A Implementation Details

Kinetics action classification. Our settings mainly follow [31, 77]. Table 5a summarizes our pre-training settings on Kinetics. Table 5b shows the corresponding fine-tuning settings for ViT-B/L/H. For fine-tuning, we add a linear classifier layer to the encoder's averaged tokens [18].

For fine-tuning the intermediately fine-tuned checkpoints from K600 in Table 7, we use the setting in Table 5b with a lower learning rate (8e-4) and shorter duration (40 epochs for ViT-L; 30 for ViT-H) and an increased drop path rate of 0.3 for ViT-H.

AVA action detection. Table 6a summarizes our fine-tuning settings on AVA [29]. The settings mainly follow [39, 77]. We follow the detection architecture in [21, 39, 77] that adapts Faster R-CNN [57] for video action detection. Only for the AVA results in Table 8, we use relative positions [59, 54] (as implemented in [39]) during fine-tuning.

SSv2 action classification. Table 6b summarizes our fine-tuning settings on SSv2 [27]. The settings mainly follow [39, 77]. For the frame sampling, we split each video into segments, and sample one frame from each segment to form a clip following [39, 19].

Fine-tuning from image pre-training. In Table 3 we have compared with ImageNet-based supervised/MAE pre-training. When fine-tuning these variants for videos, we inflate the 2D kernel of the patch embedding layer to 3D [8] and initialize the temporal position embeddings by zero.

config	value
optimizer	AdamW [43]
optimizer momentum	β_1, β_2=0.9, 0.95 [9]
weight decay	0.05
learning rate	1.6e-3
learning rate schedule	cosine decay [42]
warmup epochs [26]	120
epochs	default 800
repeated sampling [33]	4
augmentation	hflip, crop [0.5, 1]
batch size	512
gradient clipping	0.02

(a) Kinetics pre-training

config	ViT-B	ViT-L	ViT-H
optimizer	AdamW [43]		
optimizer momentum	β_1, β_2=0.9, 0.999		
weight decay	0.05		
learning rate	1.6e-2	4.8e-3	1.6e-3
learning rate schedule	cosine decay [42]		
warmup epochs [26]	5		
epochs	150	100	75
repeated sampling [33]	2	2	1
augmentation	RandAug (9, 0.5) [12]		
batch size	768	256	256
mixup [86]	0.8		
cutmix [84]	1.0		
label smoothing [64]	0.1		
drop path [34]	0.1	0.2	0.2
dropout [60]	0.3	0.3	0.5
layer-wise decay [11]	0.65	0.75	0.8

(b) Kinetics fine-tuning

Table 5: Settings on Kinetics.

config	values
optimizer	SGD
weight decay	1e-8
learning rate	7.2(L), 4.8(H)
learning rate schedule	cosine decay [42]
warmup epochs [26]	5
epochs	30
batch size	128
drop path [34]	0.2
dropout [60]	0.5
layer-wise decay [11]	0.75 (L) 0.85 (H)

(a) AVA fine-tuning

config	values
optimizer	SGD
weight decay	1e-4
learning rate	0.64 (L) 0.32 (H)
learning rate schedule	cosine decay [42]
warmup epochs [26]	3
epochs	40
augmentation	RandAug (9, 0.5) [12]
batch size	256
mixup [86]	0.8
cutmix [84]	1.0
label smoothing [64]	0.1
drop path [34]	0.2
dropout [60]	0.5
layer-wise decay [11]	0.75 (L) 0.85 (H)

(b) SSv2 fine-tuning

Table 6: Settings on AVA and SSv2. (L) and (H) stands for ViT-L and ViT-H, respectively.

pre-train	extra data	architecture	input size	top-1	top-5	FLOPs	param.
scratch		SlowFast [21]	64×224^2	79.8	93.9	$234 \times 3 \times 10$	60
scratch		X3D-XL [20]	16×312^2	79.1	93.9	$48 \times 3 \times 10$	11
scratch		MoViNet [36]	120×320^2	81.5	95.3	$386 \times 1 \times 1$	31
scratch		MViT-B [19]	64×224^2	81.2	95.1	$455 \times 3 \times 3$	37
scratch		MViTv2-B [19]	32×224^2	82.9	95.7	$255 \times 1 \times 5$	51
supervised	IN21K	Swin-B [41]	32×224^2	82.7	95.5	$282 \times 3 \times 4$	88
supervised	IN21K	Swin-L [41]	32×224^2	83.1	95.9	$604 \times 3 \times 4$	197
supervised	IN21K	Swin-L [41]	32×384^2	84.9	96.7	$2107 \times 5 \times 10$	200
BEVT [73]	IN1K+DALLE	Swin-B [41]	32×224^2	81.1	n/a	$282 \times 3 \times 4$	88
MaskFeat [77]		MViTv2-L [39]	16×224^2	84.3	96.3	$377 \times 1 \times 10$	218
MaskFeat [77]		MViTv2-L [39]	40×352^2	86.7	97.3	$3790 \times 3 \times 4$	218
MaskFeat [77]	K600	MViTv2-L [39]	40×352^2	87.0	97.4	$3790 \times 3 \times 4$	218
MAE		ViT-B	16×224^2	81.3	94.9	$180 \times 3 \times 7$	87
MAE		ViT-L	16×224^2	84.8	96.2	$598 \times 3 \times 7$	304
MAE		ViT-H	16×224^2	85.1	96.6	$1193 \times 3 \times 7$	632
MAE		ViT-L	40×312^2	85.8	96.9	$4757 \times 3 \times 7$	304
MAE		ViT-H	32×312^2	86.0	97.0	$6382 \times 3 \times 7$	632
MAE	K600	ViT-L	16×224^2	86.5	**97.2**	$598 \times 3 \times 7$	304
MAE	K600	ViT-H	16×224^2	**86.8**	**97.2**	$1193 \times 3 \times 7$	632
using in-house data for supervision:							
supervised	JFT-300M	ViViT-L [2]	32×320^2	83.5	94.3	$3980 \times 3 \times 1$	308
supervised	JFT-300M	ViViT-H [2]	32×320^2	84.9	95.8	$3981 \times 3 \times 4$	654
supervised + text	FLD-900M	Florence [83]	$n/a \times 384^2$	86.5	97.3	$n/a \times 3 \times 4$	647
SimMIM [80] + sup.	IN21K+70M	SwinV2-G [40]	8×384^2	86.8	n/a	$n/a \times 5 \times 4$	3000
supervised	JFT-3B+SSv2+MiT+IN	CoVeR [85]	16×448^2	87.2	n/a	$n/a \times 3 \times 1$	n/a
supervised	WTS-60M	MTV-H [82]	32×280^2	89.9	98.3	$6130 \times 3 \times 4$	n/a

Table 7: **System-level comparisons on Kinetics-400 action classification**. We report top-1 and top-5 accuracy on the validation set. The input size is $T \times H \times W$. FLOPs (in 10^9) are presented as "FLOPs per view $\times$ spatial views $\times$ temporal views", following the literature. Parameters are in 10^6. The "extra data" column specifies the data used in addition to K400. Entries using spatial resolution $>224^2$ are noted in gray; entries using in-house data for supervision are in light blue. Our results with K600 are with intermediate fine-tuning.

**This table does not include results using K700, because the K700 training set has 13.9k videos duplicated with the K400 validation set (19.9k). Results with K700 are in Table 8 (AVA) and Table 9 (SSv2).*

B Additional Experimental Results

B.1 System-level Comparisons

Kinetics-400. Table 7 compares on Kinetics-400 (K400). Our results are competitive with the leading ones. Importantly, our method is much *simpler* than many other entries. Our method is the only leading entry based on *vanilla* ViT, while others were based on hierarchical or specialized designs for videos. Our model does *not* use relative position embedding, which could have extra gains that are orthogonal to our thesis. Our results can compete with some strong results that were based on in-house data for supervision. Our models achieve this at standard 224×224 spatial resolution, while higher-resolution fine-tuning and testing may improve results at a higher cost, as shown in gray indicating entries using spatial resolution $>224^2$.

AVA. Table 8 compares on AVA [29] action detection. Using only a resolution of 16×224^2, our results are close to those of MaskFeat on higher-resolution inputs (40×312^2). Importantly, our architectures are plain ViT models without feature hierarchies, yet they perform strongly on this detection task.

SSv2. Table 9 compares on SSv2 [27] action classification. On the resolution of 16×224^2 and using vanilla ViT, our results compare favorably with those of MaskFeat on 40×312^2 inputs.

pre-train	pre-train data	architecture	input size	mAP center	mAP full	FLOPs	param.
supervised	K400	SlowFast [21]	32×224^2	23.8	-	138	53
supervised	K400	MViTv1-B [19]	64×224^2	27.3	-	455	36
supervised	K400	MViTv2-B [39]	32×224^2	28.1	29.0	225	51
MaskFeat [77]	K400	MViTv2-L [39]	40×312^2	**36.3**	**37.5**	2828	218
MAE	K400	ViT-L	16×224^2	35.9	36.8	598	304
MAE	K400	ViT-H	16×224^2	**36.8**	**37.4**	1193	632

(a) AVA results using Kinetics-400 pre-training

pre-train	pre-train data	architecture	input size	mAP center	mAP full	FLOPs	param.
supervised	K600	SlowFast [21]	64×224^2	27.5	-	296	59
supervised	K600	X3D-XL [20]	16×312^2	27.4	-	48	11
supervised	K600	MViT-B [19]	32×224^2	28.7	-	236	53
supervised	K600	MViTv2-B [39]	32×224^2	29.9	30.5	225	51
supervised	K600	ACAR [48]	64×224^2	-	31.4	n/a	n/a
MaskFeat [77]	K600	MViTv2-L [39]	40×312^2	**37.8**	**38.8**	2828	218
MAE	K600	ViT-L	16×224^2	37.7	38.4	598	304
MAE	K600	ViT-H	16×224^2	**39.2**	**40.3**	1193	632

(b) AVA results using Kinetics-600 pre-training

pre-train	pre-train data	architecture	input size	mAP center	mAP full	FLOPs	param.
supervised	K700	MViTv2-B [39]	32×224^2	31.3	32.3	225	51
supervised	K700	ACAR [48]	64×224^2	-	33.3	n/a	n/a
supervised	K700 + IN21K	MViTv2-L [39]	40×312^2	33.5	34.4	2828	213
MAE	K700	ViT-L	16×224^2	38.4	39.5	598	304
MAE	K700	ViT-H	16×224^2	**39.3**	**40.1**	1193	632

(c) AVA results using Kinetics-700 pre-training

Table 8: **System-level comparisons on AVA v2.2 action detection**. We report mAP using center-crop or full-resolution inference, following the literature. FLOPs (in 10^9) are measured with center-crop inference. Parameter numbers are in 10^6. Only in this table, following MaskFeat [77], our results are with intermediate fine-tuning and with relative positions [59, 54] during fine-tuning.

pre-train	pre-train data	architecture	input size	top-1	top-5	FLOPs	param.
supervised	K400	SlowFast [21]	32×224^2	63.1	87.6	$106 \times 3 \times 1$	53
supervised	K400	MViTv1-B [19]	64×224^2	67.7	90.9	$454 \times 3 \times 1$	37
supervised	K400	MViTv2-B [39]	32×224^2	70.5	92.7	$225 \times 3 \times 1$	51
supervised	K400 + IN21K	Swin-B [41]	32×224^2	69.6	92.7	$321 \times 3 \times 1$	89
supervised	K400 + IN21K	MViTv2-B [39]	32×224^2	72.1	93.4	$225 \times 3 \times 1$	51
supervised	K400 + IN21K	MViTv2-L [39]	40×224^2	73.3	94.1	$2828 \times 3 \times 1$	213
BEVT [73]	K400 + IN1K	Swin-B [41]	32×224^2	71.4	n/a	$321 \times 3 \times 1$	88
MaskFeat [77]	K400	MViTv2-L [39]	40×312^2	**74.4**	**94.6**	$2828 \times 3 \times 1$	218
MAE	K400	ViT-L	16×224^2	72.1	93.9	$598 \times 3 \times 1$	304
MAE	K400	ViT-H	16×224^2	**74.1**	**94.5**	$1193 \times 3 \times 1$	632

(a) SSv2 results using Kinetics-400 pre-training

pre-train	pre-train data	architecture	input size	top-1	top-5	FLOPs	param.
supervised	K600	MViTv1-B [19]	32×224^2	67.7	90.9	$454 \times 3 \times 1$	37
MaskFeat [77]	K600	MViTv2-L [39]	40×312^2	**75.0**	**95.0**	$2828 \times 3 \times 1$	218
MAE	K600	ViT-L	16×224^2	73.0	94.2	$598 \times 3 \times 1$	304
MAE	K600	ViT-H	16×224^2	**75.2**	**94.9**	$1193 \times 3 \times 1$	632

(b) SSv2 results using Kinetics-600 pre-training

pre-train	pre-train data	architecture	input size	top-1	top-5	FLOPs	param.
MAE	K700	ViT-L	16×224^2	73.6	94.4	$598 \times 3 \times 1$	304
MAE	K700	ViT-H	16×224^2	**75.5**	**95.0**	$1193 \times 3 \times 1$	632

(c) SSv2 results using Kinetics-700 pre-training

Table 9: **System-level comparisons on SSv2 action classification**. Notations of FLOPs (10^9) and parameters (10^6) follow Table 7. We do not use intermediate fine-tuning here (see Table 10).

B.2 Ablation on Intermediate Fine-tuning

In Table 3 we have shown results of self-supervised pre-training directly transferred to downstream datasets. Following the literature, we also investigate an another scenario: after self-supervised pre-training, we perform *intermediate fine-tuning* on the pre-training set using labels, before transferring. Table 10 studies its influence. Intermediate fine-tuning has substantial improvements on AVA, while on SSV2 its effect is marginal.

pre-train data	#	intermediate FT	K400	AVA	SSv2
K400	240k		84.8	32.3	72.1
K400	240k	✓	-	36.8	72.6
K600	387k		84.9	33.7	73.0
K600	387k	✓	86.5	37.9	73.1
K700	537k		n/a	34.2	73.6
K700	537k	✓	n/a	39.3	73.7

Table 10: **Influence of intermediate fine-tuning**, evaluated on AVA and SSv2. The model is ViT-L. The MAE pre-training length is 1600 epochs on K400/600/700. Using K700 training set for K400 validation is not legitimate due to the duplications in these training and validation sets.

B.3 Masking during fine-tuning

We perform an ablation that applies masking during the supervised fine-tuning phase. We explore a masking ratio of 50% that is annealed to 0% with a cosine schedule during fine-tuning. The result is 84.1%, comparred to 84.4% for full fine-tuning without masking, but at a 1.2× speedup. If we start fine-tuning with a masking ratio of 50% and anneal it to 0%, the accuracy is 83.8% at a speedup of 1.3×. The experiments are summarized in Table 11. We think this is an interesting result showing that masking can also speedup fine-tuning.

start fine-tune masking ratio	K400 accuracy	speed
0%	84.4	1.0×
50%	84.1	1.2×
75%	83.8	1.3×

Table 11: **Masking during fine-tuning** on Kinetics-400. We use Cosine annealing of masking ratio during fine-tuning. The starting masking ratio is varied between 0% (baseline without masking), 50% and 75%. The annealing is towards 0% at the end of fine-tuning. The model is ViT-L and the MAE pre-training length is 800 epochs on K400; *cf*. Table 2.

B.4 Ablation on SSv2

We perform a subset of the ablations that were carried out for Kinetics in Table 2 on the SSv2 dataset. We directly pre-train and fine-tune on SSv2 and use a short pre-training schedule of 200 epochs to save training resources. The results in Table 12 indicate that the default choices for Kinetics also lead to good performance on SSv2. Namely, spacetime agnostic mask sampling (Table 12a) as well as decoder width (12b) of 512 and depth (12c) of 4 provide better accuracy than other design choices.

case	ratio	acc.
agnostic	90	**63.4**
space-only	90	59.5
time-only	75	61.9

(a) **Mask sampling**. See also Fig. 4 and Table 2. Random sampling that is spacetime-*agnostic* works best.

dim	acc.
128	59.4
256	63.2
512	**63.4**

(b) **Decoder width**. Similar to Table 2, a narrow decoder (128-d) drops accuracy.

blocks	acc.
1	63.9
2	**63.4**
4	**63.4**
8	62.0

(c) **Decoder depth**. Four or two decoder layers provides good accuracy on SSv2.

Table 12: **Ablation experiments** on SSv2. We use a short pre-training length of 200 epochs. The model is ViT-L, with an input size of 16×224×224 and a spacetime patch size of 2×16×16. This table format follows [31] and Table 2. The entries marked in gray are the same, which specify the default settings, and achieve best performance (similar to the results for Kinetics in Table 2).

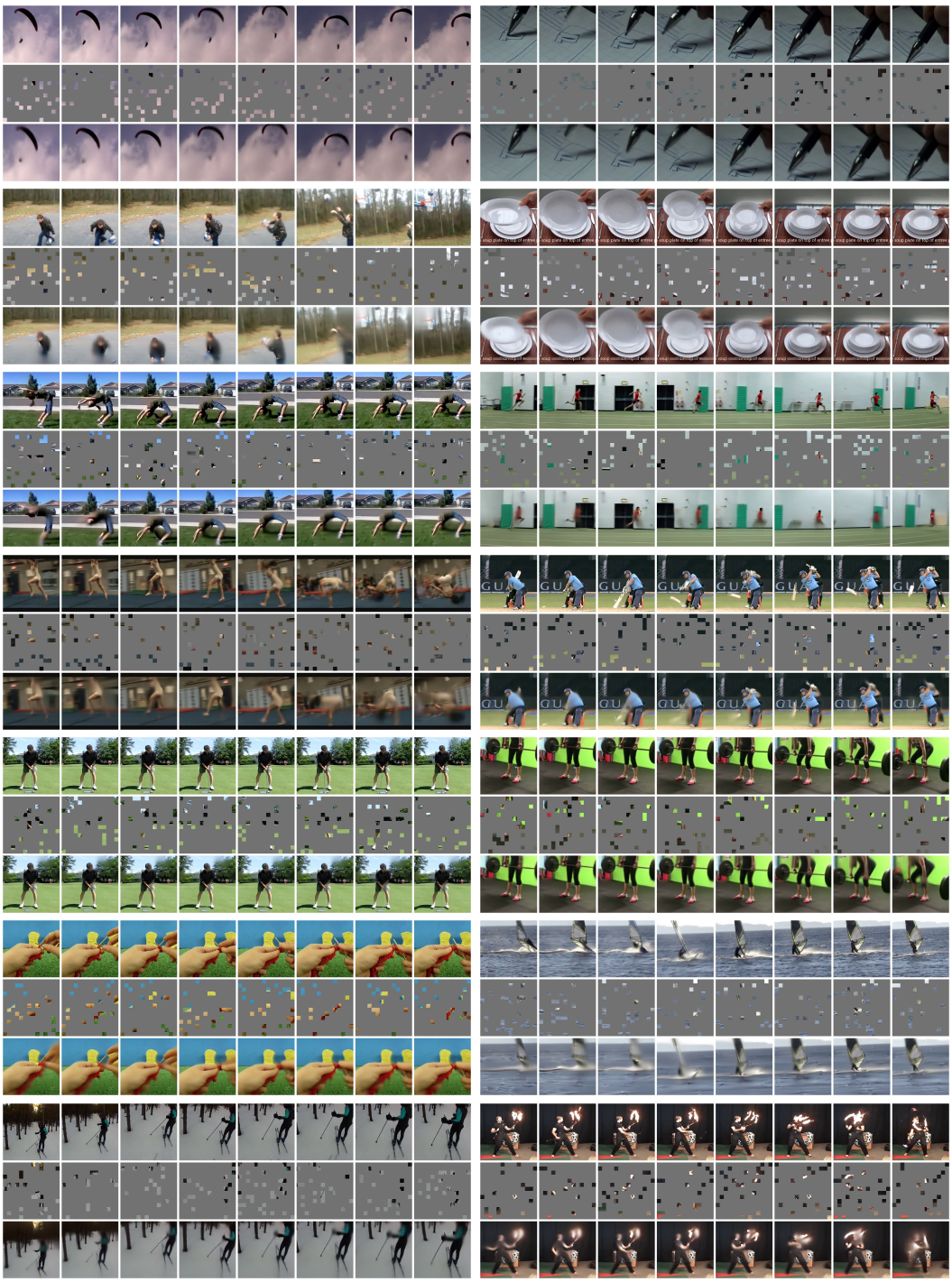

Figure 7: More visualizations on Kinetics-400 following Fig. 2 (masking ratio 90%).

Acknowledgements

We would like to thank Chen Wei, Karttikeya Mangalam, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Jitendra Malik for discussions and feedback.

C Potential negative societal impacts

The potential negative societal impact of our approach is related to other machine learning methods that are learning from training data and therefore reflect statistics and biases of the datasets used. Our method could also be used to generate content that may or may not reflect biases of the training data.

Video understanding research requires compute-intensive experiments because of the large input dimensionality and the large training datasets which can have negative environmental impact. To mitigate this effect, our approach greatly reduces the computational cost for performing research. Further, there is potential of misuse for video classification methods, *e.g*. for surveillance purpose.

References

[1] Pulkit Agrawal, João Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015.

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

[3] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image Transformers. *arXiv:2106.08254*, 2021.

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

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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.

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

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

[8] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[9] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[11] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[12] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020.

[13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *NAACL*, 2019.

[16] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. DynamoNet: Dynamic Action and Motion Network. In *ICCV*, 2019.

[17] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. PeCo: Perceptual codebook for BERT pre-training of Vision Transformers. *arXiv:2111.12710*, 2021.

[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. X3D: Expanding architectures for efficient video recognition. In *CVPR*, 2020.
[21] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, 2019.
[22] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *CVPR*, 2021.
[23] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *ICCV*, 2017.
[24] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.
[25] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *ICCV*, 2015.
[26] 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.
[27] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.
[28] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *NeurIPS*, 2020.
[29] Chunhui Gu, Chen Sun, Sudheendra Vijayanarasimhan, Caroline Pantofaru, David A. Ross, George Toderici, Yeqing Li, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.
[30] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Workshop on Large Scale Holistic Video Understanding, ICCV*, 2019.
[31] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.
[32] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.
[33] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *CVPR*, 2020.
[34] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.
[35] Will Kay, João 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.
[36] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. MoviNets: Mobile video networks for efficient video recognition. In *CVPR*, 2021.
[37] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989.
[38] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequence. In *ICCV*, 2017.
[39] 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:2112.01526*, 2021.
[40] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin Transformer v2: Scaling up capacity and resolution. *arXiv:2111.09883*, 2021.
[41] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin Transformer. *arXiv:2106.13230*, 2021.
[42] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[43] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.
[44] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017.
[45] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.
[46] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016.
[47] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.
[48] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *CVPR*, 2021.
[49] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, 2017.
[50] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.
[51] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021.
[52] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.
[53] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
[54] 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.
[55] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.
[56] Adria Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Pătrăucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. In *ICCV*, 2021.
[57] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
[58] Pierre Sermanet et al. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018.
[59] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv:1803.02155*, 2018.
[60] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 2014.
[61] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015.
[62] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv:1906.05743*, 2019.
[63] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.
[64] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.
[65] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. VIMPAC: Video pre-training via masked token prediction and contrastive learning. *arXiv:2106.11250*, 2021.
[66] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv:2203.12602*, 2022.
[67] 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.
[68] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.
[69] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010.

[70] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabelled video. In *CVPR*, 2016.

[71] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018.

[72] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016.

[73] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. BEVT: BERT pretraining of video transformers. In *CVPR*, 2022.

[74] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[75] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[76] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[77] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *arXiv:2112.09133*, 2021.

[78] Donglai Wei, Joseph J. Lim, Andrew Zisserman, and William T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018.

[79] Laurenz Wiskott and Terrence Sejnowski. Slow feature analysis: Unsupervised learning of invariances. In *Neural Computation*, 2002.

[80] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A simple framework for masked image modeling. *arXiv:2111.09886*, 2021.

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

[82] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. *arXiv:2201.04288*, 2022.

[83] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv:2111.11432*, 2021.

[84] 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 *ICCV*, 2019.

[85] 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. *arXiv:2112.07175*, 2021.

[86] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.