VideoMAE V2: Scaling Video Masked Autoencoders with Dual Masking Supplementary Material

Limin Wang[1,2,*] Bingkun Huang[1,2,*] Zhiyu Zhao[1,2] Zhan Tong[1]
Yinan He[2] Yi Wang[2] Yali Wang[3,2] Yu Qiao[2,3]
[1] State Key Laboratory for Novel Software Technology, Nanjing University, China
[2] Shanghai AI Lab, China [3] Shenzhen Institute of Advanced Technology, CAS, China

In this supplementary material, we provide more details of our VideoMAE V2 and present more experiment results. Specifically, we give a detailed description of the architectures of our VideoMAE V2 in Section A. Then, we present the details on building our pre-training datasets in Section B. After this, we provide more implementation details in our experiments in Section C. Finally, we give more results and analysis on our VideoMAE V2 in Section D.

A. Model Architecture

We build the encoder and decoder in our VideoMAE V2 by using the vanilla ViT backbone with joint space-time attention. To ensure efficient computation, our decoder does not get larger as the encoder scales up, but always stays at the size of 4 layers and 512 channels. We show the architectures of VideoMAE V2 in Tab. 1, taking ViT-giant as an example.

B. Datasets

B.1. UnlabeledHybrid

Our UnlabeledHybrid dataset is a hybrid dataset consisting of Kinetics [26], Something-Something [19], AVA [20], WebVid2M [5], and our self-collected Instagram dataset. When training VideoMAE V2, the sampling stride τ is 2 on Something-Something, and 4 on the other datasets. The detailed components of UnlabeledHybrid are shown in Tab. 2. We now specify the handling of each dataset.

Kinetics. Videos in Kinetics are from YouTube. We adapt the same method with [3] to make a mixed kinetics dataset. Kinetics has three versions, Kinetics-400/600/700, based on the number of human action categories. We merge the training set and validation set of the three versions, then remove the duplicated videos according to YouTube IDs, and finally delete the validation videos that existed in the training set. As some videos have different category names in different versions of Kinetics, we also group them together, resulting in a Kinetics dataset with 710 categories, termed Kinetics-710 (K710) or *LabeledHybrid*. K710 contains 658k training videos and 67k validation videos.

Something-Something. Videos in Something-Something are shot from video scripts, usually from a first-person perspective. We choose Something-Something V2 (SSV2) as the part of UnlabeledHbyrid dataset. SSV2 is a motion-centric dataset containing 169k training videos and 25k validation videos.

AVA. Videos in AVA are movie clips, ranging from the 15th to the 30th minute of each movie. We always cut the 15-minute movie clips from the AVA training set by 300 frames, resulting in 21k video clips.

WebVid2M. Videos in WebVid2M are scraped from the internet. We randomly pick 250k training videos from the original datasets.

Self-collected Instagram dataset. We used thousands of category tags from the already publicly available dataset as query phrases to scrape million of videos from Instagram. The average duration of the videos is 34 seconds. We also randomly pick 250k videos from the dataset.

B.2. LabledHybrid

We build the labeled datasets for our VideoMAE post-pre-training by taking the union of different versions of Kinetics dataset. The construction details is following the UniformerV2 [30] and more details could be referred to the original paper.

C. Implementation Details

In this section, we will describe the implementation details in the three stages of progressive training: *pre-training*, *post-pre-training*, and *specific fine-tuning*.

Stage	VideoMAE V2-giant	Output Size
Data	UnlabeldHybrid	$3 \times 16 \times 224 \times 224$
Cube	$2 \times 14 \times 14, 1408$ stride $2 \times 14 \times 14$	$1408 \times 8 \times 256$
Mask	tube masking mask ratio = ρ	$1408 \times 8 \times \lfloor 256 \times (1-\rho) \rfloor$
Encoder	$\begin{bmatrix} \text{MHA}(1408) \\ \text{MLP}(6144) \end{bmatrix} \times 40$	$1408 \times 8 \times \lfloor 256 \times (1-\rho) \rfloor$
Projector	MLP(512)	$512 \times 8 \times \lfloor 256 \times (1-\rho) \rfloor$
Decoder Mask	running cell masking decoder mask ratio = ρ^d concat unmasked learnable tokens	$512 \times 8 \times \left(\lfloor 256 \times (1-\rho) \rfloor + \lfloor 256 \times (1-\rho^d) \rfloor \right)$
Decoder	$\begin{bmatrix} \text{MHA}(512) \\ \text{MLP}(2048) \end{bmatrix} \times 4$	$512 \times 8 \times \left(\lfloor 256 \times (1-\rho) \rfloor + \lfloor 256 \times (1-\rho^d) \rfloor \right)$
Projector	discard visible tokens MLP(1176)	$1176 \times 8 \times \lfloor 256 \times (1-\rho^d) \rfloor$
Reshape	from 1176 to $3 \times 2 \times 14 \times 14$	$3 \times 16 \times \lfloor 224(1-\rho^d) \rfloor \times \lfloor 224(1-\rho^d) \rfloor$

Table 1. **Architectures details of VideoMAE V2-g**. The main difference between VideoMAE v2 and VideoMAE v1 is the dual masking design. VideoMAE v2 does not reconstruct the full video clip, while only calculates MSE loss on tokens that are invisible to the encoder.

Dataset	Size	Source
K710	658k	YouTube
SSV2	169k	Shot from Scripts
AVA	21k	Movie
WebVid2M	250k	Internet
self-collected	250k	Instagram
UnlabeledHybrid	1.348M	Multi-Source

Table 2. **Components of UnlabeledHybrid**. We build our unlabeled pre-train dataset by collecting clips from multiple resources to ensure the generalization ability of learned models by our VideoMAE V2.

C.1. Pre-training

We pre-train VideoMAE V2, both ViT-huge and ViT-giant, 1200 epochs on the UnlabeledHybrid dataset with 64 80G-A100 GPUs. Besides the dual masking core design of VideoMAE V2, we also adapt mix-precision training and checkpointing at the engineering level to speed up pre-training. To avoid the potential precision overflow risk during model pre-training, we train the encoder with FP-16 mixed precision and the decoder with FP32 precision. We adapt repeated augmentation to reduce the video loading overhead. The learning rate is scaled linearly according to the total batch size, *i.e.* $\text{lr} = \text{base\_lr} \times \text{batch\_size} \,/\, 256$. The detailed pre-training setting is shown in Tab. 4.

C.2. Post-pre-training

In the supervised *post-pre-training* stage, we fine-tune the pre-trained encoder on *LabeledHybrid* (K710). When training ViT-giant, we found that the dropout layer before the classification head has little positive effect on preventing overfitting, so the dropout layer was removed and the drop path rate was increased slightly. The clip grading stabilizes the optimization of large models in the early stages of fine-tuning to some extent, and it is advisable to adjust the value of the clip grading with the batch size changing. The choice of layer decay matters. A smaller layer decay better maintains the pre-training effect, but may not provide enough space for improvement in the later stages of fine-tuning. A relatively large layer decay is recommended when the model is well pre-trained, i.e. when it exhibits smaller gradients in the shallow layers and bigger gradients in the deep layers at the early stages of fine-tuning. The detailed settings are shown in Tab. 5. Notably, this setting also works for fine-tuning directly on the kinetics dataset.

C.3. Specific fine-tuning

After the post-pre-training stage, we perform the *specific fine-tuning* stage to get the specific models on action classification, action detection, and temporal action detection.

C.3.1 Action classification

We test the performance of the specific models for action classification on Kinetics [26], Something-Something [19], UCF101 [42] and HMDB51 [28] with regular 16×224^2

Config	Kinetics 16×224^2	Kinetics 64×266^2	Sth-Sth 16×224^2	UCF101 16×224^2	HMDB51 16×224^2
optimizer	AdamW				
base learning rate	1e-5	1e-4	3e-4	1e-3	5e-4
weight decay	0.05				
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$				
batch size	32	32	96	24	24
learning rate schedule	cosine decay				
warmup epoch	0	0	5	5	5
epoch	3	5	10	50	15
repeated augmentation	2				
RandAug	(0, 0.5)				
label smoothing	0.1				
mixup	0.8				
cutmix	1.0				
drop path	0.3	0.35	0.35	0.35	0.35
flip augmentation	yes	yes	no	yes	yes
augmentation	MultiScaleCrop				
dropout	0.5				
layer-wise lr decay	0.9				
clip grading	None				

Table 3. Action classification setting in specific fine-tuning stage.

Config	Value
mask ratio	0.9
decoder mask ratio	0.5
optimizer	AdamW [36]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
batch size	8192
learning rate schedule	cosine decay
warmup epoch	120
epoch	1200
repeated augmentation	4
flip augmentation	no
augmentation	MultiScaleCrop
clip grading	0.02

Table 4. **Pre-training setting**, where batch size includes the additional views produced by repeated augmentation and epochs refers to the total number of times the data is sampled.

Config	Value
optimizer	AdamW [36]
base learning rate	1e-3
weight decay	0.05 (H), 0.1 (g)
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$ [11]
batch size	128
learning rate schedule	cosine decay [35]
warmup epoch	5
epoch	40 (H), 35 (g)
repeated augmentation [23]	2
RandAug [13]	(0, 0.5)
label smoothing [43]	0.1
mixup [56]	0.8
cutmix [53]	1.0
drop path	0.2 (H), 0.3 (g)
flip augmentation	yes
augmentation	MultiScaleCrop
dropout	0.5 (H), None (g)
layer-wise lr decay [6]	0.8 (H), 0.9 (g)
clip grading	None (H), 5.0 (g)

Table 5. Post-pre-training setting.

inputs. Further, we also test the performance of the model on Kinetics [26] with larger input shapes 64×266^2. The detailed fine-tuning setting of VideoMAE V2-g can be seen in Tab. 3. At the specific fine-tuning stage, increasing the dropout and drop path can reduce the risk of overfitting to a certain extent, and the optimization of the model is more stable after the supervised post-pre-training, so clip grading is not necessary.

Config	AVA 2.2 AVA-Kinetics
optimizer	AdamW
base learning rate	3e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	128
learning rate schedule	cosine decay
warmup epoch	2
epoch	10
repeated augmentation	no
drop path	0.3
flip augmentation	yes
layer-wise lr decay	0.9
clip grading	None

Table 6. Hyper-parameter settings of action detection.

Method	**Modality**	**UCF101**	**HMDB51**
OPN [29]	V	59.6	23.8
VCOP [50]	V	72.4	30.9
SpeedNet [7]	V	81.1	48.8
VTHCL [52]	V	82.1	49.2
Pace [46]	V	77.1	36.6
MemDPC [21]	V	86.1	54.5
CoCLR [22]	V	87.9	54.6
RSPNet [12]	V	93.7	64.7
VideoMoCo [38]	V	78.7	49.2
Vi^{2}CLR [14]	V	89.1	55.7
CVRL [41]	V	94.4	70.6
CORP$_f$ [24]	V	93.5	68.0
ρSimCLR$_{\rho=2}$ [18]	V	88.9	N/A
ρSwAV$_{\rho=2}$ [18]	V	87.3	N/A
ρMoCo$_{\rho=2}$ [18]	V	91.0	N/A
ρBYOL$_{\rho=4}$ [18]	V	94.2	72.1
MIL-NCE [37]	V+T	91.3	61.0
MMV [1]	V+A+T	92.5	69.6
CPD [31]	V+T	92.8	63.8
ELO [40]	V+A	93.8	67.4
XDC [2]	V+A	94.2	67.1
GDT [39]	V+A	95.2	72.8
VideoMAE V1	V	96.1	73.3
VideoMAE V2	V	**99.6**	**88.1**

Table 7. **Comparison with the state-of-the-art methods on UCF101 and HMDB51.** 'V' refers to visual, 'A' is audio, 'T' is text narration. "N/A" indicates the numbers are not available.

C.3.2 Action detection

We follow the training pipeline of the original VideoMAE *i.e.* person detection + action classification. We adapt only two data augmentations, random scale cropping, and random horizontal flipping. When training, we use the ground-truth person boxes, while in testing, we use the person boxes detected by AIA [44]. More settings see in Tab. 6.

Method	Top 1	Top 5	Views	TFLOPs
I3D NL [48]	77.7	93.3	10×3	10.77
TDN [47]	79.4	94.4	10×3	5.94
SlowFast R101-NL [17]	79.8	93.9	10×3	7.02
TimeSformer-L [8]	80.7	94.7	1×3	7.14
MTV-B [51]	81.8	95.0	4×3	4.79
Video Swin [34]	83.1	95.9	4×3	7.25
MViT-B [15]	81.2	95.1	3×3	4.10
ViViT-L FE [4]	81.7	93.8	1×3	11.94
MViTv2-B [32]	82.9	95.7	1×5	1.13
MViTv2-L (312p) [32]	86.1	97.0	3×5	42.42
MaskFeat [49]	85.1	96.6	1×10	3.78
MaskFeat (352p) [49]	87.0	97.4	4×3	45.48
MAE-ST [16]	86.8	97.2	7×3	25.05
VideoMAE [45]	86.6	97.1	5×3	17.88
VideoMAE (320p) [45]	87.4	97.6	4×3	88.76
VideoMAE V2-H	88.6	97.9	5×3	17.88
VideoMAE V2-g	88.5	98.1	5×3	38.16
VideoMAE V2-H (64×288^2)	89.8	98.3	4×3	153.34
VideoMAE V2-g (64×266^2)	**90.0**	**98.4**	2×3	160.30
Methods using in-house labeled data				
CoVeR (JFT-3B) [54]	87.2	–	1×3	-
MTV-H (WTS) [51]	89.1	98.2	4×3	44.47
MTV-H (WTS 280^2) [51]	**89.9**	98.3	4×3	73.57

Table 8. **Results on the Kinetics-400 dataset.** We report the performance of our pre-trained model with larger input resolution and more frames.

Method	Top 1	Top 5	Views	TFLOPs
SlowFast R101-NL [17]	81.8	95.1	10×3	7.02
TimeSformer-L [8]	82.2	95.6	1×3	7.14
MTV-B [51]	83.6	96.1	4×3	4.79
MViT-B [15]	83.8	96.3	3×3	4.10
ViViT-L FE [4]	82.9	94.6	1×3	11.94
MViTv2-B [32]	85.5	97.2	1×5	1.03
MViTv2-L (352p) [32]	87.9	97.9	3×4	45.48
MaskFeat [49]	86.4	97.4	1×10	3.77
MaskFeat (312p) [49]	88.3	98.0	3×4	33.94
VideoMAE V2-H	88.3	98.1	5×3	17.88
VideoMAE V2-g	88.8	98.2	5×3	38.16
VideoMAE V2-H (32×384^2)	89.6	98.4	4×3	184.24
VideoMAE V2-g (64×266^2)	**89.9**	**98.5**	2×3	160.30
Methods using in-house labeled data				
CoVeR (JFT-3B) [54]	87.9	97.8	1×3	-
MTV-H (WTS) [51]	89.6	98.3	4×3	44.47
MTV-H (WTS 280^2) [51]	**90.3**	**98.5**	4×3	73.57

Table 9. **Results on the Kinetics-600 dataset.** We report the performance of our pre-trained model with larger input resolution and more frames.

Config	Value
optimizer	AdamW [36]
base learning rate	1e-3 (K710), 5e-4 (SSv2)
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$ [11]
batch size	1024 (K710), 512 (SSv2)
learning rate schedule	cosine decay [35]
warmup epoch	5
epoch	100
RandAug [13]	(0, 0.5)
mixup [56]	0.8
cutmix [53]	1.0
drop path	0.1
flip augmentation	yes
augmentation	MultiScaleCrop
dropout	None
layer-wise lr decay [6]	0.75
clip grading	1.0
temperature	3.0

Table 10. Knowledge distilling setting.

Model	K400	K600	SSv2
VideoMAE-B	81.5	N/A	70.8
VideoMAE V2-g	88.5	88.8	77.0
Distilled ViT-B	87.1	87.4	75.0

Table 11. The performance of distilled ViT-B models on the datasets of Kinetics400, Kinetics600, and Something-Something V2.

C.3.3 Temporal action detection

We take the model trained on the *LabeledHybrid* dataset as the backbone network and test its generalization performance on the temporal action detection task following the architecture of ActionFormer [55] on THUMOS14 [25] and FineAction [33]. When training, we use Adam [27] with warm-up and fix the maximum input sequence length. As for inference, we use Soft-NMS [10] on the result action candidates to remove the highly overlap proposals and obtain the final result.

D. More Results

More results and analysis. We report more result comparisons on Kinetics with the larger input size in Tab. 8 and Tab. 9. We also add the results on UCF101 [42] and HMDB51 [28] in Tab. 7. From these results, we see that our model can further improve the recognition results by using larger input. Meanwhile, on the smaller benchmarks of UCF101 and HMDB51, our model obtains state-of-the-art performance, which is much better than the VideoMAE V1.

Distillation results. Using the procedure of [9], we are able to compress VideoMAE V2-g into a mush smaller ViT-B. Specifically, we initialize the student model with the VideoMAE V2-B weights after the post-pre-training. Then, we conduct the distillation on K710 (or SSv2) dataset for 100 epochs, with the goal of minimizing the KL divergence between the student model's logits and those of the teacher model. Detailed settings see in Tab. 10. Our evaluation of the distilled ViT-B model is based on its performance on the K400, K600, and SSv2, as shown in Tab. 11. From these results, we see that our distilled ViT-B model achieves much better performance than the original VideoMAE ViT-B models. We hope our distilled ViT-B model can serve as an efficient foundation model for downstream tasks.

References

[1] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelovic, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. 2020. 4

[2] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. 2020. 4

[3] Anonymous. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. under review. 1

[4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6816–6826, 2021. 4

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pages 1708–1718, 2021. 1

[6] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: BERT pre-training of image transformers. In *ICLR*, 2022. 3, 5

[7] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. 2020. 4

[8] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang, editors, *ICML*, pages 813–824, 2021. 4

[9] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10925–10934, 2022. 5

[10] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 5

[11] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, pages 1691–1703, 2020. 3, 5

[12] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. Rspnet: Relative speed perception for unsupervised video representation learning. 2021. 4

[13] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 3, 5

[14] Ali Diba, Vivek Sharma, Reza Safdari, Dariush Lotfi, Saquib Sarfraz, Rainer Stiefelhagen, and Luc Van Gool. Vi2clr: Video and image for visual contrastive learning of representation. 2021. 4

[15] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6804–6815, 2021. 4

[16] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *NeurIPS*, 2022. 4

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6201–6210, 2019. 4

[18] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. 2021. 4

[19] 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*, pages 5843–5851, 2017. 1, 2

[20] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, pages 6047–6056, 2018. 1

[21] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. 2020. 4

[22] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. 2020. 4

[23] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138, 2020. 3

[24] Kai Hu, Jie Shao, Yuan Liu, Bhiksha Raj, Marios Savvides, and Zhiqiang Shen. Contrast and order representations for video self-supervised learning. 2021. 4

[25] Haroon Idrees, Amir Roshan Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Comput. Vis. Image Underst.*, 155:1–23, 2017. 5

[26] Will Kay, João 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. *CoRR*, abs/1705.06950, 2017. 1, 2, 3

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[28] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 2, 5

[29] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequence. 2017. 4

[30] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *CoRR*, abs/2211.09552, 2022. 1

[31] Tianhao Li and Limin Wang. Learning spatiotemporal features via video and text pair discrimination. *arXiv preprint arXiv:2001.05691*, 2020. 4

[32] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, pages 4794–4804, 2022. 4

[33] Yi Liu, Limin Wang, Yali Wang, Xiao Ma, and Yu Qiao. Fineaction: A fine-grained video dataset for temporal action localization. *IEEE Trans. Image Process.*, 31:6937–6950, 2022. 5

[34] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3192–3201, 2022. 4

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 3, 5

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3, 5

[37] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. 2020. 4

[38] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. 2021. 4

[39] Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. 2021. 4

[40] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. 2020. 4

[41] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. 2021. 4

[42] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 5

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3

[44] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *European Conference on Computer Vision*, pages 71–87. Springer, 2020. 4

[45] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 4

[46] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. 2020. 4

[47] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. TDN: temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021. 4

[48] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 4

[49] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan L. Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, pages 14648–14658, 2022. 4

[50] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. 2019. 4

[51] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, pages 3323–3333, 2022. 4

[52] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020. 4

[53] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 3, 5

[54] Bowen Zhang, Jiahui Yu, Christopher Fifty, Wei Han, Andrew M. Dai, Ruoming Pang, and Fei Sha. Co-training transformer with videos and images improves action recognition. *CoRR*, abs/2112.07175, 2021. 4

[55] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *ECCV*, pages 492–510, 2022. 5

[56] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3, 5