]	V+F	-	71.6	90.0	-	216 × N/A
ECO [86]	V	-	70.0	89.4	-	N/A × N/A
S3D [83]	V	-	69.4	89.1	-	66.4 × N/A
ARTNet [76]	V	-	69.2	88.3	-	23.5 × 250
STC [12]	V	-	68.7	88.5	-	N/A × N/A
ip-CSN-152 [73]	V	-	77.8	92.8	-	109 × 30
3-stream late fusion [8]	A+V+F	✓	74.9	91.6	-	N/A × N/A
3-stream LSTM [8]	A+V+F	✓	77.1	93.2	-	N/A × N/A
3-stream SATT [8]	A+V+F	✓	77.7	93.2	-	N/A × N/A
GBlend [78]	V	-	76.4	92.1	-	N/A × N/A
GBlend [78]	A+V	-	77.7	93.0	-	N/A × N/A
SlowFast, R50 [13]	V	-	75.6	92.0	80.5	36 × 30
AVSlowFast, R50	A+V	-	77.0	92.7	83.7	40 × 30
SlowFast, R101 [13]	V	-	77.9	93.2	82.7	106 × 30
AVSlowFast, R101	A+V	-	**78.8**	**93.6**	**85.0**	129 × 30

Table 2. **AVSlowFast results on Kinetics.** AVSlowFast and SlowFast instantiations are with $T\times\tau = 4\times16$ and $T\times\tau = 8\times8$ inputs for R50/R101, and without NL blocks. "N/A" indicates the numbers are not available for us. "KS" refers to top-1 accuracy on Kinetics-Sounds dataset [2], which is a subset of 34 Kinetics classes. "pretrain" refers to ImageNet pretraining.

4.1. Main Results

Kinetics. We present action recognition results of AVSlowFast on Kinetics in Table 2. First, we compare AVSlowFast with SlowFast and see a margin of 1.4% top-1 for R50 and 0.9% top-1 for R101, given the same network backbone and input size. This demonstrates the effectiveness of the audio stream despite its modest cost of only ≈10%−20% of the overall computation. Comparatively going deeper from R50 to R101 increases computation by 194% for a slightly higher gain in accuracy.

AVSlowFast compares favorably to existing methods that utilize various modalities, *i.e.*, audio (A), visual frames (V) and optical flow (F). Adding optical flow streams can bring similar gains for doubling computation (R(2+1)D in Table 2). When comparing to other methods that also utilize audio [78, 8], despite building upon a stronger baseline (our SlowFast baseline is as good as GBlend's [78] final AV model: 77.9% *vs*. 77.7%), AVSlowFast is still able to bring complementary benefits from audio. We refer readers to appendix A.3 for more comparisons to GBlend.

Furthermore, as Kinetics is a visual-heavy dataset (for many classes *e.g.* "*writing*" audio is not useful), to better study audiovisual learning, [2] proposes "Kinetics-Sounds" as a subset of 34 Kinetics classes that are potentially manifested both visually and aurally (example classes include "*blowing nose*" and "*playing drums*"). We test both SlowFast and AVSlowFast on Kinetics-Sounds in "KS" column of Table 2. As expected, the gain from SlowFast to AVSlowFast is stronger on Kinetics-Sounds – for R50/R101, gains doubled to +3.2%/+2.3%, showing the potential of audio on relevant data.

	verbs		nouns		actions	
	top-1	top-5	top-1	top-5	top-1	top-5
validation						
3D CNN [81]	49.8	80.6	26.1	51.3	19.0	37.8
LFB [81]	52.6	81.2	**31.8**	56.8	22.8	41.1
SlowFast [13]	55.8	83.1	27.4	52.1	21.9	39.7
AVSlowFast	**58.7**	**83.6**	31.7	**58.4**	**24.2**	**43.6**
test s1 (seen)						
LFB [81]	60.0	88.4	45.0	**71.8**	32.7	55.3
FBK-HUPBA [66]	63.3	89.0	44.8	69.9	35.5	57.2
HF-TSN [67]	57.6	87.8	39.9	65.4	28.1	48.6
EPIC-Fusion [36]	64.8	**90.7**	46.0	71.3	34.8	56.7
AVSlowFast	**65.7**	89.5	**46.4**	71.7	**35.9**	**57.8**
test s2 (unseen)						
LFB [81]	50.9	77.6	31.5	57.8	21.2	39.4
FBK-HUPBA [66]	49.4	77.5	27.1	52.0	20.3	37.6
HF-TSN [67]	42.4	75.8	25.2	49.0	16.9	33.3
EPIC-Fusion [36]	52.7	79.9	27.9	53.8	19.1	36.5
AVSlowFast	**55.8**	**81.7**	**32.7**	**58.9**	**24.0**	**43.2**

Table 3. **EPIC-Kitchens validation and test set results**. Backbone: 8×8, R101 (w/o NL).

model	pretrain	mAP	GFLOPs×views
CoViAR, R-50 [82]	ImageNet	21.9	N/A
Asyn-TF, VGG16 [59]	ImageNet	22.4	N/A
MultiScale TRN [85]	ImageNet	25.2	N/A
Nonlocal, R101 [79]	ImageNet+Kinetics	37.5	544 × 30
STRG, R101+NL [80]	ImageNet+Kinetics	39.7	630 × 30
Timeception [31]	Kinetics-400	41.1	N/A×N/A
LFB, +NL [81]	Kinetics-400	42.5	529 × 30
SlowFast	Kinetics	42.5	234 × 30
SlowFast+Audio	Kinetics	42.8	-
AVSlowFast	Kinetics	**43.7**	278 × 30

Table 4. **Comparison with the state-of-the-art on Charades**. SlowFast and AVSlowFast are with R101+NL backbone and 16×8 sampling. "SlowFast+Audio" refers to applying late-fusion.

An comparison of our Audio pathway for Audio-only classification on Kinetics is provided in appendix A.2, and a class-level analysis for Kinetics in appendix A.3.

EPIC-Kitchens. Next, we compare to state-of-the-art methods on EPIC-Kitchens in Table 3. First, AVSlowFast advances SlowFast with *strong margins* of +2.9%/+4.3%/+2.3% for verb/noun/action, which clearly demonstrates the benefits of audio in egocentric video recognition. Second as system-level comparison, AVSlowFast exhibits higher performance in all three categories (verb/noun/action) and two test sets (seen/unseen), to the related, previous best EPIC-Fusion [36]. We observe larger performance gains on the unseen split (*i.e.*, novel kitchen scenes) of the test set (+3.1%/+4.8%/+4.9% for verb/noun/action), which demonstrates good generalization of our method. Comparing to LFB [81], that uses an object detector to localize objects, AVSlowFast achieve similar performance as for nouns (objects) on both the seen and unseen test sets, whereas SlowFast *without audio* is largely lacking behind LFB (-4.4% compared to LFB on val noun), which is intuitive as sound can be beneficial for recognizing objects. Overall, we echo the findings in [36] that *audio is a very useful signal for egocentric video* recognition and our AVSlowFast Networks makes good use of it.

connection	top-1	top-5	GFLOPs
A→F→S	75.3	91.8	51.4
A→FS	77.0	92.7	39.8
AV Nonlocal	77.2	92.9	39.9

(a) **Audiovisual fusion connection**.

β_A	top-1	top-5	GFLOPs
1/8	76.0	92.5	36.0
1/4	76.6	92.7	36.8
1/2	77.0	92.7	39.8
1	75.9	92.4	51.9

(b) **Audio channels**.

P_d	top-1	top-5
N/A	75.2	91.8
0.2	76.0	92.5
0.5	76.7	92.7
0.8	77.0	92.7

(c) **DropPathway rate** P_d.

AVS	top-1	top-5
N/A	76.4	92.5
res_5	76.7	92.8
$\text{res}_{4,5}$	76.9	92.9
$\text{res}_{3,4,5}$	77.0	92.7

(d) **Hierarchical AV sync**.

Table 5. **Ablations on AVSlowFast design** on Kinetics-400. We show top-1 and top-5 classification accuracy (%), as well as computational complexity measured in GFLOPs for a single clip input of spatial size 256^2. Backbone: 4×16, R-50.

Charades. We test the effectiveness of AVSlowFast on videos of longer range activities on Charades in Table 4. We observe that audio can facilitate recognition (+1.2% over a strong SlowFast baseline) and we achieve state-of-the-art performance under Kinetics-400 pre-training. We further report the performance for late fusion: It only improves marginally over SlowFast (+0.3%) while AVSlowFast better exploits audio as an *integrated audiovisual* architecture.

4.2. Ablation Studies

We show ablation results on Kinetics-400 to study the effectiveness and tradeoffs of various design choices.

fusion stage	top-1	top-5	GFLOPs
SlowFast	75.6	92.0	36.1
SlowFast+Audio	76.1	92.0	-
pool_5	75.4	92.0	38.4
res_4 + pool_5	76.5	92.6	39.1
$\text{res}_{3,4}$ + pool_5	**77.0**	**92.7**	39.8
$\text{res}_{2,3,4}$ + pool_5	75.8	92.4	40.2

Table 6. **Effects of hierarchical fusion.** All models are based on R50 and input size 4×16. "pool_5" refers to fusing audio and visual features at the output of the last ResNet stage (i.e., late-fusion).

Hierarchical fusion. We first study the effectiveness of fusion in Table 6. The first interesting phenomenon is that direct ensembling of audio/visual models produces modest gain (76.1% vs 75.6%), whereas joint training with late-fusion ("pool_5") hurts (75.6% → 75.4%).

Next for our hierarchical, multi-level fusion, it is beneficial to fuse audio and visual features at multiple levels. Specifically, we found that recognition accuracy steadily increases from 75.4% to 77.0% when we increase the number of fusion connections from one (*i.e.*, only concatenating pool5 outputs) to three ($\text{res}_{3,4}$ + pool_5) where it peaks. If we further add a lateral connection at res_2, the performance starts to drop. This suggests that it is beneficial to start fusing audio and visual features from intermediate levels (res_3) all the way to the top of the network. We hypothesize that this is because audio facilitates the formation of visual concept, but only when features mature to intermediate concepts that are generalizable across modalities (*e.g.* local edges typically do not have a general sound pattern).

Lateral connections. We ablate the the effectiveness of different instantiations of lateral connections between audio and visual pathways in Table 5a. First, A→F→S, which enforces strong temporal alignment between audio and visual streams, produces much worse classification accuracy comparing to A→FS, which relaxes the requirement on alignment. This coincides with [36] arguing it is beneficial to have tolerance on alignment between the modalities, since class-level audio signals might happen out-of-sync to visual frames (*e.g.*, when shooting 3 pointers in basketball, net-touching sound only comes after the action finishes). Finally, the straightforward A→FS connection performs similarly to the more complex Audiovisual Nonlocal [79] fusion (77.0% vs 77.2%). We use A→FS as our default lateral connection for its good performance and simplicity.

Audio pathway capacity. We study the impact of the number of channels of the Audio pathway (β_A) in Table 5b. As expected, when we increase the number of channels (*e.g.*, increasing β_A from 1/8 to 1/2, which is the ratio between Audio and Slow pathway's channels), accuracy improves at the cost of increased computation. However, performance starts to degrade when we further increase it to 1, likely due to overfitting. We use $\beta_A = 1/2$ across all our experiments.

DropPathway. As we discussed before, we apply pathway dropping to adjust the incompatibility of the learning speed across modalities. Here we conduct ablative experiments to study the effects of different drop rates P_d. The results are shown in Table 5c. As shown in the table, a high value of P_d (0.5 or 0.8) is required to slow down the Audio pathway when training audio and visual pathways jointly. In contrast, when we train AVSlowFast without DropPathway (N/A), the accuracy degrades dramatically to be even worse than visual-only models (75.2% vs 75.6%). This is because the Audio pathways learns too fast and start to overfit and dominate the visual feature learning. This demonstrates the importance of DropPathway for joint audiovisual training.

Hierarchical audiovisual synchronization. We study the effectiveness of hierarchical audiovisual synchronization in Table. 5d. We use AVSlowFast with and without AVS, and vary the layers for multiple losses. We observe that adding AVS as an auxiliary task is beneficial (+0.6% gain). Furthermore, having synchronization loss at multiple levels slightly increases the performance (without cost). This shows that it is beneficial to have a feature representation that is generalizable across audio and visual modalities and hierarchical AVS could facilitate to produce such representation.

5. Experiments: AVA Action Detection

In addition to the image-level action recognition task, we also apply AVSlowFast models on action detection which requires both localizing and recognizing actions. Although audio does not provide spatial localization information, we hope it can help recognition and therefore benefit detection.

Dataset. The AVA dataset [23] focuses on spatiotemporal localization of human actions. Spatiotemporal labels are provided for one frame per second, with people annotated with a bounding box and (possibly multiple) actions. There are 211k training and 57k validation video segments. We follow the standard protocol [23] of evaluating on 60 classes. The metric is mean Average Precision (mAP) over 60 classes, using a frame-level IoU threshold of 0.5.

Detection architecture. We follow the detection architecture introduced in [13], which is adapted from Faster R-CNN [55] for video. During training, the input to our audiovisual detector is $\alpha_F T$ RGB frames sampled with temporal stride τ and spatial size 224×224, to SlowFast pathways, and the corresponding log-mel-spectrogram covering this time window to Audio pathway. During testing, the backbone feature extractor is computed fully convolutional with RGB frame shorter side of 256 pixels [13], as is standard in Faster R-CNN [55]. For details on architecture, training and inference, please refer to appendix A.7.

Main Results. We compare AVSlowFast to SlowFast as well as several other existing methods in Table 7. AVSlowFast, with both R50 and R101 backbones, outperforms SlowFast with a consistent margin (~1.2%), but only increases FLOPs[2] slightly, *e.g.* for R50 *by only 2%*, whereas going from SlowFast R50 to R101 (without audio) increases computation by 180% more FLOPs. This demonstrates that information from audio can be cheap and beneficial also for action detection, where spatiotemporal localization is required. Interestingly, the ActivityNet Challenge 2018 [19] hosted a separate track for multiple modalities but no team could achieve gains using audio information on AVA. For system-level comparison to other approaches, Table 7 shows that AVSlowFast achieves state-of-the-art performance on AVA under Kinetics-400 (K400) pretraining.

For future comparisons, we show results on v2.2 of AVA, which provides more consistent annotations. We see consistent results as for v2.1. As for per-class results, we found classes like ["*swim*" +30.2%], ["*dance*" +10.0%], ["*shoot*" +8.6%], and ["*hit (an object)*" +7.6%] has the largest gain from audio; see appendix A.3 for a figure and more details.

[2]We report FLOPs for fully-convolutional inference of a clip with 256×320 spatial size for SlowFast and AVSlowFast models, full test-time computational cost for these models is directly proportional to this.

model	inputs	AVA	pretrain	val mAP	GFLOPs
I3D [23]	V+F			15.6	-
ACRN, S3D [69]	V+F			17.4	-
ATR, R50+NL [33]	V+F			21.7	-
9-model ensemble [33]	V+F			25.6	-
I3D+Transformer [21]	V			25.0	-
LFB, + NL R50 [81]	V	v2.1	K400	25.8	529
LFB, + NL R101 [81]	V			26.8	677
SlowFast 4×16, R50	V			24.3	65.7
AVSlowFast 4×16, R50	A+V			25.4	67.1
SlowFast 8×8, R101	V			26.3	184
AVSlowFast 8×8, R101	A+V			**27.8**	210
SlowFast 4×16, R50	V			24.7	65.7
AVSlowFast 4×16, R50	A+V	v2.2	K400	25.9	67.1
SlowFast 8×8, R101	V			27.4	184
AVSlowFast 8×8, R101	A+V			**28.6**	210

Table 7. **Comparison with the state-of-the-art on AVA**. Both AVSlowFast and SlowFast use 8×8 frame inputs. For R101, both AVSlowFast and SlowFast also use NL [79].

method	inputs	#param	FLOPs	pretrain	UCF	HMDB
Shuffle&Learn [47, 68]	V	58.3M	-	K600	26.5	12.6
3D-RotNet [34, 68]	V	33.6M	-	K600	47.7	24.8
CBT [68]	A*+V	-	-	K600	54.0	29.5
AVSlowFast	A+V	38.5M	63.4G	K400	**77.4**	**42.2**

Table 8. **Comparison under the linear classification protocol**. We only train the the last *fc* layer after *self-supervised* pretraining on Kinetics-400 (abbreviated as K400). Top-1 accuracy averaged over three splits is reported for comparison to previous work. AVSlowFast use R50 backbone with $T \times \tau = 8 \times 8$ sampling. A*+V refers to using transcripts obtained from ASR on audio.

6. Experiments: Self-supervised Learning

To further demonstrate the generalization of AVSlowFast models, we apply it to self-supervised audiovisual representation learning. To demonstrate that AVSlowFast is readily applicable to existing audio/visual tasks, we directly adopt the AudioVisual synchronization [2, 40, 50] and image rotation prediction [20] (0°, 90°, 180°, 270°; as a four-way softmax-classification) losses proposed in existing literature. With the learned representation, we then retrain the last *fc* layer of AVSlowFast on UCF101 [63] and HMDB51 [41] following standard practice. Table 8 shows that, without bells and whistles, our smallest AVSlowFast, R50 model compares favorably to state-of-the-art self-supervised classification accuracy on these two datasets. For more details, please refer to appendix A.1.

7. Conclusion

This work has presented AVSlowFast Networks, an architecture for integrated audiovisual perception. We demonstrate the effectiveness of AVSlowFast with state-of-the-art performance on multiple datasets for video action classification and detection. We hope that AVSlowFast Networks will foster further research in video understanding.

A. Appendix

A.1. Results: Self-supervised Learning

method	inputs	#param	FLOPs	pretrain	UCF	HMDB
Shuffle&Learn [47, 68]	V	58.3M	N/A	K600	26.5	12.6
3D-RotNet [34, 68]	V	33.6M	N/A	K600	47.7	24.8
CBT [68]	A*+V	N/A	N/A	K600	54.0	29.5
AVSlowFast 4×16	A+V	38.5M	36.2G	K400	76.8	41.0
AVSlowFast 8×8	A+V	38.5M	63.4G	K400	77.4	42.2
AVSlowFast 16×4	A+V	38.5M	117.9G	K400	**77.4**	**44.1**
ablation (split1)						
SlowFast 4×16 (ROT)	V	33.0M	34.2G	K400	71.9	**42.0**
AVSlowFast 4×16 (AVS)	A+V	38.5M	36.2G	K400	73.2	39.5
AVSlowFast 4×16	A+V	38.5M	36.2G	K400	**77.0**	40.2

Table A.1. **Comparison under the linear classification protocol**. We only train the the last *fc* layer after self-supervised pretraining on Kinetics-400 (abbreviated as K400). Top-1 accuracy averaged over three splits is reported when comparing to previous work (top), results on split1 is used for ablation (bottom). All SlowFast models use use R50 backbones with $T \times \tau$ sampling. A*+V refers to using transcripts obtained from ASR on audio.

In this section, we provide more results and detailed analysis on self-supervised learning using AVSlowFast. First, we pretrain AVSlowFast with self-supervised objectives of audiovisual synchronization [2, 40, 50] (AVS) and image rotation prediction [20] (ROT) on Kinetics-400. Then, following the standard linear classification protocol used for image recognition tasks [25], we use the pretrained network as a fixed, *frozen* feature extractor and train a linear classifier on top of the self-supervisedly learned features. In Table A.1 (top), we compare to previous work that follows the *same protocol*. We note this is the same experiment as Table 8 of the main paper, only that we compare now just with methods that use frozen features, instead of the main paper which compares methods that train *all layers* on target datasets. The results indicate that features learned by AVSlowFast are significantly better than baselines including the recently introduced CBT method [68] (+23.4% for UCF101 and +14.6% for HMDB51), which pretrains on the larger Kinetics-600 dataset.

In addition, we also ablate the contribution of individual tasks of AVS and ROT in Table A.1 (bottom). On UCF101, SlowFast/AVSlowFast trained under either ROT or AVS objective outperforms previous work by large margins, while the combination of them perform the best. Whereas on the smaller HMDB51, all three variants of our method perform similarly well and audio seems less important.

Another aspect is that, although many previous approaches on self-supervsied feature learning focus on reporting number of parameters, FLOPs is in fact another important factor to consider – as shown in Table A.1 (top), the performance keeps increasing when we take higher temporal resolution inputs by varying $T \times \tau$ (*i.e.* larger FLOPs), even though model parameters remain identical.

method	inputs	#param	pretrain	UCF101	HMDB51
Shuffle & Learn [47]	V	58.3M	UCF/HMDB	50.2	18.1
OPN [43]	V	8.6M	UCF/HMDB	59.8	23.8
O3N [16]	V	N/A	Kinetics-400	60.3	32.5
3D-RotNet [34]	V	33.6M	Kinetics-400	62.9	33.7
3D-ST-Puzzle [38]	V	33.6M	Kinetics-400	65.8	33.7
DPC [24]	V	32.6M	Kinetics-400	75.7	35.7
CBT [68]	A*+V	N/A	Kinetics-600	79.5	44.6
Multisensory [50]	A+V	N/A	Kinetics-400	82.1	-
AVTS [39]	A+V	N/A	Kinetics-400	85.8	**56.9**
Optical flow stream [61, 15]	V	90.7M	-	83.7	54.6
IDT [75, 52]	V	N/A	-	87.9	61.1
AVSlowFast	A+V	38.5M	Kinetics-400	**87.0**	54.6

Table A.2. **Training all layers.** We present results using the popular protocol of fine-tuning all layers after self-supervised pretraining. Top-1 accuracy averaged over three splits is reported. We use AVSlowFast 16×4, R50 for this experiment. A*+V refers to CBT uses transcripts obtained from ASR on audio input.

Although we think the linear classification protocol serves as a better method to evaluate self-supervised feature learning (as features are frozen and therefore less sensitive to hyper-parameter settings such as learning schedule on target datasets, especially when target datasets are relatively small), we also evaluate by fine-tuning all layers of AVSlowFast on the target datasets to compare to a larger corpus of previous self-supervised feature learning work. Table A.2 shows that AVSlowFast also achieves competitive performance comparing to prior work under this setting.

A.2. Results: Audio-only Classification

model	dataset	pretrain	top-1	top-5	GFLOPs
VGG* [28]	Kinetics-600	Kinetics-400	23.0	-	-
SE-ResNext [19]	Kinetics-600	ImageNet	21.3	38.7	-
Inception-ResNet [19]	Kinetics-600	ImageNet	23.2	-	-
Audio-only (ours)	Kinetics-600	-	**26.5**	44.7	14.2
VGG [8]	Kinetics-400	-	21.6	39.4	-
GBlend [78]	Kinetics-400	-	19.7	33.6	-
Audio-only (ours)	Kinetics-400	-	**24.8**	**43.3**	14.2

Table A.3. **Results of Audio-only models.** VGG* model results are taken from "iTXN" submission from Baidu Research to ActivityNet challenge [19].

To understand the effectiveness of our Audio pathway, we evaluate it in terms of Audio-only classification accuracy on Kinetics (in addition to Kinetics-400, we also train and evaluate on Kinetics-600 to be comparable to methods that use this data in challenges [19]). In Table A.3, we compare our Audio-only network to several other audio models. We observe that our Audio-only model performs better than existing methods by solid margins (+3.3% top-1 accuracy on Kinetics-600 and +3.2% on Kinetics-400, compared to best-performing methods), which demonstrates the effectiveness of our Audio pathway design. Note also that unlike some other methods in Table A.3, we train our audio network from scratch on Kinetics, without any pretraining.

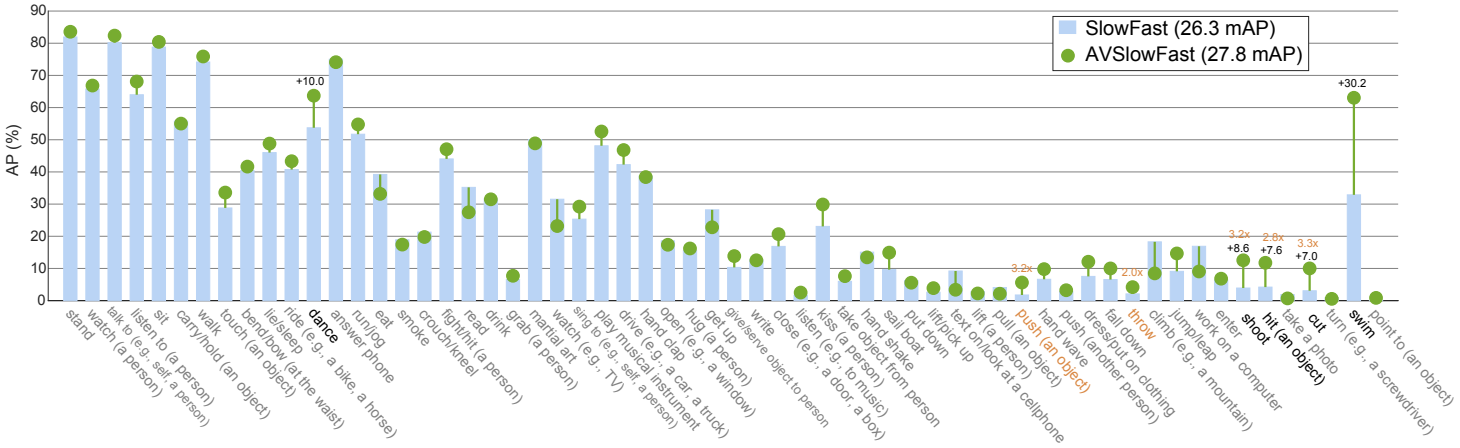

Figure 4. **AVA per-class average precision**. AVSlowFast (27.8 mAP) *vs*. its SlowFast counterpart (26.3 mAP). The highlighted categories are the 5 highest absolute increase (**bold**) and 5 highest relative increase from SlowFast (**orange**). Best viewed in color, zoomed in.

A.3. Results: Classification & Detection Analysis

Comparison to GBlend We also explored using Gradient Blending (GBlend) [78] to train our base AVSlowFast 4×16, R50 model. Specifically, we add a prediction head from audio feature only, to classify Kinetics classes, and set w (the weight for loss) for audio to be 1/3, which is approximately the ratio between the accuracy of audio and visual models (24.8% *vs*. 75.6%). Also, we turn off DropPathway such that we can compare it to the effects of GBlend in facilitating multi-modal training. We keep other hyper-parameters to be the same. Interestingly, we found that GBlend does not help in our setting, as it yields 75.1% top-1 accuracy (*vs*. 75.6% for our visual SlowFast model and 77.0% for our AVSlowFast model trained with DropPathway). We hypothesize that this is because GBlend is targeted for late-fusion of audio and visual streams.

Per-class analysis on Kinetics Comparing AVSlowFast to SlowFast (77.0% *vs*. 75.6% for 4×16, R50 backbone), classes that benefited most from audio include ["*dancing macarena*" +24.5%], ["*whistling*" +24.0%], ["*beatboxing*" +20.4%], ["*salsa dancing*" +19.1%] and ["*singing*" +16.0%], etc. Clearly, all these classes have distinct sound signatures to be recognized. On the other hand, classes like ["*skiing (not slalom or crosscountry)*" -12.3%], ["*triple jump*" -12.2%], ["*dodgeball*" -10.2%] and ["*massaging legs*" -10.2%] have the largest performance loss, as sound of these classes tend to be much less correlated the action.

Per-class analysis on AVA We compare per-class results of AVSlowFast to its SlowFast counterparts in Fig. 4. As mentioned in the main paper, classes with largest absolute gain (marked with bold black font) are "*swim*", "*dance*", "*shoot*", "*hit (an object)*" and "*cut*". Further, the classes "*push (an object)*" (3.2×) and "*throw*" (2.0×) largely benefit from audio in relative terms (marked with orange font in Fig. 4). As expected, all these classes have strong sound signature that are easy to recognize from audio. On the other hand, the largest performance loss arises for classes such as "watch (e.g., TV)", "read", "eat" and "work on a computer", which either do not have a distinct sound signature ("read", "work on a computer") or have strong background noise sound ("watch (e.g., TV)"). We believe explicitly modeling foreground and background sound might be a fruitful future direction to alleviate these challenges.

A.4. Details: Kinetics Action Classification

We train our models on Kinetics from scratch without any pretraining. Our training and testing closely follows [13]. We use a synchronous SGD optimizer and train with 128 GPUs using the recipe in [22]. The mini-batch size is 8 clips per GPU (so the total mini-batch size is 1024). The initial base learning rate η is 1.6 and we decrease the it according to half-period cosine schedule [44]: the learning rate at the n-th iteration is $\eta \cdot 0.5[\cos(\frac{n}{n_{\max}}\pi) + 1]$, where $n_{\max}$ is the maximum training iterations. We adopt a linear warm-up schedule [22] for the first 8k iterations. We use a scale jittering range of [256, 340] pixels for R101 model to improve generalization [13]. To aid convergence, we initialize all models that use Non-Local blocks (NL) from their counterparts that are trained without NL. We only use NL on res_4 (instead of res_3+res_4 used in [79]). We train with Batch Normalization (BN) [32], and the BN statistics are computed within each 8 clips. Dropout [29] with rate 0.5 is used before the final classifier layer. In total, we train for 256 epochs (60k iterations with batch size 1024, for ~240k Kinetics videos) when $T \leq 4$ frames, and 196 epochs when the Slow pathway has $T > 4$ frames: it is sufficient to train shorter when a clip has more frames. We use momentum of 0.9 and weight decay of 10^{-4}.

A.5. Details: EPIC-Kitchens Classification

We fine-tune from Kinetics pretrained AVSlowFast 8×8, R101 (w/o NL) for this experiment. For fine-tuning, we freeze all BNs by converting them into affine layers. We train using a single machine with 8 GPUs. Initial base learning rate η is set to 0.01 and 0.0006 for verb and noun. We train with batch size 32 for 24k and 30k for verb and noun respectively. We use a step wise decay of the learning rate by a factor of 10× at 2/3 and 5/6 of full training. For simplicity, we only use a single center crop for testing.

A.6. Details: Charades Action Classification

We fine-tune from the Kinetics pretrained AVSlowFast 16×8, R101 + NL model, to account for the longer activity range of this dataset, and a per-class sigmoid output is used to account for the mutli-class nature of the data. We train on a single machine (8 GPUs) for 40k iterations using a batch size of 8 and a base learning rate η of 0.07 with one 10× decay after 32k iterations. We use a Dropout rate of 0.7. For inference, we temporally max-pool scores [79, 13]. All other settings are the same as those of Kinetics.

A.7. Details: AVA Action Detection

We follow the detection architecture introduced in [13], which is adapted from Faster R-CNN [55] for video. Specifically, we set the spatial stride of res_5 from 2 to 1, thus increasing the spatial resolution of res_5 by 2×. RoI features are then computed by applying RoIAlign [26] spatially and global average pooling temporally. These features are then fed to a per-class, sigmoid-based classifier for multi-label prediction. Again, we initialize from Kinetics pretrained models and train 52k iterations with initial learning rate η of 0.4 and batch size 16 (we train across 16 machines, so effective batch size 16×16=256). We pre-compute proposals using an off-the-shelf Faster R-CNN person detector with ResNeXt-101-FPN backbone. It is pretrained on ImageNet and the COCO human keypoint data and more details can be found in [13, 81].

B. Details: Kinetics-Sound dataset

The original 34 classes selected in [2] are based on an earlier version of the Kinetics dataset. Some classes are removed since then. Therefore, we use the following 32 classes that are kept in current version of Kinetics-400 dataset: "blowing nose", "blowing out candles", "bowling", "chopping wood", "dribbling basketball", "laughing", "mowing lawn", "playing accordion", "playing bagpipes", "playing bass guitar", "playing clarinet", "playing drums", "playing guitar", "playing harmonica", "playing keyboard", "playing organ", "playing piano", "playing saxophone", "playing trombone", "playing trumpet", "playing violin", "playing xylophone", "ripping paper", "shoveling snow", "shuffling cards", "singing", "stomping grapes", "strumming guitar", "tap dancing", "tapping guitar", "tapping pen", "tickling".

References

[1] Huda Alamri, Chiori Hori, Tim K Marks, Dhruv Batra, and Devi Parikh. Audio visual scene-aware dialog (avsd) track for natural language generation in dstc7. In *DSTC7 at AAAI Workshop*, 2018. 2

[2] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. 1, 2, 5, 6, 8, 9, 11

[3] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *ECCV*, 2018. 1, 2, 5

[4] John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017. 2

[5] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NIPS*, 2016. 1, 2, 5

[6] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018. 5

[7] Lynne Bernstein, Edward Auer, Jintao Jiang, and Silvio Eberhardt. Auditory perceptual learning for speech perception can be enhanced by audiovisual training. *Frontiers in Neuroscience*, 2013. 3

[8] Yunlong Bian, Chuang Gan, Xiao Liu, Fu Li, Xiang Long, Yandong Li, Heng Qi, Jie Zhou, Shilei Wen, and Yuanqing Lin. Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. *arXiv preprint arXiv:1708.03805*, 2017. 2, 6, 9

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2, 6

[10] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *ACCV*, 2016. 5

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 2, 5

[12] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *ECCV*, 2018. 6

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *ICCV*, 2019. 1, 2, 3, 4, 5, 6, 8, 10, 11

[14] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016. 2

[15] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 2, 9

[16] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 9

[17] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016. 2

[18] R. Gao, R. Feris, and K. Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018. 1, 2

[19] Bernard Ghanem, Juan Carlos Niebles, Cees Snoek, Fabian Caba Heilbron, Humam Alwassel, Victor Escorcia, Ranjay Khrisna, Shyamal Buch, and Cuong Duc Dao. The ActivityNet large-scale activity recognition challenge 2018 summary. *arXiv preprint arXiv:1808.03766*, 2018. 1, 2, 8, 9

[20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 8, 9

[21] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video Action Transformer Network. In *CVPR*, 2019. 8

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

[23] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 2, 5, 8

[24] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, 2019. 5, 9

[25] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 9

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 11

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[28] 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 *ICASSP*, 2017. 2, 9

[29] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 10

[30] Jinggang Huang and David Mumford. Statistics of natural images and models. In *CVPR*, 1999. 4

[31] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *CVPR*, 2019. 6

[32] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 10

[33] Jianwen Jiang, Yu Cao, Lin Song, Shiwei Zhang Yunkai Li, Ziyao Xu, Qian Wu, Chuang Gan, Chi Zhang, and Gang Yu. Human centric spatio-temporal action localization. In *ActivityNet Workshop on CVPR*, 2018. 8

[34] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018. 8, 9

[35] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 5

[36] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. EPIC-Fusion: Audio-Visual Temporal Binding for Egocentric Action Recognition. *arXiv preprint arXiv:1908.08498*, 2019. 1, 2, 4, 6, 7

[37] Christian Keysers, Evelyne Kohler, M Alessandra Umiltà, Luca Nanetti, Leonardo Fogassi, and Vittorio Gallese. Audiovisual mirror neurons and action recognition. *Experimental brain research*, 2003. 1, 2, 5

[38] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019. 9

[39] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NIPS*, 2018. 2, 5, 9

[40] Bruno Korbar, Du Tran, and Lorenzo Torresani. SCSampler: Sampling salient clips from video for efficient action recognition. *arXiv preprint arXiv:1904.04289*, 2019. 5, 8, 9

[41] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 8

[42] François Laurent, Mario Valderrama, Michel Besserve, Mathias Guillard, Jean-Philippe Lachaux, Jacques Martinerie, and Geneviève Florence. Multimodal information improves the rapid detection of mental fatigue. *Biomedical Signal Processing and Control*, 2013. 2

[43] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017. 9

[44] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 10

[45] Dominic W Massaro and Michael M Cohen. Tests of auditory–visual integration efficiency within the framework of the fuzzy logical model of perception. *The Journal of the Acoustical Society of America*, 2000. 1

[46] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 1976. 1

[47] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 8, 9

[48] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *ICML*, 2011. 2

[49] Kei Omata and Ken Mogi. Fusion and combination in audio-visual integration. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2007. 1

[50] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018. 1, 2, 5, 8, 9

[51] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016. 1, 2, 5

[52] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 150:109–125, 2016. 9

[53] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 2003. 2

[54] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 2

[55] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 8, 11

[56] Daniel L Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5(4):517–548, 1994. 4

[57] Jean-Luc Schwartz, Frederic Berthommier, and Christophe Savariaux. Audio-visual scene analysis: evidence for a "very-early" integration process in audio-visual speech perception. In *International Conference on Spoken Language Processing*, 2002. 1

[58] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *CVPR*, 2018. 1, 2

[59] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, 2017. 6

[60] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2, 5

[61] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NIPS*, 2014. 2, 9

[62] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015. 2, 5

[63] Khurram Soomro, Amir Roshan Zamir, and M Shah. A dataset of 101 human action classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 8

[64] Salvador Soto-Faraco, Daria Kvasova, Emmanuel Biau, Nara Ikumi, Manuela Ruzzoli, Luis Morís-Fernández, and Mireia Torralba. *Multisensory Interactions in the Real World*. Elements in Perception. Cambridge University Press, 2019. 1

[65] Barry E Stein. *The new handbook of multisensory processing*. MIT Press, 2012. 1

[66] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. FBK-HUPBA Submission to the EPIC-Kitchens 2019 Action Recognition Challenge. *arXiv preprint arXiv:1906.08960*, 2019. 6

[67] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Hierarchical feature aggregation networks for video action recognition. *arXiv preprint arXiv:1905.12462*, 2019. 6

[68] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019. 8, 9

[69] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018. 8

[70] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 2

[71] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2

[72] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[73] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. *arXiv preprint arXiv:1904.02811*, 2019. 6

[74] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 6

[75] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 9

[76] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, 2018. 6

[77] 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 *ECCV*, 2016. 2

[78] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal networks hard? *arXiv preprint arXiv:1905.12681*, 2019. 2, 6, 9, 10

[79] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 4, 5, 6, 7, 8, 10, 11

[80] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 6

[81] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019. 6, 8, 11

[82] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Compressed video action recognition. In *CVPR*, 2018. 6

[83] S Xie, C Sun, J Huang, Z Tu, and K Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*. 6

[84] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning:

Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 2

[85] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 6

[86] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: Efficient convolutional network for online video understanding. In *ECCV*, 2018. 6