Anticipative Video Transformer @ EPIC-Kitchens Action Anticipation Challenge 2021

Rohit Girdhar[1] Kristen Grauman[1,2]
[1]Facebook AI Research [2]University of Texas, Austin
http://facebookresearch.github.io/AVT

Abstract

In this report, we describe an Anticipative Video Transformer (AVT) [11] based solution for the EPIC-Kitchens-100 anticipation challenge. AVT leverages a vision transformer based backbone architecture followed by causal attention based transformer decoder to model the sequential nature of videos. For the challenge, we aggregate predictions from multiple variants of AVT, applied to different input modalities and backbone architectures, along with prior work. Our final model obtains strong performance on the challenge test set with 16.5% mean top-5 recall in predicting future actions.

1. Introduction

Anticipating actions that a person might do in the future is an important task in egocentric computer vision. It forms the basis for various downstream applications on wearable devices, from safety systems that warn the user before potentially dangerous actions, to an assistive systems that help a user to perform actions by suggesting next steps. Compared to traditional action recognition, anticipation tends to be significantly more challenging. 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 [12], 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 [9, 18] finds that modeling long temporal context is often important for anticipation, unlike action recognition where frame-level modeling is often enough [14, 21].

To that end, there have been attempts to use sequential modeling architectures for action anticipation. While recurrent models like LSTMs have been explored for anticipation [1, 9, 23], 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 [18]. However, it still reduces the video to multiple aggregate representations and loses its sequential nature.

Hence, we introduce *Anticipative Video Transformer* (AVT), an alternate video modeling architecture that replaces "aggregation" based temporal modeling with a *anticipative* 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. Furthermore, while it is compatible with various backbone architectures, we leverage the recently proposed vision transformer based architectures [7] as the frame encoder, resulting in an end-to-end attention based architecture.

2. Our Approach

We now describe AVT briefly as illustrated in Figure 1, and refer the readers to the full paper [11] for details.

2.1. 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 [4, 8, 20, 21] and can be used with AVT as we demonstrate later, in this work we propose an alternate architec-

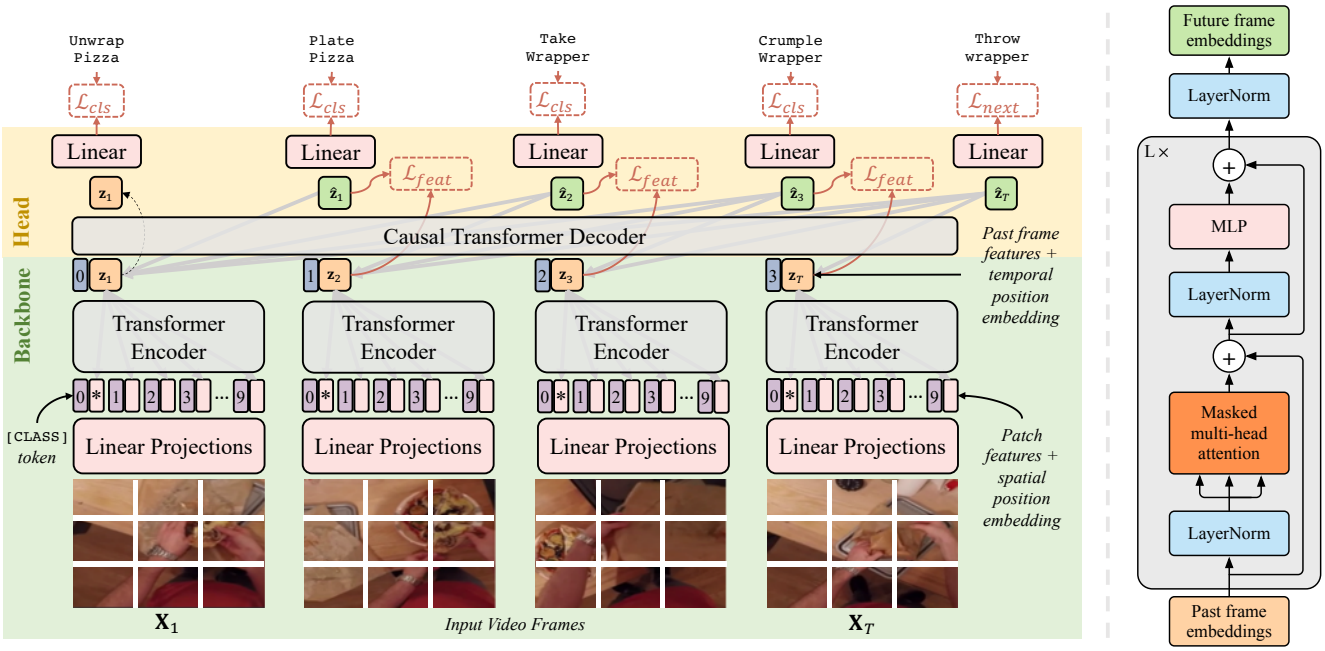

Figure 1: *(Left)* **AVT architecture.** We split the T input frames into non-overlapping patches that are linearly projected. We add a learned `[CLASS]` token, along with spatial position embeddings, and the resulting features are passed through multiple layers of multi-head attention, with shared weights 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 [22], causal masking in attention, and a final LayerNorm [16].

ture for video understanding based purely on attention. This backbone, which we refer to as AVT-b, adopts the recently proposed Vision Transformer (ViT) [7] architecture, which has shown impressive results for static image classification. Specifically, we adopt the ViT-B/16 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 [19, 21], 3D CNNs [4, 8, 20], or fixed feature representations based on detected objects [2, 3] or visual attributes [15]. In § 3 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.

2.2. 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 [16]. 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). The final output is then passed through another

LN, akin to GPT-2 [16], to obtain the future frame embeddings.

2.3. Training Details

The models are then trained with a combination of three objectives that include next action anticipation, future feature prediction, and current action classification. We refer the reader to the main paper [11] for details.

3. Experiments

3.1. Implementation Details

We preprocess the input video clips by randomly scaling the height between 248 and 280px, and take a 224px crops at training time. We sample 10 frames at 1FPS by default. We adopt network architecture details from [7] 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 pretrained 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 by default, with a 20 epoch warmup [13] and 30 epochs of cosine annealed decay. In all experiments, we train the model to predict the future actions, and verbs/nouns are inferred from the action prediction by marginalizing over the other. At test time, we employ 3-crop testing, where we compute three 224px spatial crops from 248px input frames, and average the predictions over the corresponding three clips.

The default backbone for AVT is AVT-b, based on the ViT-B/16 architecture. However, we also experiment with only our head model operating on fixed features from 1) a frame-level TSN [21] backbone pre-trained for action classification, or 2) a recent spatiotemporal convolutional architecture irCSN-152 [20] pre-trained on a large weakly labeled video dataset [10], 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. We use the validation set to optimize the hyperparameters for each setting, and use that setup on the held out test sets.

3.2. Ablations

In Table 1, we experimentally compare AVT to prior work and variants of itself with different backbones and modalities on the validation set. We find AVT-h over features from prior work [9] already outperforms prior work. We are able to further improve results with the AVT-b backbone and training jointly, especially with the IN21k initial-

	#	Head	Backbone	Init	Context	Verb	Noun	Action
RGB	1	RULSTM [5]	TSN	IN1k	2.8s	27.5	29.0	13.3
	2	AVT-h	TSN	IN1k	10s	27.2	30.7	13.6
	3	AVT-h	irCSN152	IG65M	10s	25.5	28.1	12.8
	4	AVT-h	AVT-b	IN1k	10s	28.2	29.3	13.4
	5	AVT-h	AVT-b	IN21+1k	10s	28.7	32.3	14.4
	6	AVT-h	AVT-b	IN21k	10s	**30.2**	31.7	14.9
	7	AVT-h	AVT-b	IN21k	15s	30.1	**33.8**	**15.7**
OBJ	8	RULSTM [5]	Faster R-CNN	IN1k	2.8s	17.9	23.3	7.8
	9	AVT-h	Faster R-CNN	IN1k	10s	**18.0**	**24.3**	**8.7**
Flow	10	RULSTM [5]	TSN	IN1k	2.8s	19.1	16.7	**7.2**
	11	AVT-h	TSN	IN1k	10s	**20.9**	**16.9**	6.6

Table 1: EK100 (val) using individual modalities. AVT outperforms prior work using the exact same features, and further improves with our AVT-b backbone. The 15s model (row 7) was also trained for longer (70 epochs as opposed to 50 default). Performance reported using overall class-mean recall@5.

Models fused	Weights	Action
2 + 9	1.5:0.5	14.8
6 + 9	2.5:0.5	15.9
1 + 6 + 9	1.0:1.0:0.5	16.9
1 + 2 + 3 + 6 + 9	1.0:1.0:1.0:1.0:0.5	18.2
1 + 2 + 3 + 6 + 7 + 9 + 11	1.0:1.0:1.0:0.5:1.5:0.5:0.5	**19.2**

Table 2: EK100 (val) late fusing predictions from different architectures. The numbers refer to the model in the corresponding row in Table 1. Performance reported using overall class-mean recall@5 for action prediction.

		Overall			Unseen Kitchen			Tail Classes		
Split	Method	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 [5]	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 [5]	25.3	26.7	11.2	19.4	26.9	9.7	17.6	16.0	7.9
	TBN [24]	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 [17]	21.8	30.6	12.6	17.9	27.0	10.5	13.6	20.6	8.9
	ICL+SJTU	**36.2**	32.2	13.4	**27.6**	24.2	10.1	**32.1**	**29.9**	11.9
	Panasonic	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 submission. We outperform prior work including all challenge submissions, with especially significant gains on tail classes. Performance reported using class-mean recall@5. AVT+ and AVT++ late fuse predictions from multiple modalities, please see text for details.

ization. Finally, by using additional frames of context and training for longer, we obtained the best RGB-only performance of 15.7%, showing AVT is effective in incorporating long-term context.

Next, to further improve the performance, we aggregate predictions across modalities and models by simple weighted averaging of L_2 normalized predictions, as shown in Table 2. The model numbers refer to the model in the corresponding row in Table 1. We find that combining multiple RGB models, based on fixed features and end-to-end trained, as well as ones using other architectures [9], and AVT-h applied on obj and flow features gave the best results on val set. We use a similar model on the test set as described next.

3.3. Final Model

For the test submission, we first train our models on the train+val set, and test those models as well as the models trained only on train set, on the test set. Then, we late fuse predictions using similar weights as the best combination in Table 2, and for each case where we use both train+val and train-only models, we use the same weight on predictions from both. Specifically, we use both train+val and train-only models for 2, 3, 6, 7 and 9; and train-only models for 1 and 11. This model obtains 16.53% mean top-5 recall for actions, as reported in our challenge submission on the leaderboard. We show the full comparison to existing state-of-the-art as well as challenge submissions in Table 3. Our RGB+Obj (6 + 9) late fused model is referred to as AVT+, and final late fused model is referred to as AVT++. It was submitted to the challenge using CodaLab username "shef" with team name "AVT-FB-UT".

In terms of the supervision scales [6], our pre-training scale is 2 since we use publicly available models pre-trained on public weakly supervised videos [10]. The full available supervision in Epic Kitchens is used for training, leading to supervision level of 4. The training data used is train + val sets, leading to training data scale of 4.

4. Conclusion

We have presented the Anticipative Video Transformer (AVT) architecture as used in the EPIC-Kitchens 2021 challenge. We propose a end-to-end Transformer based architecture for predictive video tasks such as anticipation, and show that it improves over prior work. Our best model, that aggregates predictions across modalities and models, obtains strong performance of 16.5% mean top-5 recall in predicting future actions on the test set.

References

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

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

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

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

[5] 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.

[6] Dima Damen and Michael Wray. Supervision levels scale (sls). *arXiv preprint arXiv:2008.09890*, 2020.

[7] 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.

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

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

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

[11] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. *arXiv preprint arXiv:2106.02036*, 2021.

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

[13] 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.

[14] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

[15] 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.

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

[17] Fadime Sener, Dibyadip Chatterjee, and Angela Yao. Technical report: Temporal aggregate representations. *arXiv preprint arXiv:2106.03152*, 2021.

[18] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *ECCV*, 2020.

[19] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.

[20] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolu-

tional networks. In *ICCV*, 2019.

[21] 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.

[22] 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.

[23] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. *TIP*, 2021.

[24] Olga Zatsarynna, Yazan Abu Farha, and Juergen Gall. Multi-modal temporal convolutional network for anticipating actions in egocentric videos. In *CVPR Workshop*, 2021.