VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

Pre-training video transformers on extra large-scale datasets is generally required to achieve premier performance on relatively small datasets. In this paper, we show that video masked autoencoders (VideoMAE) are data-efficient learners for self-supervised video pre-training (SSVP). We are inspired by the recent ImageMAE [22] and propose customized video tube masking with an extremely high ratio. This simple design makes video reconstruction a more challenging and meaningful self-supervision task, thus encouraging extracting more effective video representations during this pre-training process. We obtain three important findings on SSVP: (1) An extremely high proportion of masking ratio (i.e., 90% to 95%) still yields favorable performance of VideoMAE. The temporally redundant video content enables higher masking ratio than that of images. (2) VideoMAE achieves impressive results on very small datasets (i.e., around 3k-4k videos) without using any extra data. This is partially ascribed to the challenging task of video reconstruction to enforce high-level structure learning. (3) VideoMAE shows that data quality is more important than data quantity for SSVP. Domain shift between pre-training and target datasets is an important issue. Notably, our VideoMAE with the vanilla ViT backbone can achieve 84.7% on Kinects-400, 75.3% on Something-Something V2, 90.8% on UCF101, and 61.1% on HMDB51, without using any extra data.

1 Introduction

Transformer [45] has brought significant progress in natural language processing [12, 5, 34]. The vision transformer [14] also improves a series of computer vision tasks including image classification [42, 59], object detection [6, 25], semantic segmentation [54], object tracking [11], and video recognition [4, 1]. The multi-head self-attention upon linearly projected image/video tokens is capable of modeling global dependency among visual content either spatially or temporally. The inductive bias is effectively reduced via this flexible attention mechanism.

Training effective vision transformers (ViTs) typically necessitates large-scale supervised datasets. Initially, the pre-trained ViTs achieve favorable performance by using hundreds of millions of labeled images [14]. For video transformers [1, 4], they are usually derived from image-based transformers and heavily depend on the pre-trained models from large-scale image data (e.g., ImageNet [37]). Previous trials [1, 4] on training video transformers from scratch yield unsatisfied results (except for MViT [15] with a strong inductive bias). Therefore, the learned video transformers are naturally biased by image-based models, and it still remains a challenge that *how to effectively and efficiently train a vanilla vision transformer on the video dataset itself without using any pre-trained model or extra image data.* Moreover, the existing video datasets are relatively small compared with image datasets, which further increases the difficulty of training video transformers from scratch. Meanwhile, self-supervised learning has shown remarkable performance by using large-scale image

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

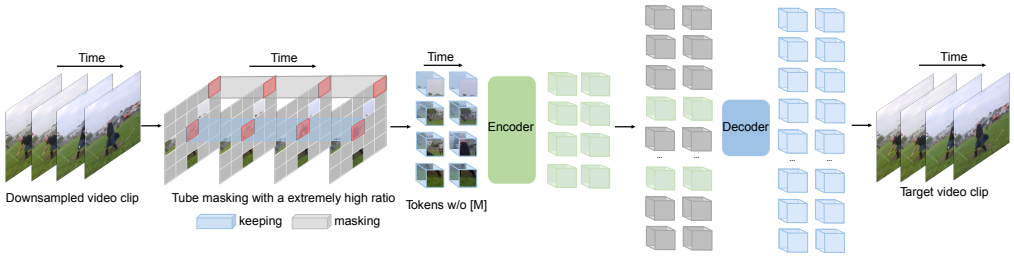


Figure 1: **Our VideoMAE Architecture**. Our VideoMAE performs the task of masking random cubes and reconstructing the missing ones with an asymmetric encoder-decoder architecture. Due to high redundancy and temporal correlation in videos, we present the customized design of tube masking with an extremely high ratio (90% to 95%). This simple design enables us to create a more challenging and meaningful self-supervised task to make the learned representations capture more useful spatiotemporal structures.

datasets [10, 7]. The learned representations have outperformed the ones via supervised learning when being transferred to downstream tasks. It is expected that this self-supervised learning paradigm can provide a promising solution to address the challenge of training video transformers.

Following the success of masked autoencoding in NLP [12] and images [22, 2], we present a new self-supervised video pre-training (SSVP) method, termed as *Video Masked Autoencoder* (VideoMAE). Our VideoMAE inherits the simple pipeline of masking random cubes and reconstructing the missing ones. However, the extra time dimension of videos makes them different from images in this masked modeling. First, video frames are often densely captured, and their semantics varies slowly in time [58]. This temporal redundancy would increase the risk of recovering missing pixels from the spatiotemporal neighborhood with little high-level understanding. Furthermore, video could be viewed as the temporal evolution of static appearance, and there exists a correspondence between frames. This temporal correlation could lead to information leakage (i.e., masked spatiotemporal content re-occurrence) during reconstruction unless a specific masking strategy is considered. In this sense, for each masked cube, it is easy to find a corresponding and unmasked copy in adjacent frames. This property would make the learned models identify some "shortcut" features that are hard to generalize to new scenarios.

To make video masked modeling more effective, in this paper, we present a customized design of tube masking with an extremely high ratio in our VideoMAE. First, due to temporal redundancy, we use an *extremely high* masking ratio to drop the cubes from the downsampled clips. This simple strategy not only effectively increases the pre-training performance but also greatly reduces the computational cost due to the asymmetric encoder-decoder architecture. Second, to consider temporal correlation, we devise a simple yet effective *tube masking* strategy, which turns out to be helpful in relieving the risk of information leakage for cubes with no or negligible motion during reconstruction. With this simple yet effective design in our VideoMAE, we are able to successfully train vanilla ViT backbones on the relatively small-scale video datasets such as Something-Something [19], UCF101 [39], and HMDB51 [24], which significantly outperform the previous state of the art under the setting without extra data. In summary, the main contribution of this paper is threefold:

- We present a simple but effective video masked autoencoder that unleashes the potential of vanilla vision transformer for video analysis and performs well for SSVP on relatively small-scale video datasets. To relieve the information leakage issue in the video reconstruction task, we present a simple design of tube masking with an extremely high ratio, which turns out to be effective in improving the performance of VideoMAE.
- Aligned with the results in NLP and Images on masked modeling, our VideoMAE demonstrates that this simple masking and reconstruction strategy provides a good solution to self-supervised video pre-training. The models pre-trained with our VideoMAE significantly outperform those trained from scratch or pre-trained with contrastive learning methods.
- We obtain extra important finds on masked modeling that might be ignored in the previous research in NLP and Images. (1) Our results demonstrate that VideoMAE is a data-efficient learner that could be successfully trained with only 3.5k videos. (2) VideoMAE shows that data quality is more important than quantity for SSVP when a domain shift exists between the source and target dataset.

2 Related Works

Video representation learning. Learning good video representations has been heavily investigated in the literature. The supervised learning methods [49, 44, 17, 16, 4] usually depend on the image backbones. The video encoder backbones are first pre-trained with image data in a supervised form. Then, these backbones are fine-tuned on the video dataset for classifying human actions. Besides supervised learning, semi-supervised video representation learning is also studied [38]. The representations of labeled training samples are utilized to generate supervision signals for unlabeled ones. Supervised or semi-supervised representation learning mainly uses a top-down training paradigm, which is not effective in exploring the inherent video data structure itself.

For self-supervised learning, the prior knowledge of temporal information has been widely exploited to design pretext tasks [52, 28, 56, 3] for SSVP. Recently, contrastive learning [20, 29, 21, 33] is popular to learn better video representation. However these methods heavily rely on strong data augmentation and large batch size [18]. Predicting the video clip with autoencoders in pixel space has been explored for representation learning [31, 40] by using CNN or LSTM backbones or video generation with autoregressive GPT [57]. Our VideoMAE aims to use the simple masked autoencoder with recent ViT backbones to perform data-efficient SSVP.

Masked visual modeling. Masked visual autoencoder has been proposed to learn effective visual representations based on the simple pipeline of masking and reconstruction. These works mainly focus on the image domain. The early work [47] treated the masking and a noise type in denoised autoencoders [46] or inpainted missing regions with context [30] by using convolutions. Recently, masked visual autoencoder used Transformer backbones for masking and reconstruction. iGPT [9] followed the success of GPT [5, 35] in NLP and operated a sequence of pixels for prediction. The original ViT [14] investigated the masked token prediction for self-supervised pre-training. More recently, the success of vision transformer has led to investigation of Transformer-based architectures for masked visual modeling [2, 13, 22, 53, 55, 60]. BEiT [2], BEVT [50] and VIMPAC [41] followed BERT [12] and proposed to learn visual representations from images and videos by predicting the discrete tokens [36]. ImageMAE [22] introduced an asymmetric encoder-decoder architecture for masked image modeling. MaskFeat [53] proposed to reconstruct the features of masked tokens to perform self-supervised pre-training in videos.

VideoMAE is inspired by the ImageMAE and introduces specific design in implementation for SSVP. Our work differs from these video masked modeling methods in several aspects. 1) *Prediction target*. Our VideoMAE performs pre-training by directly predicting the normalized cube pixels. BEVT and VIMPAC predict the discrete tokens from pre-trained VQ-VAE. MaskFeat predict the HOG features. 2) *Computational efficiency*. Our VideoMAE only needs to feed the unmasked tokens (e.g., 10%) into the encoder in self-supervised learning. BEVT, VIMPAC and MaskFeat put all tokens into the encoder during the pre-training phase. 3) *Backbone*. Our VideoMAE uses the plain ViT backbone, while the Swin and MViT in BEVT and MaskFeat are with multi-scale and hierarchical architectures. VIMPAC uses a pre-trained VQ-VAE to discrete the input video into discrete video tokens and adopt a sparse transformer on top of the tokens. 4) *Prediction head*. Our VideoMAE generates the target by a shallow Transformer decoder, while BEVT and MaskFeat all use two-layer MLP as the target head. VIMPAC performs pre-training by predicting the discrete tokens from pre-trained VQ-VAE and introducing a contrastive learning head. Overall, our VideoMAE enjoys a simpler design for the whole pipeline, obtaining higher efficiency and slightly better accuracy.

3 Proposed Method

In this section, we first revisit ImageMAE [22]. Then we analyze the characteristics of video data. Finally, we show how we explore MAE in the video data by presenting our VideoMAE.

3.1 Revisiting Masked Autoencoders

ImageMAE [22] performs the masking and reconstruction task with an asymmetric encoder-decoder architecture. The input image $I \in \mathcal{R}^{3\times H\times W}$ is first divided into regular non-overlapping patches of size 16 $\times$16, and each patch is represented with *token embedding*. Then a subset of tokens are randomly masked with a high masking ratio (75%), and only the remaining ones are fed into the transformer *encoder* Φ_{enc}. Finally, a shallow *decoder* Φ_{dec} is placed on top of the visible tokens from

the encoder and learnable mask tokens to reconstruct the image. The *loss function* is mean squared error (MSE) loss between the normalized masked tokens and reconstructed ones in the pixel space:

$$\mathcal{L} = \frac{1}{\Omega} \sum_{p \in \Omega} |I(p) - \hat{I}(p)|^2, \tag{1}$$

where p is the token index, Ω is the set of masked tokens, I is the input image, and $\hat{I}$ is the reconstructed one.

3.2 Characteristics of Video Data

Compared with static images, video data contain temporal relations. We show the motivation of our VideoMAE by analyzing video characteristics.

Temporal redundancy. There are frequently captured frames in a video. The semantics vary slowly in the temporal dimension [58]. We observe that consecutive frames are highly redundant, as shown in Figure 2. This property leads to two critical issues in masked video autoencoding. First, it would be less efficient to keep the original temporal frame rate for pre-training. This would draw us to focus more on static or slow motions in our masked modeling. Second, temporal redundancy greatly dilutes motion representations. This would make the task of reconstructing missing pixels not difficult under the normal masking ratio (e.g., 50% to 75%). The encoder backbone is not effective in capturing motion representations.

Temporal correlation. Videos could be viewed as the temporal extension of static appearance, and therefore there exists an inherent correspondence between adjacent frames. This temporal correlation could increase the risk of information leakage in the masking and reconstruction pipeline. In this sense, as shown in Figure 2, we can reconstruct the masked patches by finding the spatiotemporal corresponding unmasked patches in the adjacent frames under plain random masking or frame masking. In this case, it might guide the VideoMAE to learn low-level temporal correspondence rather than high-level information such as spatiotemporal reasoning over the content. To alleviate this behavior, we need to propose a new masking strategy to make the reconstruction more challenging and encourage effective learning of spatiotemporal structure representations.

3.3 VideoMAE

To relieve the above issues in video masked modeling, we make the customized design in our VideoMAE, and the overall pipeline is shown in Figure 1. Our VideoMAE takes the *downsampled frames* as inputs and uses the *cube embedding* to obtain video tokens. Then, we propose a simple design of *tube masking with high ratio* to perform MAE pre-training with an asymmetric encoder-decoder architecture. Our backbone uses the vanilla ViT with *joint space-time attention*.

Temporal downsampling. According to the above analysis on temporal redundancy over consecutive frames, we propose to use the strided temporal sampling strategy to perform more efficient video pre-training. Formally, one video clip consisting of t consecutive frames is first randomly sampled from the original video V. We then use temporal sampling to compress the clip to T frames, each of which contains $H \times W \times 3$ pixels. In experiments, the stride τ is set to 4 and 2 on Kinetics and Something-Something, respectively.

Cube embedding. We adopt the joint space-time cube embedding [1, 15, 26] in our VideoMAE, where we treat each cube of size $2 \times 16 \times 16$ as one token embedding. Thus, the cube embedding layer obtains $\frac{T}{2} \times \frac{H}{16} \times \frac{W}{16}$ 3D tokens and maps each token to the channel dimension D. This design can decrease the spatial and temporal dimension of input, which helps to alleviate the spatiotemporal redundancy in videos.

Tube masking with extremely high ratio. First, temporal redundancy is a factor affecting VideoMAE design. We find that VideoMAE is in favor of extremely high masking ratios (e.g. 90% to 95%) compared with the ImageMAE. Video information density is much lower than images, and we expect a high ratio to increase the reconstruction difficulty. This high masking ratio is helpful to mitigate the information leakage during masked modeling and make masked video reconstruction a meaningful self-supervised pre-training task.

Second, temporal correlation is another factor in our VideoMAE design. We find even under the extremely high masking ratio, we can still improve the masking efficiency by proposing the temporal

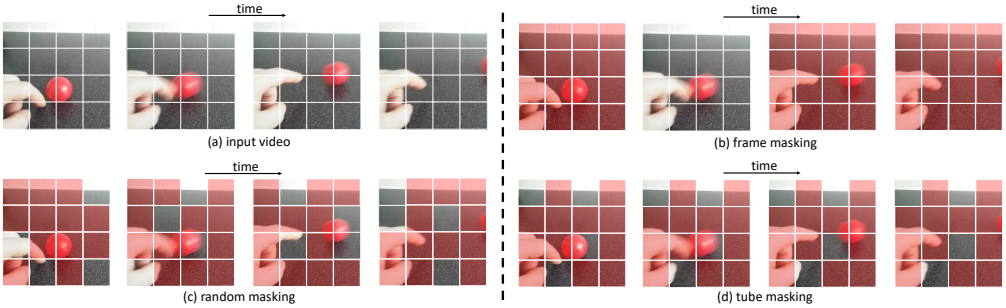


Figure 2: Slowness is a general prior in (a) input video data [58]. This leads to two important characteristics in time: temporal redundancy and temporal correlation. Temporal redundancy makes it possible to recover the pixel under an extremely high masking ratio. Temporal correlation leads us to easily reconstruct the missing pixel by finding those corresponding patches in adjacent frames under plain (b) frame masking or (c) random masking. To avoid this simple task and encourage learning representative representation, we propose a (d) tube masking, where the masking map is the same for all frames.

tube masking mechanism. Temporal tube masking enforces a mask to expand over the whole temporal axis, namely, different frames sharing the same masking map. Mathematically, the tube mask mechanism can be expressed as $\mathbb{I}[p_{x,y,\cdot} \in \Omega] \sim \mathrm{Bernoulli}(\rho_{\mathrm{mask}})$ and different time t shares the same value. With this mechanism, temporal neighbors of masked cubes are always masked. So for some cubes with no or small motion (e.g., finger cube in 4th row of Figure 2 (d)), we can not find the spatiotemporal corresponding content in all frames. In this way, it would encourage our VideoMAE to reason over high-level semantics to recover these totally missing cubes. This simple strategy can alleviate the information leakage for cubes with no or negligible motion and turns out to be effective in practice for masked video pre-training.

Backbone: joint space-time attention. Due to the high proportion of masking ratio mentioned above, only a few tokens are left as the input for the encoder. To better capture high-level spatio-temporal information in the remaining tokens, we use the vanilla ViT backbone [14] and adopt the joint space-time attention [1, 26]. Thus, all pair tokens could interact with each other in the multi-head self-attention layer [45]. The specific architecture design for the encoder and decoder is shown in supplementary materials. The quadratic complexity of the joint space-time attention mechanism is a computational bottleneck, while our design of an extremely high masking ratio alleviates this issue by only putting the unmasked tokens (e.g., 10%) into the encoder during the pre-training phase.

4 Experiments

4.1 Datasets

We evaluate our VideoMAE on four canonical video recognition datasets: Kinetics-400 [23], Something-Something V2 [19], UCF101 [39], and HMDB51 [24]. The Kinetics-400 contains around 240k training videos and 20k validation videos of 10s from 400 classes. The Something-Something V2 is a relatively large-scale video dataset, having around 169k videos for training and 20k videos for validation. In contrast to Kinetics-400, this dataset contains 174 motion-centric action classes. These two large-scale video datasets focus on different visual cues for action recognition. UCF101 and HMDB51 are two relatively small and old video datasets, which contain around 9.5k/3.5k train/val videos and 3.5k/1.5k train/val videos, respectively. Compared with those large-scale video datasets, these two small datasets are more suitable for verifying the effectiveness of VideoMAE, as training large ViT models is more challenging on small datasets. In experiments, we strictly follow the official setting on these datasets by training VideoMAE on the training set and reporting the top-1 accuracy on the validation set. The implementation details are described in supplementary materials.

4.2 Ablation Studies

In this subsection, we perform in-depth ablation studies of VideoMAE with the default backbone 16-frame vanilla ViT-B on Something-Something V2. The specific architecture design for the encoder

blocks	top-1	top-5	mem.
1	68.2	91.7	7.9G
2	68.9	92.1	10.2G
4	**69.3**	**92.3**	14.7G
8	69.0	92.1	23.7G

(a) **Decoder depth**. 4 blocks of decoder achieve the best trade-off. "mem." is GPU memory during pre-training, benchmarked in one GPU with a batch size of 16.

case	ratio	top-1	top-5
tube	75	67.3	91.5
tube	90	**69.3**	**92.3**
random	90	68.3	91.8
frame	87.5*	61.5	87.6

(b) **Mask sampling**. We compare different masking strategies. Our proposed tube masking with an extremely high ratio works the best. $*$"87.5" means masking 14/16 frames.

input	target	top-1	top-5
16×2	*center*	62.7	88.4
16×1	16×1	53.2	80.6
16×2	16×2	**69.3**	**92.3**
16×2	32×1	69.0	92.0

(c) **Reconstruction target**. $T \times \tau$ denotes frames $\times$stride. *center* denotes the center frame of the input clip. 16×2 subsampled video frames as reconstruction targets are effective and best.

case	top-1	top-5
from scratch	32.6	61.9
ImageNet-21k sup.	61.4	86.6
IN-21k+K400 sup.	64.8	89.8
VideoMAE	**69.3**	**92.3**

(d) **Pre-training strategy**. Our VideoMAE works the best without using any extra data. "sup." is supervised training.

dataset	method	top-1	top-5
IN-1K	ImageMAE	64.5	89.5
K400	VideoMAE	68.2	91.6
SSV2	VideoMAE	**69.3**	**92.3**

(e) **Pre-training dataset**. Our VideoMAE works best when directly pre-training on Something-Something V2.

case	top-1	top-5
L1 loss	68.8	92.1
MSE loss	**69.3**	**92.3**
Smooth L1 loss	68.6	92.1

(f) **Loss function**. MSE loss works the best for the masking and reconstruction task in VideoMAE.

Table 1: Ablation experiments on **Something-Something V2**. Our backbone is 16-frame vanilla ViT-B and all models are pre-trained with *mask ratio* ρ=90% for 800 epochs. We use the fine-tuning setting for evaluation. We perform the uniform sampling following TSN [49] and all models share the same inference protocol, i.e., 2 clips $\times$3 crops. The default choice for our model is colored in gray.

and decoder is shown in supplementary materials. For the fine-tuning setting, We perform the uniform sampling following TSN [49] and all models share the same inference protocol, i.e., 2 clips $\times$3 crops.

Decoder design. The lightweight decoder is one key component of our VideoMAE. We conduct experiments with the different depths in Table 1a. Unlike in ImageMAE, a deep decoder here is important for better performance, while a shallow decoder could reduce the GPU memory consumption. We take 4 blocks for the decoder by default. The decoder width is set to half channel of the encoder (e.g., 384-d for ViT-B), following the empirical design in the image domain.

Masking strategy. We compare different masking strategies in Table 1b. When increasing the masking ratio from 75% to 90% for tube masking, the performance boosts from 67.3% to 69.3%. Then, with an extremely high ratio, we find tube masking also achieves better performance than plain random masking and frame masking. We attribute these interesting observations to the redundancy and temporal correlation in videos. We argue that our default designs enforce the networks to capture more useful spatiotemporal structures and therefore make VideoMAE a more challenging task, which a good self-supervised learner hunger for.

Reconstruction target. First, if we only employ the center frame as the target, the results would decrease greatly. The sampling stride is also sensitive. The result of consecutive 16 frames is significantly lower than our default downsampled clip (53.2% vs. 69.3%). We also try to reconstruct the consecutive 32 frames from the downsampled 16 frames, but it obtains slightly worse results.

Pre-training strategy. We compare different pre-training strategies in Table 1d. Similar to previous trials [1, 4], training video transformers from scratch yields unsatisfied results on this motion-sensitive dataset. When pre-trained on the large-scale ImageNet-21K dataset, the video transformer obtains better accuracy from 32.6% to 61.4%. Using the models pre-trained on both ImageNet-21K and Kinetics further increases accuracy to 64.8%. Our VideoMAE can effectively train a video transformer on the video dataset itself without using any extra data and achieve the best performance of 69.3%.

Pre-training dataset. First, we pre-train the ViT-B on ImageNet-1K for 1600 epochs, following the recipes in [22]. Then we inflate the 2D patch embedding layer to our cube embedding layer following [8] and fine-tune the model on the Something-Something dataset. This result surpasses the model trained *from scratch* as shown in Table 1e. We also compare this ImageMAE pre-trained model with the VideoMAE model pre-trained on Kinetics. We see that our VideoMAE model can achieve better performance than ImageMAE. However, these two pre-trained models fail to yield

dataset	training data	*from scratch*	MoCo v3	VideoMAE
K400	240k	68.8	74.2	**79.4**
Sth-Sth V2	169k	32.6	54.2	**69.3**
UCF101	9.5k	51.4	81.7	**90.8**
HMDB51	3.5k	18.0	39.2	**61.1**

Table 2: Comparisons with the results of previous **self-supvised pre-training methods** on different datasets. We take 16-frame ViT-B as the default backbone. Notably, here MoCo v3 and VideoMAE all only use the *unlabelled* data in the training set of each dataset for pre-training and are all fine-tuned for evaluation.

method	epoch	ft. acc.	lin. acc.	hours	speedup
MoCo v3	300	54.2	33.7	61.7	-
VideoMAE	800	**69.3**	38.9	19.5	**3.2**$\times$

Table 3: Comparisons with the **efficiency and effectiveness** on Something-Something V2. We report the fine-tuning (ft) and linear probing (lin) accuracy (%). The **wall-clock time** of pre-training is benchmarked in 64 Tesla V100 GPUs with PyTorch.

method	K400 $\rightarrow$ SSV2	K400 $\rightarrow$ UCF	K400 $\rightarrow$ HMDB
MoCo v3	62.4	93.2	67.9
VideoMAE	**68.2**	**96.1**	**73.3**

Table 4: Comparisons with the **feature transferability** on smaller datasets. We take 16-frame ViT-B as the default backbone. Notably, here MoCo v3 and VideoMAE are all pre-trained on Kinetics-400 with *unlabelled* data in the training set. Then the pre-trained model is fine-tuned on target datasets for evaluation.

better performance than our VideoMAE only pre-trained on the Something-Something dataset. We argue that domain shift between pre-training and target datasets could be an important issue.

Loss function. Table 1f contains an ablation study of loss function. We find that the MSE loss could achieve a higher result compared with the L1 loss and smooth L1 loss. Therefore, we employ the MSE loss by default.

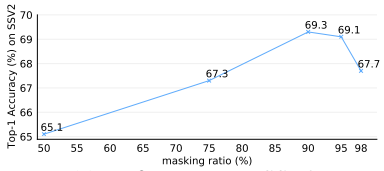
4.3 Main Results and Analysis

VideoMAE: data-efficient learner. The self-supervised video pre-training (SSVP) has been extensively studied in previous works, but they mainly use the CNN-based backbones. Few works have investigated transformer-based backbone in SSVP. Therefore, to demonstrate the effectiveness of VideoMAE for transformer-based SSVP, we compare two methods implemented by ourselves: (1) training from scratch and (2) pre-training with contrastive learning (MoCo v3 [10]). For training from scratch, we carefully tune these hyper-parameters to successfully pre-train ViT-Base from the training set of the dataset. For pre-training with MoCo v3, we strictly follow the training practice in its image counterpart and carefully avoid the collapse issue.

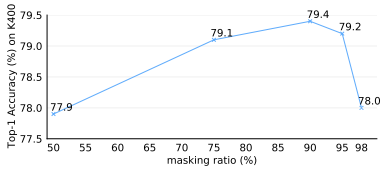
The recognition accuracy is reported in Table 2. We see that our VideoMAE significantly outperforms the other two training methods. For instance, on the largest dataset of Kinetics-400, our VideoMAE outperforms training from scratch by around 10% and MoCo v3 pre-training by around 5%. This superior performance demonstrates that masked autoencoder provides an effective pre-training mechanism for video transformers. We also see that the performance gap between our VideoMAE and the other two methods becomes larger as the training set becomes smaller. Notably, even with only 3.5k training clips on HMDB51, our VideoMAE pre-training can still obtain a satisfying accuracy (around 61%). This new result demonstrates that VideoMAE is a more data-efficient learner for SSVP. This property is particularly important for scenarios with limited data available and different with contrastive learning methods.

We compare the efficiency of VideoMAE pre-training and MoCo v3 pre-training in Table 3. The task of masked autoencoding with a high ratio is more challenging and thereby requires more training epochs (800 vs. 300). Thanks to the asymmetric encoder-decoder in our VideoMAE and extremely high masking ratio, our pre-training time is much shorter than MoCo v3 (19.5 vs. 61.7 hours).

High masking ratio. In VideoMAE, one core design is the extremely high masking ratio. We perform an investigation of this design on the Kinetics-400 and Something-Something V2 datasets. The results are shown in Figure 3. We see that the best masking ratio is extremely high, and even



(a) Performance on SSV2



(b) Performance on Kinetics-400

Figure 3: The effect of **masking ratio** on (a) Something-Something V2 and (b) Kinetics-400. We take 16-frame vanilla ViT-B as default. The results show that an extremely high masking ratio (90%) achieves the best efficiency and effectiveness trade-off on both video datasets.

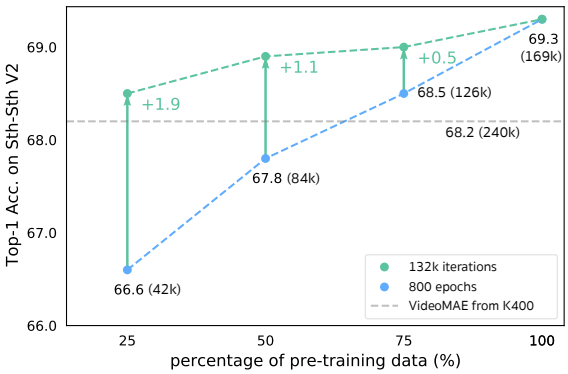


Figure 4: **Data efficiency** of VideoMAE representations. Our default backbone is 16-frame vanilla ViT-B. ● denotes that all models are trained for the **same** 132k iterations, and ● denotes that all models are trained for the **same** 800 epochs. Note that it takes 132k iterations to pre-train the model for 800 epochs on the full training set of Something-Something V2.

Method	Backbone	Extra data	Ex. labels	Frames	GFLOPs	Param	Top-1	Top-5
SlowFast [17]	ResNet101	Kinetics-400	✓	8+32	106×1×3	53	63.1	87.6
MViTv1 [15]	MViTv1-B		✓	64	455×1×3	37	67.7	90.9
TimeSformer [4]	ViT-B	ImageNet-21K	✓	8	196×1×3	121	59.5	N/A
TimeSformer [4]	ViT-L		✓	64	5549×1×3	430	62.4	N/A
ViViT FE [1]	ViT-L	IN-21K+K400	✓	32	995×4×3	N/A	65.9	89.9
Motionformer [32]	ViT-B		✓	16	370×1×3	109	66.5	90.1
Motionformer [32]	ViT-L		✓	32	1185×1×3	382	68.1	91.2
Video Swin [26]	Swin-B		✓	32	321×1×3	88	69.6	92.7
VIMPAC [41]	ViT-L	HowTo100M+DALLE	✗	10	N/A×10×3	307	68.1	N/A
BEVT [50]	Swin-B	IN-1K+K400+DALLE	✗	32	321×1×3	88	70.6	N/A
MaskFeat↑312 [53]	MViT-L	Kinetics-600	✓	40	2828×1×3	218	75.0	95.0
VideoMAE$_{\rho=90\%}$	ViT-B	Kinetics-400	✗	16	180×2×3	87	69.7	92.3
VideoMAE$_{\rho=90\%}$	ViT-L	Kinetics-400	✗	16	597×2×3	305	74.0	94.6
VideoMAE$_{\rho=90\%}$	ViT-B	*no external data*	✗	16	180×2×3	87	70.6	92.7
VideoMAE$_{\rho=90\%}$	ViT-L		✗	16	597×2×3	305	74.2	94.7
VideoMAE$_{\rho=90\%}$	ViT-L		✗	32	1436×1×3	305	**75.3**	**95.2**

Table 5: **Comparison with the state-of-the-art methods on Something-Something V2**. Our VideoMAE$_{\rho=90\%}$ reconstructs normalized cube pixels and is pre-trained for 2400 epochs. "Ex. labels ✗" means only *unlabelled* data is used during the pre-training phase.

95% can achieve good performance for both datasets. This result is difference from BERT [12] in NLP and MAE [22] in images. We analyze the temporal redundancy and correlation in videos makes it possible for our VideoMAE to learn plausible outputs with such a high masking ratio.

We also visualize the reconstructed examples in supplementary materials. We see that even under an extremely high masking ratio, VideoMAE can produce satisfying reconstructed results. This implies that our VideoMAE is able to learn useful representations that capture the holistic spatiotemporal structure in videos.

Transfer learning: quality vs. quantity. To further investigate the generalization ability of VideoMAE in representation learning, we transfer the learned VideoMAE from Kinetics-400 to Something-Something V2, UCF101, and HMDB51. The results are shown in Table 4, and we compare them with the MoCo v3 pre-training method. The models pre-trained by our VideoMAE are better than those pre-trained by MoCo v3, demonstrating that our VideoMAE learns more transferable representations.

Method	Backbone	Extra data	Ex. labels	Frames	GFLOPs	Param	Top-1	Top-5
NL I3D [51]	ResNet101	ImageNet-1K	✓	128	359×10×3	62	77.3	93.3
TANet [27]	ResNet152		✓	16	242×4×3	59	79.3	94.1
TDN_{En} [48]	$\text{ResNet101}_{\times 2}$		✓	8+16	198×10×3	88	79.4	94.4
Video Swin [26]	Swin-B		✓	32	282×4×3	88	80.6	94.6
TimeSformer [4]	ViT-B	ImageNet-21K	✓	8	196×1×3	121	78.3	93.7
TimeSformer [4]	ViT-L		✓	96	8353×1×3	430	80.7	94.7
ViViT FE [1]	ViT-L		✓	128	3980×1×3	N/A	81.7	93.8
Motionformer [32]	ViT-B		✓	16	370×10×3	109	79.7	94.2
Motionformer [32]	ViT-L		✓	32	1185×10×3	382	80.2	94.8
Video Swin [26]	Swin-L		✓	32	604×4×3	197	83.1	95.9
ViViT FE [1]	ViT-L	JFT-300M	✓	128	3980×1×3	N/A	83.5	94.3
ViViT [1]	ViT-H	JFT-300M	✓	32	3981×4×3	N/A	84.9	95.8
VIMPAC [41]	ViT-L	HowTo100M+DALLE	✗	10	N/A×10×3	307	77.4	N/A
BEVT [50]	Swin-B	IN-1K+DALLE	✗	32	282×4×3	88	80.6	N/A
$\text{MaskFeat}_{\uparrow 352}$ [53]	MViT-L	Kinetics-600	✗	40	3790×4×3	218	**87.0**	**97.4**
ip-CSN [43]	ResNet152	*no external data*	✗	32	109×10×3	33	77.8	92.8
SlowFast [17]	R101+NL		✗	16+64	234×10×3	60	79.8	93.9
MViTv1 [15]	MViTv1-B		✗	32	170×5×1	37	80.2	94.4
MaskFeat [53]	MViT-L		✗	16	377×10×1	218	84.3	96.3
VideoMAE$_{\rho=90\%}$	ViT-B	*no external data*	✗	16	180×5×3	87	80.9	94.7
VideoMAE$_{\rho=90\%}$	ViT-L		✗	16	597×5×3	305	**84.7**	**96.5**

Table 6: **Comparison with the state-of-the-art methods on Kinetics-400**. Our $\text{VideoMAE}_{\rho=90\%}$ reconstructs normalized cube pixels. Here models are self-supervised pre-trained for 1600 epochs on Kinetics-400. "Ex. labels ✗" means only *unlabelled* data is used during the pre-training phase.

Comparing Table 2 and Table 4, the transferred representation outperforms the original VideoMAE models trained from its own dataset on UCF101 and HMDB51. In contrast, the transferred representation is worse on Something-Something V2. To figure out whether this inconsistent result is caused by the large scale of Something-Something V2, we further perform a detailed investigation by decreasing the pre-training video numbers. In this study, we run two experiments: (1) pre-training with the same epochs and (2) pre-training with the same time budget. The result is shown in Figure 4. We see that more training iterations could contribute to better performance when we decrease the size of the pre-training set. Surprisingly, even with only 42k pre-training videos, we can still obtain better accuracy than the K400 pre-trained models with 240k videos (68.5% vs. 68.2%). This result implies that domain shift is another important factor, and data quality is more important than data quantity in SSVP when there exists a difference between pre-training and target datasets. It also demonstrates that VideoMAE is a data-efficient learner for SSVP.

Comparison with the state of the art. We compare with the previous state-of-the-art performance on the Kinetics-400 and Something-Something V2 datasets. The results are reported in Table 5 and Table 6. Our VideoMAE can easily scale up with more powerful backbones (ViT-Large) and more frames (32). Our VideoMAE achieves the top-1 accuracy of 75.3% on the Something-Something V2 and 84.7% on the Kinetics-400 without using any extra data. We see that the existing state-of-the-art methods all depend on the external data for pre-training on the Something-Something V2. On the contrary, our VideoMAE without any external data significantly outperforms previous methods with the same input resolution by around 5%. Our ViT-L VideoMAE also achieves very competitive performance on the Kinetics-400 without using any extra data, which is slightly better than the ViViT FE-L with on JFT-300M pre-training.

5 Conclusion

In this paper, we have presented a simple and data-efficient self-supervised learning method (VideoMAE) for video transformer pre-training. Our VideoMAE introduces two critical designs of extremely high masking ratio and tube masking strategy to make the video reconstruction task more challenging. This harder task would encourage VideoMAE to learn more representative features and relieve the information leakage issue. Empirical results demonstrate this simple algorithm works well for video datasets of different scales. In particular, we are able to learn effective VideoMAE only with thousands of video clips, which has significant practical value for scenarios with limited data available.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *IEEE/CVF International Conference on Computer Vision*, 2021.

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

[3] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 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 *Advances in Neural Information Processing Systems*, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[8] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

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

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[11] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[13] 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 preprint arXiv:2111.12710*, 2021.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[15] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[16] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[18] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[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 *IEEE/CVF International Conference on Computer Vision*, 2017.

[20] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, 2020.

[21] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems*, 2020.

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[23] 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.

[24] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *IEEE/CVF International Conference on Computer Vision*, 2011.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[26] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[27] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[28] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, 2016.

[29] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[30] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[31] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. *arXiv preprint arXiv:1511.06309*, 2015.

[32] Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joo F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems*, 2021.

[33] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

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

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

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 2021.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[38] Ankit Singh, Omprakash Chakraborty, Ashutosh Varshney, Rameswar Panda, Rogerio Feris, Kate Saenko, and Abir Das. Semi-supervised action recognition with temporal contrastive learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[39] 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.

[40] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *International Conference on Machine Learning*, 2015.

[41] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021.

[42] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021.

[43] Du Tran, Heng Wang, Matt Feiszli, and Lorenzo Torresani. Video classification with channel-separated convolutional networks. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[44] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[46] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, 2008.

[47] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 2010.

[48] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. TDN: Temporal difference networks for efficient action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[49] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[50] 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. *arXiv preprint arXiv:2112.01529*, 2021.

[51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[52] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *IEEE/CVF International Conference on Computer Vision*, 2015.

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

[54] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, 2021.

[55] 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. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[56] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[57] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[58] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[59] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

[60] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Shown in supplementary materials.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Shown in supplementary materials.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Shown in supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Shown in supplementary materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Because of the computation costs, we did not run the experiments multiple times.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see Table 3 and supplementary materials.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] Shown in supplementary materials.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We used publicly available datasets whose licenses allow research usage.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] To the best of our knowledge, the data we used contains no personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]