Supplementary Material:
MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition

Chao-Yuan Wu [*,1] Yanghao Li [*,1] Karttikeya Mangalam [1,2]
Haoqi Fan [1] Bo Xiong [1] Jitendra Malik [1,2] Christoph Feichtenhofer [*,1]
[*]equal technical contribution
[1]Facebook AI Research [2]UC Berkeley

1. Architecture Specifications

The architecture design of MeMViT is based on MViTv2 [6, 11]. Table 1 presents the exact specification.

stage	operators	output sizes
data	stride $4\times1\times1$	$\mathbf{16}\times224\times224$
cube_1	$3\times7\times7$, 96 stride $2\times4\times4$	$96\times\mathbf{8}\times56\times56$
scale_2	$\begin{bmatrix}\text{MHPA}(96)\\ \text{MLP}(384)\end{bmatrix}\times\mathbf{1}$	$96\times\mathbf{8}\times56\times56$
scale_3	$\begin{bmatrix}\text{MHPA}(192)\\ \text{MLP}(768)\end{bmatrix}\times\mathbf{2}$	$192\times\mathbf{8}\times28\times28$
scale_4	$\begin{bmatrix}\text{MHPA}(384)\\ \text{MLP}(1536)\end{bmatrix}\times\mathbf{11}$	$384\times\mathbf{8}\times14\times14$
scale_5	$\begin{bmatrix}\text{MHPA}(768)\\ \text{MLP}(3072)\end{bmatrix}\times\mathbf{2}$	$768\times\mathbf{8}\times7\times7$

(a) **MeMViT-16, 16×4**

stage	operators	output sizes
data	stride $4\times1\times1$	$\mathbf{32}\times224\times224$
cube_1	$3\times7\times7$, 96 stride $2\times4\times4$	$96\times\mathbf{16}\times56\times56$
scale_2	$\begin{bmatrix}\text{MHPA}(96)\\ \text{MLP}(384)\end{bmatrix}\times\mathbf{2}$	$96\times\mathbf{16}\times56\times56$
scale_3	$\begin{bmatrix}\text{MHPA}(192)\\ \text{MLP}(768)\end{bmatrix}\times\mathbf{3}$	$192\times\mathbf{16}\times28\times28$
scale_4	$\begin{bmatrix}\text{MHPA}(384)\\ \text{MLP}(1536)\end{bmatrix}\times\mathbf{16}$	$384\times\mathbf{16}\times14\times14$
scale_5	$\begin{bmatrix}\text{MHPA}(768)\\ \text{MLP}(3072)\end{bmatrix}\times\mathbf{3}$	$768\times\mathbf{16}\times7\times7$

(b) **MeMViT-24, 32×3**

Table 1. **Architecture specification** for our "MeMViT-16, 16×4" (default) and "MeMViT-24, 32×3" models. Bold face highlights the difference between the two (*i.e.*, temporal resolution and depth). MHPA(c): Multi-Head Pooling Attention [6] with c channels. MLP(c'): MultiLayer Perceptron with c' channels.

Relative Positional Embeddings. As discussed in §4, it is important use relative positional embeddings instead of absolute positional embeddings as used in MViTv1 [6]. Our implementation is based on Shaw *et al.* [15], *i.e.*,[1]

$$\text{Attn}(Q,K,V) = \text{Softmax}\left((QK^\top + E^{(\text{rel})})/\sqrt{d}\right)V,$$

$$\text{where} \quad E^{(\text{rel})}_{ij} = Q_i \cdot R_{p(i),p(j)}. \tag{1}$$

$p(i)$ and $p(j)$ denote the spatiotemporal positions of tokens i (in queries) and j (in keys/values), respectively. In other words, we learn relative positional embeddings R that interact with queries Q depending on the relative positions between the queries and the keys/values. Note, however, that the number of possible embeddings grows in $\mathcal{O}(T\times H\times W)$, which is significantly more expensive than the one-dimensional case considered in Shaw *et al.* [15] for language modeling. We thus decompose the relative positional embeddings into

$$R_{p(i),p(j)} = R^{\text{t}}_{t(i),t(j)} + R^{\text{h}}_{h(i),h(j)} + R^{\text{w}}_{w(i),w(j)}, \tag{2}$$

where R^t, R^h, and R^w denote the relative positional embeddings along the temporal, frame hight, and frame width dimensions, respectively. $t(i)$, $h(i)$, $w(i)$ denote the temporal position, the vertical position, and the horizontal position of token i, respectively.

Compression Module Details. The compression module with a downsampling factor of $r_t \times r_h \times r_w$ is implemented as a learnable pooling (*i.e.*, depth-wise convolution) layer with a kernel size of $(2r_t{+}1)\times(2r_h{+}1)\times(2r_w{+}1)$ and a stride of $r_t \times r_h \times r_w$.

2. Kinetics Pre-training Details

To pre-train MeMViT on the Kinetics datasets [2, 3, 10] efficiently, we propose a progressive strategy. Namely, instead of training on full Kinetics videos throughout, we

[1]The only difference between our implementation and Shaw *et al.* [15] is that we do not add the additional embeddings on "values", as in preliminary experiments we did not find it to improve accuracy.

progressively increase the video length from one clip long (randomly sampled from full video) to the full video (10 seconds for Kinetics).[2] Intuitively, this strategy allows the model to see more diverse spatial patterns in earlier epochs for faster spatial pattern learning and gradually adapt to longer videos in later epochs. Concretely, we extend the original MViTv2 recipe (that trains on one-clip-long videos sampled from full videos) by a "second stage", which contains 40 epochs with 4 epochs of warm-up [9]. Within the 40 epochs, we train on videos that are 2-, 3-, 4-, and finally 5-clip-long for 10 epochs each. For data augmentation, we randomly drop $m \in [0, M-1]$ steps out of the M steps of memory tensors at each iteration of training. (At inference time, we still use all M steps of memory.) All other optimization hyperparameters follow the original MViTv2 recipe [11].

3. AVA Experiments

Person Detector. The person detector used in AVA experiments is a Faster R-CNN [14] with a ResNeXt-101-FPN [12, 19] backbone from Wu *et al.* [18]. The model obtains 93.9 AP@50 on the AVA validation set [18]. Please refer to the original paper [18] for details.

Output Head. Instead of using a linear output head for AVA, we additionally add a transformer layer (namely, an MViTv2 layer without pooling, since each token is already RoI-pooled) before the linear classifier. We find this to improve accuracy. Table 2 presents ablation results.

4. EPIC-Kitchens-100 Experiments

We train our EPIC-Kitchens models with AdamW [13] for 30 epochs using a base learning rate of 0.0002, a weight decay of 0.05, and a batch size of 128. Other training hyperparameters follow the Kinetics [10] recipe of MViTv2 [11]. We fine-tune action anticipation models from action classification models using the same training recipe.

For the anticipation task, we perform experiments on a *causal* version of MeMViT, to make sure our prediction does not depend on frames beyond the "observed video" [4, 5]. In particular, we 1) modify the learnable pooling so that it strictly pools only current or past contents, 2) mask attention so that it attends only current or past contents, 3) make the convolutions in the data layer 'causal', and 4) remove the global 'classification token'. Following common practice in the object detection community [16, 17], we use equalization loss [16] with threshold $\lambda = 0.003$ to address the class imbalance issue.

[2]When MeMViT operates on videos that are one-clip-long, it effectively falls back to a short-term MViTv2 (since there is no memory about the video cached from the previous step).

Our action classification model has two heads to predict verb and noun, respectively, following prior work [1, 18]. Our action anticipation model has only one head to predict the action directly and marginalize the output probabilities to obtain the verb and noun predictions, following standard practice [7, 8].

5. Supplementary Experiments

Model Detail Ablation. Table 2 presents additional ablation on our implementations choices.

	mAP
MViTv2-B, 16×4 [11] (abs. positional embedding)	24.5
+ relative positional embedding	25.4
+ pool first	25.5
+ test on full frame	26.6
+ attention head (our default baseline)	**27.0**

Table 2. **Detailed ablation on our default baseline model**.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *Proc. ICCV*, 2021. 2

[2] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about Kinetics-600. *arXiv:1808.01340*, 2018. 1

[3] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 1

[4] 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. 2

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *PAMI*, 2021. 2

[6] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proc. ICCV*, 2021. 1

[7] Antonino Furnari and Giovanni Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *PAMI*, 2020. 2

[8] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. In *Proc. ICCV*, 2021. 2

[9] 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:1706.02677*, 2017. 2

[10] 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:1705.06950*, 2017. 1, 2

[11] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021. 1, 2

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2017. 2

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[15] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL*, 2018. 1

[16] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proc. CVPR*, 2020. 2

[17] Jingru Tan, Gang Zhang, Hanming Deng, Changbao Wang, Lewei Lu, Quanquan Li, and Jifeng Dai. 1st place solution of lvis challenge 2020: A good box is not a guarantee of a good mask. *arXiv preprint arXiv:2009.01559*, 2020. 2

[18] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proc. CVPR*, 2019. 2

[19] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, 2017. 2