Anticipative Video Transformer

Rohit Girdhar[†] Kristen Grauman[†‡]
[†]Facebook AI Research [‡]University of Texas, Austin
http://facebookresearch.github.io/AVT

Abstract

We propose Anticipative Video Transformer (AVT), an end-to-end attention-based video modeling architecture that attends to the previously observed video in order to anticipate future actions. We train the model jointly to predict the next action in a video sequence, while also learning frame feature encoders that are predictive of successive future frames' features. Compared to existing temporal aggregation strategies, AVT has the advantage of both maintaining the sequential progression of observed actions while still capturing long-range dependencies—both critical for the anticipation task. Through extensive experiments, we show that AVT obtains the best reported performance on four popular action anticipation benchmarks: EpicKitchens-55, EpicKitchens-100, EGTEA Gaze+, and 50-Salads; and it wins first place in the EpicKitchens-100 CVPR'21 challenge.

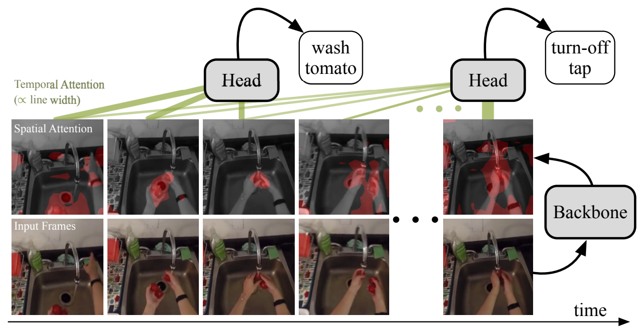

Figure 1: Anticipating future actions using AVT involves encoding video frames with a spatial-attention backbone, followed by a temporal-attention head that attends only to frames before the current one to predict future actions. In this example, it spontaneously learns to attend to hands and objects without being supervised to do so. Moreover, it attends to frames most relevant to predict the next action. For example, to predict 'wash tomato' it attends equally to all previous frames as they determine if any more tomatoes need to be washed, whereas for 'turn-off tap' it focuses most on the current frame for cues whether the person might be done. Please see § 5.3 for details and additional results.

1. Introduction

Predicting future human actions is an important task for AI systems. Consider an autonomous vehicle at a stop sign that needs to predict whether a pedestrian will cross the street or not. Making this determination requires modeling complex visual signals—the past actions of the pedestrian, such as speed and direction of walking, or usage of devices that may hinder his awareness of the surroundings—and using those to predict what he may do next. Similarly, imagine an augmented reality (AR) device that observes a user's activity from a wearable camera, *e.g*. as they cook a new dish or assemble a piece of furniture, and needs to anticipate his next steps to provide timely assistance. In many such applications, it is insufficient to *recognize* what is happening in the video. Rather, the vision system must also *anticipate* the likely actions that are to follow. Hence, there is a growing interest in formalizing the *activity anticipation* task [24, 45, 49, 64, 73, 82] along with development of multiple challenge benchmarks to support it [13, 14, 49, 55, 82].

Compared to traditional action recognition, anticipation tends to be significantly more challenging. First of all, it requires going beyond classifying current spatiotemporal visual patterns into a single action category—a task nicely suited to today's well-honed discriminative models—to instead predict the multi-modal distribution of future activities. Moreover, while action recognition can often side-step temporal reasoning by leveraging instantaneous contextual cues [31], anticipation inherently requires modeling the progression of past actions to predict the future. For instance, the presence of a plate of food with a fork may be sufficient to indicate the action of eating, whereas anticipating that same action would require recognizing and reasoning over the sequence of actions that precede it, such as chopping, cooking, serving, *etc*. Indeed, recent work [23, 77] finds that modeling long temporal context is often critical for anticipation, unlike action recognition where frame-level modeling is often enough [43, 50, 81]. These challenges are also borne out in practice. For example, accuracy for one of today's top performing video models [77] drops from 42% to 17% when treating recognition versus anticipation on the same test clips [13]—predicting even one second into the future is much harder than declaring the current action.

The typical approach to solving long-term predictive reasoning tasks involves extracting frame or clip level features using standard architectures [12, 86, 91], followed by aggregation using clustering [32, 62], recurrence [23, 24, 42], or attention [28, 59, 77, 95] based models. Except the recurrent ones, most such models merely aggregate features over the temporal extent, with little regard to modeling the sequential temporal evolution of the video over frames. While recurrent models like LSTMs have been explored for anticipation [2, 23, 96], they are known to struggle with modeling long-range temporal dependencies due to their sequential (non-parallel) nature. Recent work mitigates this limitation using attention-based aggregation over different amounts of the context to produce short-term ('recent') and long-term ('spanning') features [77]. However, it still reduces the video to multiple aggregate representations and loses its sequential nature. Moreover, it relies on careful and dataset-specific tuning of the architecture and the amounts of context used for the different aggregate features.

In this work, we introduce *Anticipative Video Transformer* (AVT), an alternate video modeling architecture that replaces "aggregation" based temporal modeling with a *anticipative*[1] architecture. Aiming to overcome the tradeoffs described above, the proposed model naturally embraces the sequential nature of videos, while minimizing the limitations that arise with recurrent architectures. Similar to recurrent models, AVT can be rolled out indefinitely to predict further into the future (*i.e*. generate future predictions), yet it does so while processing the input in parallel with long-range attention, which is often lost in recurrent architectures.

Specifically, AVT leverages the popular transformer architecture [89, 92] with *causal*[2] masked attention, where each input frame is allowed to attend only to frames that precede it. We train the model to jointly predict the next action while also learning to predict future features that match the true future features and (when available) their intermediate action labels. Figure 1 shows examples of how AVT's spatial and temporal attention spreads over previously observed frames for two of its future predictions (wash tomato and turn-off tap). By incorporating intermediate future prediction losses, AVT encourages a predictive video representation that picks up patterns in how the visual activity is likely to unfold into the future. This facet of our model draws an analogy to language, where transformers trained with massive text corpora are now powerful tools to anticipate sequences of words (*cf*. GPT and variants [8, 69, 70]). The incremental temporal modeling aspect has been also been explored for action recognition [53], albeit with convolutional architectures and without intermediate self-supervised losses.

While the architecture described so far can be applied on top of various frame or clip encoders (as we will show in experiments), we further propose a *purely attention-based video modeling architecture* by replacing the backbone with an attention-based frame encoder from the recently introduced Vision Transformer [18]. This enables AVT to attend not only to specific frames, but also to spatial features within the frames in one unified framework. As we see in Figure 1, when trained on egocentric video, the model spontaneously learns to attend to spatial features corresponding to hands and objects, which tend to be especially important in anticipating future activities [57].

In summary, our contributions are: 1) AVT, a novel end-to-end purely attention based architecture for predictive video modeling; 2) Incorporation of a self-supervised future prediction loss, making the architecture especially applicable to predictive tasks like action anticipation; 3) Extensive analysis and ablations of the model showing its versatility with different backbone architectures, pre-trainings, *etc*. on the most popular action anticipation benchmarks, both from first and third person viewpoints. Specifically, we outperform all published prior work on EpicKitchens-55[3] [13], EpicKitchens-100[3] [14], EGTEA Gaze+ [55], and 50-Salads [82]. Most notably, our method outperforms all submissions to the EpicKitchens-100 CVPR'21 challenge[4], and is ranked #1 on the EpicKitchens-55 leaderboard[5] for seen (S1) and #2 on unseen (S2) test sets.

2. Related Work

Action anticipation is the task of predicting future actions given a video clip. While well explored in third-person video [2, 26, 38, 39, 47, 49, 82, 90], it has recently gained in popularity for first-person (egocentric) videos [13, 14, 16, 24, 57, 64, 77], due to its applicability on wearable computing platforms. Various approaches have been proposed for this task, such as learning representations by predicting future features [90, 96], aggregating past features [24, 77], or leveraging affordances and hand motion [57, 64]. Our work contributes a new video architecture for anticipation, and we demonstrate its promising advantages on multiple popular anticipation benchmarks.

Self-supervised feature learning from video methods learn representations from unlabeled video, often to be fine-

[1]We use the term "anticipative" to refer to our model's ability to predict future video features and actions.

[2]Throughout we use the term "causal" to refer to the constraint that video be processed in a forward, online manner, *i.e*. functions applied at time t can only reference the frames preceding them, akin to Causal Language Modeling (CLM) [51]. This is not to be confused with other uses of "causal" in AI where the connotation is instead cause-and-effect.

[3]EpicKitchens-55/100 datasets are licensed under the Creative Commons Attribution-NonCommercial 4.0 International License.

[4]competitions.codalab.org/competitions/25925

[5]competitions.codalab.org/competitions/20071

tuned for particular downstream tasks. Researchers explore a variety of "free" supervisory signals, such as temporal consistency [21, 41, 44, 94, 99], inter-frame predictability [36, 37, 40, 83], and cross-modal correspondence [3, 48, 83, 84]. AVT incorporates losses that encourage features predictive of future features (and actions); while this aspect shares motivation with prior [25, 36, 37, 58, 60, 75, 78, 83, 84, 90] and concurrent work [96], our architecture to achieve predictive features is distinct (transformer based rather than convolutional/recurrent [25, 36, 37, 78, 96]), it operates over raw frames or continuous video features as opposed to clustered 'visual words' [84], assumes only visual data (rather than vision with speech or text [83, 84]), and is jointly trained for action anticipation (rather than pre-trained and then fine-tuned for action recognition [36, 37, 83]).

Language modeling (LM) has been revolutionized with the introduction of self-attention architectures [89]. LM approaches can generally be classified in three categories: (1) *encoder-only* [17, 67], which leverage bidirectional attention and are effective for discriminative tasks such as classification; (2) *decoder-only* [8, 69], which leverage a causal attention [51] attending on past tokens, and are effective for generative tasks such as text generation; and (3) *encoder-decoder* [52, 71], which incorporate both a bidirectional encoder and causal decoder, and are effective for tasks such as machine translation. Capitalizing on the analogy between action prediction and generative language tasks, we explore causal decoder-only attention architectures in our model. While language models are typically trained on discrete inputs (words), AVT trains with continuous video features. This distinction naturally influences our design choices, such as an L_2 loss for generative training as opposed to a cross entropy loss for the next word.

Self-attention and transformers in vision. The general idea of self-attention in vision dates back to non-local means [9], and is incorporated into contemporary network architectures as non-local blocks [10, 56, 93, 95] and gating mechanisms [30, 46, 62, 97]. While self-attention approaches like transformers [89, 92] offer strong results for high-level vision reasoning tasks [11, 101], more recently, there is growing interest in completely replacing convolutional architectures with transformers for image recognition [18, 85]. For video, prior work has mostly leveraged attention architectures [28, 93, 95] on top of standard spatiotemporal convolutional base architectures [12, 86, 88]. In contrast, AVT is an end-to-end transformer architecture for video—to our knowledge the first (concurrent with [4, 7, 19, 54, 65]). Unlike the concurrent methods [4, 7, 19, 54, 65], which are bidirectional and address traditional action recognition, AVT has a causal structure and tackles predictive tasks (anticipation). AVT yields the best results to date for several well-studied anticipation benchmarks.

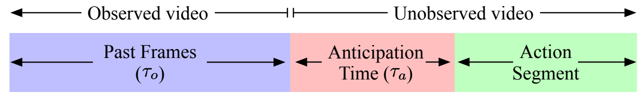

Figure 2: Action anticipation problem setup. The goal is to use the observed video segment of length τ_o to anticipate the future action τ_a seconds before it happens.

3. Anticipation Problem Setup

While multiple anticipation problem setups have been explored in the literature [45, 64, 73], in this work we follow the setup defined in recent challenge benchmarks [13, 14] and illustrated in Figure 2. For each action segment labeled in the dataset starting at time τ_s, the goal is to recognize it using a τ_o length video segment τ_a units before it, *i.e.* from $\tau_s - (\tau_a + \tau_o)$ to $\tau_s - \tau_a$. While methods are typically allowed to use any length of observed segments (τ_o), the anticipation time (τ_a) is usually fixed for each dataset.

4. Anticipative Video Transformer

We now present the AVT model architecture, as illustrated in Figure 3. It is designed to predict future actions given a video clip as input. To that end, it leverages a two-stage architecture, consisting of a *backbone* network that operates on individual frames or short clips, followed by a *head* architecture that operates on the frame/clip level features to predict future features and actions. AVT employs *causal* attention modeling—predicting the future actions based only on the frames observed so far—and is trained using objectives inspired from self-supervised learning. We now describe each model component in detail, followed by the training and implementation details.

4.1. Backbone Network

Given a video clip with T frames, $V = \{\mathbf{X}_1, \cdots, \mathbf{X}_T\}$ the backbone network, $\mathcal{B}$, extracts a feature representation for each frame, $\{\mathbf{z}_1, \cdots, \mathbf{z}_T\}$ where $\mathbf{z}_t = \mathcal{B}(\mathbf{X}_t)$. While various video base architectures have been proposed [12, 20, 87, 91] and can be used with AVT as we demonstrate later, in this work we propose an alternate architecture for video understanding based purely on attention. This backbone, which we refer to as AVT-b, adopts the recently proposed Vision Transformer (ViT) [18] architecture, which has shown impressive results for static image classification.

Specifically, we adopt the ViT-B/16 architecture. We split each input frame into 16×16 non-overlapping patches. We flatten each patch into a 256D vector, and linearly project them to 768D, which is the feature dimension used throughout the encoder. While we do not need to classify each frame individually, we still prepend a learnable `[class]` token embedding to the patch features, whose

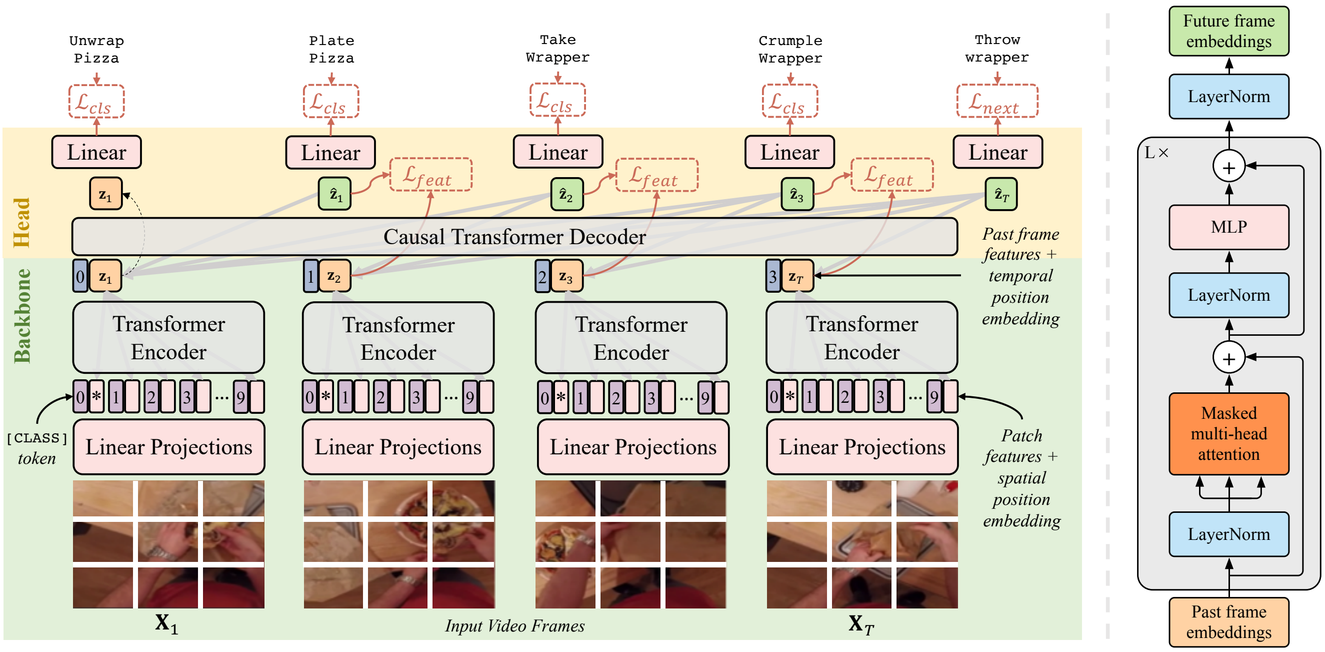

Figure 3: *(Left)* **AVT architecture.** We split the T input frames into non-overlapping patches that are linearly projected. We add a learned [CLASS] token, along with spatial position embeddings, and the resulting features are passed through multiple layers of multi-head attention, with shared weights across the transformers applied to all frames. We take the resulting features corresponding to the [CLASS] token, append a temporal position encoding and pass it through the Causal Transformer Decoder that predicts the future feature at frame t, after attending to all features from $1 \cdots t$. The resulting feature is trained to regress to the true future feature ($\mathcal{L}_{feat}$) and predict the action at that time point if labeled ($\mathcal{L}_{cls}$), and the last prediction is trained to predict the future action ($\mathcal{L}_{next}$). *(Right)* **Causal Transformer Decoder.** It follows the Transformer architecture with pre-norm [92], causal masking in attention, and a final LayerNorm [70].

output will be used as a frame-level embedding input to the head. Finally, we add learned position embeddings to each patch feature similar to [18]. We choose to stick to frame-specific spatial position encodings, so that the same backbone model with shared weights can be applied to each frame. We will incorporate the temporal position information in the head architecture (discussed next). The resulting patch embeddings are passed through a standard Transformer Encoder [89] with pre-norm [92]. We refer the reader to [18] for details of the encoder architecture.

AVT-b is an attractive backbone design because it makes our architecture purely attentional. Nonetheless, in addition to AVT-b, AVT is compatible with other video backbones, including those based on 2D CNNs [80, 91], 3D CNNs [12, 20, 87], or fixed feature representations based on detected objects [5, 6] or visual attributes [63]. In § 5 we provide experiments testing several such alternatives. For the case of spatiotemporal backbones, which operate on clips as opposed to frames, we extract features as $\mathbf{z}_t = \mathcal{B}(\mathbf{X}_{t-L}, \cdots, \mathbf{X}_t)$, where the model is trained on L-length clips. This ensures the features at frame t do not incorporate any information from the future, which is not allowed in the anticipation problem setting.

4.2. Head Network

Given the features extracted by the backbone, the head network, referred to as AVT-h, is used to predict the future features for each input frame using a Causal Transformer Decoder, $\mathcal{D}$:

$$\hat{\mathbf{z}}_1, \cdots, \hat{\mathbf{z}}_T = \mathcal{D}(\mathbf{z}_1, \cdots, \mathbf{z}_T). \quad (1)$$

Here $\hat{\mathbf{z}}_t$ is the predicted future feature corresponding to frame feature $\mathbf{z}_t$, after attending to all features before and including it. The predicted features are then decoded into a distribution over the semantic action classes using a linear classifier θ, *i.e.* $\hat{\mathbf{y}}_t = \theta(\hat{\mathbf{z}}_t)$. The final prediction, $\hat{\mathbf{y}}_T$, is used as the model's output for the next-action anticipation task. Note that since the next action segment $(T+1)$ is τ_a seconds from the last observed frame (T) as per the problem setup, we typically sample frames at a stride of τ_a so that the model learns to predict future features/actions at that frame rate. However, empirically we find the model is robust to other frame rate values as well.

We implement $\mathcal{D}$ using a masked transformer decoder inspired from popular approaches in generative language modeling, such as GPT-2 [70]. We start by adding a temporal position encoding to the frame features implemented as a learned embedding of the absolute frame position within the clip. The embedded features are then passed through multiple decoder layers, each consisting of masked multi-head attention, LayerNorm (LN) and a multi-layer perceptron (MLP), as shown in Figure 3 (right). The final output is then passed through another LN, akin to GPT-2 [70], to obtain the future frame embeddings.

Aside from being visual rather than textual, this model differs from the original Transformer Decoder [89] in terms of the final LN and the masking operation in the multi-head attention. The masking ensures that the model only attends to specific parts of the input, which in the case of predictive tasks like ours, is defined as a 'causal' mask. That is, for the output corresponding to the future after frame t, *i.e.* $\hat{\mathbf{z}}_t$, we set the mask to only attend to $\mathbf{z}_1 \cdots \mathbf{z}_t$. We refer the reader to [70] for details on the masking implementation.

This design differs considerably from previous applications of language modeling architectures to video, such as VideoBERT [84]. It operates directly on continuous clip embeddings instead of first clustering them into tokens, and it leverages causal attention to allow for anticipative training (discussed next), instead of needing masked language modeling (MLM) as in BERT [17]. These properties make AVT suited for predictive video tasks while allowing for the long-range reasoning that is often lost in recurrent architectures. While follow-ups to VideoBERT such as CBT [83] operate on raw clip features, they still leverage a MLM objective with bidirectional attention, with the primary goal of representation learning as opposed to future prediction.

4.3. Training AVT

To sample training data, for each labeled action segment in a given dataset, we sample a clip preceding it and ending τ_a seconds before the start of the action. We pass the clip through AVT to obtain future predictions, and then supervise the network using three losses.

First, we supervise the next-action prediction using a cross-entropy loss with the labeled future action, c_{T+1}:

$$\mathcal{L}_{next} = -\log \hat{\mathbf{y}}_T[c_{T+1}]. \tag{2}$$

Second, to leverage the causal structure of the model, we supervise the model's intermediate future predictions at the *feature* level and the *action class* level. For the former, we predict future features to match the true future features that are present in the clip, *i.e.*

$$\mathcal{L}_{feat} = \sum_{t=1}^{T-1} ||\hat{\mathbf{z}}_t - \mathbf{z}_{t+1}||_2^2. \tag{3}$$

This loss is inspired from the seminal work by Vondrick *et al.* [90] as well as follow ups [36, 37] that show that anticipating future visual representations is an effective form of self-supervision, though typically for traditional action recognition tasks. Concurrent and recent work adopts similar objectives for anticipation tasks, but with recurrent architectures [25, 78, 96]. Whereas recent methods [36, 37, 96] explore this loss with NCE-style [66] objectives, in initial experiments we found simple L_2 loss to be equally effective. Since our models are always trained with the final supervised loss, we do not suffer from potential collapse during training that would necessitate the use of contrastive losses.

Third, as an *action class* level anticipative loss, we leverage any action labels available in the dataset to supervise the intermediate predictions, *i.e.*, when the input clip overlaps with any labeled action segments that precede the segment to be anticipated.[6] Setting $c_t = -1$ for any earlier frames for which we do not have labels, we incur the following loss:

$$\mathcal{L}_{cls} = \sum_{t=1}^{T-1} \mathcal{L}_{cls}^t; \quad \mathcal{L}_{cls}^t = \begin{cases} -\log \hat{\mathbf{y}}_t[c_{t+1}] & \text{if } c_{t+1} \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We train our model with

$$\mathcal{L} = \mathcal{L}_{next} + \mathcal{L}_{cls} + \mathcal{L}_{feat} \tag{5}$$

as the objective, and refer to it as the **anticipative** [a] training setting. As a baseline, we also experiment with a model trained solely with $\mathcal{L} = \mathcal{L}_{next}$, and refer to it as the **naive** [n] setting, as it does not leverage our model's causal attention structure, instead supervising only the final prediction which attends to the full input. As we will show in Table 7, the anticipative setting leads to significant improvements.

4.4. Implementation Details

We preprocess the input video clips by randomly scaling the height between 248 and 280px, and take 224px crops at training time. We sample 10 frames at 1FPS for most experiments. We adopt network architecture details from [18] for the AVT-b backbone. Specifically, we use a 12-head, 12-layer transformer encoder model that operates on 768D representations. We initialize the weights from a model pre-trained on ImageNet-1K (IN1k), ImageNet-21K (IN21k) or ImageNet-1K finetuned from ImageNet-21K (IN21+1k), and finetune end-to-end for the anticipation tasks. For AVT-h, we use a 4-head, 6-layer model that operates on a 2048D representation, initialized from scratch. We employ a linear layer between the backbone and head to project the features to match the feature dimensions used in the head. We train AVT end-to-end with SGD+momentum using 10^{-6} weight decay and 10^{-4} learning rate for 50 epochs, with a 20 epoch warmup [33] and 30 epochs of cosine annealed decay. At test time, we employ 3-crop testing, where we compute three 224px spatial crops from 248px input frames, and

[6] For example, this would be true for each frame for densely labeled datasets like 50-Salads, and a subset of frames for sparsely labeled datasets like EpicKitchens-55.

Dataset	Viewpoint	Segments	Classes	τ_a (s)	Metric(s)
EK100 [14]	1st	90.0K	3,807	1.0 [14]	recall
EK55 [13]	1st	39.6K	2,513	1.0 [13]	top-1/5, recall
EGTEA Gaze+ [55]	1st	10.3K	106	0.5 [57]	top-1, cm top-1
50S [82]	3rd	0.9K	17	1.0 [2]	top-1

Table 1: Datasets used for evaluation. We use four popular benchmarks, spanning first and third person videos. Class-mean ('cm') $\Longrightarrow$ evaluation is done per-class and averaged over classes. Recall refers to class-mean recall@5 from [22]. For all, higher is better.

Head	Backbone	Init	Verb	Noun	Action
RULSTM [14]	TSN	IN1k	27.5	29.0	13.3
AVT-h	TSN	IN1k	27.2	30.7	13.6
AVT-h	irCSN152	IG65M	25.5	28.1	12.8
AVT-h	AVT-b	IN1k	28.2	29.3	13.4
AVT-h	AVT-b	IN21+1k	28.7	**32.3**	14.4
AVT-h	AVT-b	IN21k	**30.2**	31.7	**14.9**
RULSTM [14]	Faster R-CNN	IN1k	17.9	23.3	7.8
AVT-h	Faster R-CNN	IN1k	**18.0**	**24.3**	**8.7**

Table 2: EK100 (val) using RGB and detected objects (OBJ) modalities separately. AVT outperforms prior work using the exact same features, and further improves with our AVT-b backbone. Performance reported using class-mean recall@5.

average the predictions over the corresponding three clips. The default backbone for AVT is AVT-b, based on the ViT-B/16 architecture. However, to enrich our comparisons with some baselines [23, 24, 77], below also we report performance of only our head model operating on fixed features from 1) a frame-level TSN [91] backbone pre-trained for action classification, or 2) a recent spatiotemporal convolutional architecture irCSN-152 [87] pre-trained on a large weakly labeled video dataset [27], which has shown strong results when finetuned for action recognition. We finetune that model for action classification on the anticipation dataset and extract features that are used by the head for anticipation. In these cases, we only train the AVT-h layers. For all datasets considered, we use the validation set or split 1 to further optimize the hyperparameters, and use that setup over multiple splits or the held out test sets. Code and models will be released for reproducibility.

5. Experiments

We empirically evaluate AVT on four popular action anticipation benchmarks covering both first- and third-person videos. We start by describing the datasets and evaluation protocols (§ 5.1), followed by key results and comparisons to the state of the art (§ 5.2), and finally ablations and qualitative results (§ 5.3).

5.1. Experimental Setup

Datasets and metrics. We test on four popular action anticipation datasets summarized in Table 1. *EpicKitchens-100 (EK100)* [14] is the largest egocentric (first-person) video dataset with 700 long unscripted videos of cooking activities totalling 100 hours. *EpicKitchens-55 (EK55)* [13] is an earlier version of the same, and allows for comparisons to a larger set of baselines which have not yet been reported on EK100. For both, we use the standard train, val, and test splits from [14] and [23] respectively to report performance. The test evaluation is performed on a held-out set through a submission to their challenge server. *EGTEA Gaze+* [55] is another popular egocentric action anticipation dataset. Following recent work [57], we report performance on the split 1 [55] of the dataset at $\tau_a = 0.5s$. Finally, *50-Salads (50S)* [82] is a popular third-person anticipation dataset, and we report top-1 accuracy averaged over the pre-defined 5 splits following prior work [77]. Some of these datasets employ top-5/recall@5 criterion to account for the multi-modality in future predictions, as well as class-mean (cm) metrics to equally weight classes in a long-tail distribution. The first three datasets also decompose the action annotations into verb and nouns. While some prior work [77] supervises the model additionally for nouns and verbs, we train all our model solely to predict actions, and estimate the verb/noun probabilities by marginalizing over the other, similar to [23]. In all tables, we highlight the columns showing the metric used to rank methods in the official challenge leaderboards. Unless otherwise specified, the reported metrics correspond to future action (act.) prediction, although we do report numbers for verb and nouns separately where applicable. Please see Appendix A for further details.

Baselines. We compare AVT to its variants with different backbones and pretrained initializations, as well as to the strongest recent approaches for action anticipation, *i.e.* RULSTM [23, 24], ActionBanks [77], and Forecasting HOI (FHOI) [57]. Please see Appendix B for details on them. While FHOI trains the model end-to-end, RULSTM and ActionBanks operate on top of features from a model pre-trained for action classification on that dataset. Hence, we report results both using the exact same features as well as end-to-end trained backbones to facilitate fair comparisons.

5.2. Comparison to the state-of-the-art

EK100. We first compare AVT to prior work using individual modalities (RGB and Obj [23]) in Table 2 for apples-to-apples comparisons and to isolate the performance of each of our contributions. First, we compare to the state-of-the-art RULSTM method using only our AVT (head) model applied to the exact same features from TSN [91] trained for classification on EK100. We note this already improves over RULSTM, particularly in anticipating future objects (nouns). Furthermore, we experiment with backbone fea-

Split	Method	Overall			Unseen Kitchen			Tail Classes		
		Verb	Noun	Act	Verb	Noun	Act	Verb	Noun	Act
Val	chance	6.4	2.0	0.2	14.4	2.9	0.5	1.6	0.2	0.1
	RULSTM [14]	27.8	30.8	14.0	28.8	**27.2**	**14.2**	19.8	22.0	11.1
	AVT+ (TSN)	25.5	31.8	14.8	25.5	23.6	11.5	18.5	25.8	12.6
	AVT+	**28.2**	**32.0**	**15.9**	**29.5**	23.9	11.9	**21.1**	**25.8**	**14.1**
Test	chance	6.2	2.3	0.1	8.1	3.3	0.3	1.9	0.7	0.0
	RULSTM [14]	25.3	26.7	11.2	19.4	26.9	9.7	17.6	16.0	7.9
	TBN [100]	21.5	26.8	11.0	20.8	**28.3**	**12.2**	13.2	15.4	7.2
	AVT+	**25.6**	**28.8**	**12.6**	**20.9**	22.3	8.8	**19.0**	**22.0**	**10.1**
Challenge	IIE_MRG	25.3	26.7	11.2	19.4	26.9	9.7	17.6	16.0	7.9
	NUS_CVML [76]	21.8	30.6	12.6	17.9	27.0	10.5	13.6	20.6	8.9
	ICL+SJTU [35]	**36.2**	32.2	13.4	**27.6**	24.2	10.1	**32.1**	**29.9**	11.9
	Panasonic [98]	30.4	**33.5**	14.8	21.1	27.1	10.2	24.6	27.5	12.7
	AVT++	25.2	32.0	**16.5**	20.4	**27.9**	**12.8**	17.6	23.5	**13.6**

Table 3: EK100 val and test sets using all modalities. We split the test comparisons between published work and CVPR'21 challenge submissions. We outperform prior work including all challenge submissions, with especially significant gains on tail classes. Performance is reported using class-mean recall@5. AVT+ and AVT++ late fuse predictions from multiple modalities; please see text for details.

Head	Backbone	Init	Top-1	Top-5	Recall
RULSTM [24]	TSN	IN1k	13.1	30.8	12.5
ActionBanks [77]	TSN	IN1k	12.3	28.5	13.1
AVT-h	TSN	IN1k	13.1	28.1	13.5
AVT-h	AVT-b	IN21+1k	12.5	30.1	**13.6**
AVT-h	irCSN152	IG65M	**14.4**	**31.7**	13.2

Table 4: EK55 using only RGB modality for action anticipation. AVT performs comparably, and outperforms when combined with a backbone pretrained on large weakly labeled dataset.

Method	Top-1 acc.			Class mean acc.		
	Verb	Noun	Act.	Verb	Noun	Act.
I3D-Res50 [12]	48.0	42.1	34.8	31.3	30.0	23.2
FHOI [57]	49.0	45.5	36.6	32.5	32.7	25.3
AVT-h (+TSN)	51.7	50.3	39.8	41.2	41.4	28.3
AVT	**54.9**	**52.2**	**43.0**	**49.9**	**48.3**	**35.2**

Table 5: EGTEA Gaze+ Split 1 at $\tau_a = 0.5s$. AVT outperforms prior work by significant margins, especially when trained end-to-end with the AVT-b backbone.

Head	Top-1
DMR [90]	6.2
RNN [2]	30.1
CNN [2]	29.8
ActionBanks [77]	40.7
AVT	**48.0**

Table 6: 50-Salads. AVT outperforms prior work even in 3rd person videos.

tures from a recent state-of-the-art video model, irCSN-152 [87] pretrained on a large weakly supervised dataset, IG65M [27]. We finetune this backbone for recognition on EK100, extract its features and train AVT-h same as before, but find it to not be particularly effective at the EK100 anticipation task. Next, we replace the backbone with our AVT-b and train the model end-to-end, leading to the best performance so far, and outperforming RULSTM by 1.6%. We make the same comparison over features from an object-detector [72] trained on EK100 provided by RULSTM (referred to as OBJ modality, details in Appendix A), and similarly find our method outperforms RULSTM on this modality as well.

Note that the fixed features used above can be thought of as a proxy for past recognized actions, as they are trained only for action recognition. Hence, AVT-h on TSN or irCSN152 features is comparable to a baseline that trains a language model over past actions to predict future ones. As the later experiments show, end-to-end trained AVT is significantly more effective, supporting AVT's *from-pixels* anticipation as opposed to *label-space* anticipation.

Finally, we compare models using all modalities on the EK100 val and the held-out test set in Table 3. While RULSTM fuses models trained on RGB, Flow, and OBJ features using an attention based model (MATT [23]), we simply late fuse predictions from our best RGB and OBJ models (resulting model referred to as AVT+), and outperform all reported work on this benchmark, establishing a new state-of-the-art. Note we get the largest gains on tail classes, suggesting our model is particularly effective at few-shot anticipation. Finally, AVT++ ensembles multiple model variants, and outperforms all submissions on the EK100 CVPR'21 challenge leaderboard. Please refer to the workshop paper [29] for details on AVT++.

EK55. Since EK100 is relatively new and has few baseline methods reported, we also evaluate AVT on EK55. As before, we start by comparing single modality methods (RGB-only) in Table 4. For AVT-h models, we found a slightly different set of (properly validated) hyperparameters performed better for top-1/5 metrics vs. the recall metric, hence we report our best models for each set of results. Here we find AVT-h performs comparably to RULSTM, and outperforms another attention-based model [77] (one of the winners of the EK55 2020 challenge) on the top-1 metrics. The gain is more significant on the recall metric, which averages performance over classes, indicating again that AVT-h is especially effective on tail classes which get ignored in top-1/5 metrics. Next, we replace the backbone with AVT-b, and find it to perform comparably on top-1/5 metrics, and outperforms on the recall metric. Finally, we experiment with irCSN-152 [87] pretrained using IG65M [27] and finetuned on EK55, and find it to outperform all methods by a significant margin on top-1/5. We show further comparisons with the state-of-the-art on EK55 in Appendix C.

EGTEA Gaze+. In Table 5 we compare our method at $\tau_a = 0.5s$ on the split 1 as in recent work [57]. Even using fixed features with AVT-h on top, AVT outperforms the best reported results, and using the AVT-b backbone further improves performance. Notably, FHOI leverages attention on hand trajectories to obtain strong performance, which, as we see in Figure 1, emerges spontaneously in our model.

50-Salads. Finally, we show that our approach is not lim-

	Losses		Backbones	
Setting	$\mathcal{L}_{cls}$	$\mathcal{L}_{feat}$	TSN	AVT-b
naive [n]	-	-	10.1	13.1
	✓	-	11.5	14.4
	-	✓	13.7	13.0
anticipative [a]	✓	✓	13.6	14.4

Table 7: Anticipative training. Employing the anticipative training losses are imperative to obtain strong performance with AVT. Reported on EK100/cm recall@5.

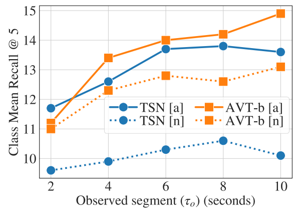

Figure 4: Temporal context. AVT effectively leverages longer temporal context, especially in the [a] setting.

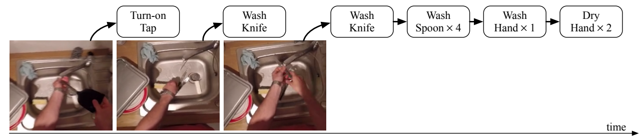

Figure 5: Long-term anticipation. AVT can also be used to predict further into the future by rolling out predictions autoregressively. The text on top represents the next action predicted at provided frames, followed by subsequently predicted actions, with the number representing how long that action would repeat.

ited to egocentric videos and is also effective in third-person settings. In Table 6, we report top-1 performance on 50-Salads averaged over standard 5 splits. We observe it outperforms previous RNN [2] and attention [77] based approaches by a significant 7.3% absolute improvement, again establishing a new state-of-the-art.

5.3. Ablations and Analysis

We now analyze the AVT architecture, using the RGB modality and EK100 validation set as the test bed.

Anticipative losses. In Table 7, we evaluate the contribution of the two intermediate prediction losses that leverage the causal structure of AVT. We find using those objectives leads to significant improvements for both backbones. We find $\mathcal{L}_{cls}$ is more effective for TSN, and $\mathcal{L}_{feat}$ for AVT-b. Given that both combined work well in both settings, we use both for all experiments. Note that the naive setting also serves as a baseline with AVT-b backbone followed by simple aggregation on top, and shows our proposed losses encouraging the predictive structure are imperative to obtain strong performance. We analyze per-class gains in Appendix D.1 and find classes like 'cook', which require understanding the sequence of actions so far to anticipate well, obtain the largest gains in the anticipative setting.

Temporal context. Next, we analyze the effect of temporal context. In Figure 4, we train and test the model with different lengths of temporal context, τ_o. We notice that the performance improves as we incorporate more frames of context, with more consistent gains for AVT-b. The gains are especially pronounced when trained using the anticipative setting ($11.2 \rightarrow 14.9 = 3.5 \uparrow$) vs. the naive ($11.0 \rightarrow 13.1 = 2.1 \uparrow$). This suggests end-to-end trained AVT using anticipative losses is better suited at modeling sequences of long-range temporal interactions.

Attention visualization. To better understand how AVT models videos, we visualize the learned attention in the backbone and head. For the backbone, following prior work [18], we use attention rollout [1] to aggregate attention over heads and layers. For the head, since our causal modeling would bias aggregated attention towards the first few frames, we visualize the last layer attention averaged over heads. As shown in Figure 1, the model spontaneously learns to attend to hands and objects, which has been found beneficial for egocentric anticipation tasks [57]—but required manual designation in prior work. The temporal attention also varies between focusing on the past or mostly on the current frame depending on the predicted future action. We show additional results in Appendix D.2.

Long-term anticipation. So far we have shown AVT's applicability in the next-action anticipation task. Thanks to AVT's predictive nature, it can also be rolled out autoregressively to predict a sequence of future actions given the video context. We append the predicted feature and run the model on the resulting sequence, reusing features computed for past frames. As shown in Figure 5, AVT makes reasonable future predictions—'wash spoon' after 'wash knife', followed by 'wash hand' and 'dry hand'—indicating the model has started to learn certain 'action schemas' [68], a core capability of our causal attention and anticipative training architecture. We show additional results in Appendix D.3.

6. Conclusion and Future Work

We presented AVT, an end-to-end attention-based architecture for anticipative video modeling. Through extensive experimentation on four popular benchmarks, we show its applicability in anticipating future actions, obtaining state-of-the-art results and demonstrating the importance of its anticipative training objectives. We believe AVT would be a strong candidate for tasks beyond anticipation, such as self-supervised learning [37, 90], discovering action schemas and boundaries [68, 79], and even for general action recognition in tasks that require modeling temporal ordering [34]. We plan to explore these directions in future work.

Acknowledgements: Authors would like to thank Antonino Furnari, Fadime Sener and Miao Liu for help with prior work; Naman Goyal and Myle Ott for help with language models; and Tushar Nagarajan, Gedas Bertasius and Laurens van der Maaten for feedback on the manuscript.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *ACL*, 2020.

[2] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *CVPR*, 2018.

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

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

[5] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020.

[6] Gedas Bertasius and Lorenzo Torresani. Cobe: Contextualized object embeddings from narrated instructional video. In *NeurIPS*, 2020.

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

[8] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[9] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.

[10] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCV Workshop*, 2019.

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[12] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 2017.

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

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

[15] Roeland De Geest and Tinne Tuytelaars. Modeling temporal structure with lstm for online action detection. In *WACV*, 2018.

[16] Eadom Dessalene, Michael Maynord, Chinmaya Devaraj, Cornelia Fermuller, and Yiannis Aloimonos. Forecasting action through contact representations from first person video. *TPAMI*, 2021.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[19] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021.

[20] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

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

[22] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *ECCV Workshop*, 2018.

[23] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *ICCV*, 2019.

[24] Antonino Furnari and Giovanni Maria Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *TPAMI*, 2020.

[25] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *ICCV*, 2019.

[26] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In *BMVC*, 2017.

[27] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.

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

[29] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer @ EPIC-Kitchens Action Anticipation Challenge 2021. In *CVPR Workshop*, 2021.

[30] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *NeurIPS*, 2017.

[31] Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning. In *ICLR*, 2020.

[32] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. ActionVLAD: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.

[33] 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.

[34] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

[35] Xiao Gu, Jianing Qiu, Yao Guo, Benny Lo, and Guang-Zhong Yang. Transaction: Icl-sjtu submission to epic-kitchens action anticipation challenge 2021. In *CVPR Workshop*, 2021.

[36] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshop*, 2019.

[37] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *ECCV*, 2020.

[38] De-An Huang and Kris M Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*, 2014.

[39] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *ICRA*, 2016.

[40] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *ICCV*, 2015.

[41] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *CVPR*, 2016.

[42] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[43] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[44] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019.

[45] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*, 2012.

[46] Shu Kong and Charless Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *CVPR*, 2017.

[47] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *TPAMI*, 2015.

[48] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[49] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014.

[50] Hilde Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[51] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. In *NeurIPS*, 2019.

[52] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[53] Xinyu Li, Bing Shuai, and Joseph Tighe. Directional temporal modeling for action recognition. In *ECCV*, 2020.

[54] Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. VidTr: Video transformer without convolutions. *arXiv preprint arXiv:2104.11746*, 2021.

[55] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018.

[56] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2019.

[57] Miao Liu, Siyu Tang, Yin Li, and James Rehg. Forecasting human object interaction: Joint prediction of motor attention and actions in first person video. In *ECCV*, 2020.

[58] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *CVPR*, 2018.

[59] Xiang Long, Chuang Gan, Gerard de Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*, 2018.

[60] Pauline Luc, Camille Couprie, Yann Lecun, and Jakob Verbeek. Predicting future instance segmentation by forecasting convolutional features. In *ECCV*, 2018.

[61] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, 2016.

[62] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. In *CVPR Workshop*, 2017.

[63] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *CVPR Workshop*, 2019.

[64] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. EGO-TOPO: Environment affordances from egocentric video. In *CVPR*, 2020.

[65] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

[66] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[67] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

[68] Jean Piaget. *La naissance de l'intelligence chez l'enfant*, page 216. 1935.

[69] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[70] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[71] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine

Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.
[72] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
[73] Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In *ICCV*, 2017.
[74] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *CVPR*, 2017.
[75] Cristian Rodriguez, Basura Fernando, and Hongdong Li. Action anticipation by predicting future dynamic images. In *ECCV Workshop*, 2018.
[76] Fadime Sener, Dibyadip Chatterjee, and Angela Yao. Technical report: Temporal aggregate representations. *arXiv preprint arXiv:2106.03152*, 2021.
[77] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *ECCV*, 2020.
[78] Yuge Shi, Basura Fernando, and Richard Hartley. Action anticipation with rbf kernelized feature mapping rnn. In *ECCV*, 2018.
[79] Mike Zheng Shou, Deepti Ghadiyaram, Weiyao Wang, and Matt Feiszli. Generic event boundary detection: A benchmark for event segmentation. *arXiv preprint arXiv:2101.10511*, 2021.
[80] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.
[81] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*, 2012.
[82] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *UbiComp*, 2013.
[83] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019.
[84] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *ICCV*, 2019.
[85] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention. In *ICML*, 2021.
[86] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.
[87] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019.
[88] 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.
[89] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
[90] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016.
[91] 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.
[92] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. In *ACL*, 2019.
[93] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
[94] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, 2018.
[95] 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.
[96] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. *TIP*, 2021.
[97] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. In *ECCV*, 2018.
[98] Yutaro Yamamuro, Kazuki Hanazawa, Masahiro Shida, Tsuyoshi Kodake, Shinji Takenaka, Yuji Sato, and Takeshi Fujimatsu. Submission to epic-kitchens action anticipation challenge 2021. In *CVPR Workshop*, 2021.
[99] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. In *arXiv preprint arXiv:2006.15489*, 2020.
[100] Olga Zatsarynna, Yazan Abu Farha, and Juergen Gall. Multi-modal temporal convolutional network for anticipating actions in egocentric videos. In *CVPR Workshop*, 2021.
[101] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021.