VideoMAE V2: Scaling Video Masked Autoencoders with Dual Masking

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

Abstract

Scale is the primary factor for building a powerful foundation model that could well generalize to a variety of downstream tasks. However, it is still challenging to train video foundation models with billions of parameters. This paper shows that video masked autoencoder (VideoMAE) is a scalable and general self-supervised pre-trainer for building video foundation models. We scale the VideoMAE in both model and data with a core design. Specifically, we present a dual masking strategy for efficient pre-training, with an encoder operating on a subset of video tokens and a decoder processing another subset of video tokens. Although VideoMAE is very efficient due to high masking ratio in encoder, masking decoder can still further reduce the overall computational cost. This enables the efficient pre-training of billion-level models in video. We also use a progressive training paradigm that involves an initial pre-training on a diverse multi-sourced unlabeled dataset, followed by a post-pre-training on a mixed labeled dataset. Finally, we successfully train a video ViT model with a billion parameters, which achieves a new state-of-the-art performance on the datasets of Kinetics (90.0% on K400 and 89.9% on K600) and Something-Something (68.7% on V1 and 77.0% on V2). In addition, we extensively verify the pre-trained video ViT models on a variety of downstream tasks, demonstrating its effectiveness as a general video representation learner.

1. Introduction

Effectively pre-training large foundation models [5] on huge amounts of data is becoming a successful paradigm in learning generic representations for multiple data modalities (e.g., language [6, 16], audio [13, 50], image [3, 22, 79], video [18, 63, 76], vision-language [27, 55]). These foundation models could be easily adapted to a wide range of downstream tasks through zero-shot recognition, linear probe, prompt tuning, or fine tuning. Compared with the specialized model to a single task, they exhibit excellent generalization capabilities and have become the main driving force for advancing many areas in AI.

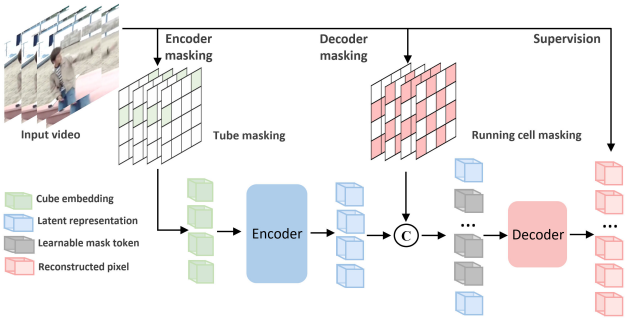

Figure 1. **VideoMAE with dual masking.** To improve the overall efficiency of computation and memory in video masked autoencoding, we propose to mask the decoder as well and devise the dual masking strategy. Like encoder, we also apply a masking map to the deocoder and simply reconstruct a subset of pixel cubes selected by the running cell masking. The final reconstruction loss only applies for the invisible tokens dropped by the encoder.

For vision research, many efforts have been devoted to developing effective pre-trained models. Among them, Transformer [65] with masked autoencoding [16] is becoming a conceptually simple yet effective self-supervised visual learner (e.g., BEiT [3], SimMIM [79], MAE [22] for images, and MaskFeat [76], VideoMAE [63], MAE-ST [18] for videos). Meanwhile, based on the results in language models [6], scaling model capacity and data size is an important ingredients for its remarkable performance improvement. However, for pre-trained vision models, very few work [44] has tried to scale up this masked autoencoder pre-training to the billion-level models in image domain, partially due to the high data dimension and the high computational overhead. This issue is even more serious for scaling up video masked autoencoder pre-training owning to its extra time dimension and strong temporal variations.

Following the promising findings in languages and images, we aim to *study the scaling property of video*

* : Equal contribution.

masked autoencoder (VideoMAE), and *push its performance limit on a variety of video downstream tasks*. We scale VideoMAE in both model and data. For model scaling, we try to instantiate the VideoMAE with vision transformer (ViT) [17] having billion-level parameters (e.g., ViT-g [84]), and for data scaling, we hope to increase the pre-training dataset size to million-level to fully unleash the power of billion-level ViT model. However, to successfully train giant VideoMAE on such huge amounts of data and achieve impressive improvements on all considered downstream tasks, we still need to carefully address a few issues.

First, we find computational cost and memory consumption is the bottleneck of scaling VideoMAE on the current GPUs with limited memory. Although VideoMAE [63] has improved its pre-training efficiency and reduced its memory consumption by employing the efficient asymmetric encoder-decoder architecture [22] (i.e., dropping large numbers of tokens in encoder), it still fails to well support the billion-level video transformer pre-training. It takes more than two weeks to pre-train a ViT-g model with VideoMAE on 64 A100 GPUs. To further improve its pre-training efficiency, we find video data redundancy can be used to not only mask a high portion of cubes in the encoder, but also drop some cubes in the decoder. This solution yields higher pre-training efficiency and creates a similarly challenging and meaningful self-supervised task. In practice, it will increase the pre-training batchsize and reduce the pre-training time by a third with almost no performance drop.

Second, MAE is still demanding for large data [80] and billion-level video transformer tends to overfit on relatively small data. Unlike images, the existing public video dataset is much smaller. For example, there are only 0.24M videos in the Kinetics400 dataset [28], while the ImageNet-22k dataset [15] has 14.2M images, let alone those publicly inaccessible image datasets such as JFT-3B [84]. Therefore, we need to come up with new ways to build a larger video pre-training dataset to well support the billion-level video transformer pre-training. We show that simply mixing the video datasets from multiple resources could produce an effective and diverse pre-training dataset for VideoMAE and improve its downstream performance of pre-trained models.

Finally, it is still unknown how to adapt the billion-level pre-trained model by VideoMAE. Masked autoencoding is expected to learn invariant features that provide a favored initialization for vision transformer fine-tuning [30]. However, directly fine-tuning billion-level pre-trained models on a relatively small video dataset (e.g., 0.24M videos) might be suboptimal, as the limited labeled samples might lead to overfitting issue in fine-tuning. In fact, in image domain, the intermediate fine-tuning technique [3, 44] has been employed to boost the performance of masked pre-trained models. We show that collecting multiple labeled video datasets and building a supervised hybrid dataset can act as a bridge between the large-scale unsupervised dataset and the small-scale downstream target dataset. Progressive fine-tuning of the pre-trained models through this labeled hybrid dataset could contribute to higher performance in the downstream tasks.

Based on the above analysis, we present a simple and efficient way to scale VideoMAE to billion-level ViT models on a dataset containing million-level pre-training videos. Our technical improvement is to introduce the *dual masking strategy* for masked autoencoder pipeline as shown in Figure 1. In addition to the masking operation in encoder, we propose to mask decoder as well based on the data redundancy prior in video. With this dual-masked VideoMAE, we follow the intermediate fine-tuning in images [3, 44], and use a progressive training pipeline to perform the video masked pre-training on the million-level unlabeled video dataset and then post-pre-training on the labeled hybrid dataset. These core designs contribute to an efficient billion-level video autoencoding framework, termed as **VideoMAE V2**. Within this framework, *we successfully train the first video transformer model with one billion parameters*, which attains a new state-of-the-art performance on a variety of downstream tasks, including action recognition [20, 28, 32, 57], spatial action detection [21, 34], and temporal action detection [26, 43].

2. Related Work

Vision foundation models. The term of foundation model was invented in [5]. It refers to those powerful models that are pre-trained on broad data and can be adapted to a wide range of downstream tasks. Early research works in vision focused on pre-training CNNs [33] or Transformers [65] on large-scale labeled datasets such as ImageNet-1k [24, 31], ImageNet-22k [45, 73], and JFT [84]. Some recent works tried to perform unsupervised pre-training using contrastive learning [10, 23, 77] or siamese learning [11]. Meanwhile, following the success in NLP [6, 16], masked autoencoding was also introduced to pre-train image foundation models in a self-supervised manner, such as BEiT [3], SimMIM [79], and MAE [22]. Some vision-language pre-trained models, such as CLIP [55] and ALIGN [27], were also proposed by learning from the alignment between images and text on web-scale and noisy samples. These VL foundation models have shown excellent performance on zero-shot transfer.

Concerning video foundation models, their progress lags behind images, partially due to the relatively smaller video datasets and higher complexity of video modeling. Since the introduction of Kinetics benchmarks [28], some supervised pre-trained models on it have been transferred to small-scale datasets for action recognition, such as 2D CNNs (TSN [71], TSM [40], TANet [48], TDN [70]), 3D CNNs (I3D [7], R(2+1)D [64], ARTNet [69], SlowFast [19]), Transformer (TimeSformer [4], Video Swin [47],

UniFormer [35]). Recently, some self-supervised video models are developed based on masked autoencoding such as BEVT [72], MaskedFeat [76], VideoMAE [63], and MAE-ST [18] by directly extending these image masked modeling frameworks. However, these video foundation models often limit in their pre-training data size and model scale. More importantly, their downstream tasks have a narrow focus on action recognition, without consideration of other video tasks such as temporal action localization.

Masked visual modeling. Early works treated masking in denoised autoencoders [66] or context inpainting [52]. Inspired by the great success in NLP [6, 16], iGPT [9] operated pixel sequences for prediction and ViT [17] investigated the masked token prediction for self-supervised pre-training. Recently, there has been a surge of research into Transformer-based architectures for masked visual modeling [3, 18, 22, 63, 72, 76, 79]. BEiT [3], BEVT [72], and VIMPAC [60] learned visual representations by predicting discrete tokens. MAE [22] and SimMIM [79] directly performed pixel masking and reconstruction for pre-training without discrete token representation. MaskFeat [76] reconstructed the HOG [14] features of masked tokens to perform self-supervised pre-training in videos. VideoMAE [63] and MAE-ST [18] extended MAE [22] to video domain for self-supervised video pre-training and achieved impressive performance on action recognition.

Vision model scaling. Many works tried to scale up CNNs to improve recognition performance [24, 56, 59]. EfficientNet [62] presented a scaling strategy to balance depth, width, and resolution for CNN design. Several works [25, 29, 49] tried to train much larger CNNs to obtain excellent performance by enlarging model capacities and training data size. Recently, a few works [44, 84] tried to scale up the vision transformer to the billion-level models with large-scale supervised pre-training on JFT-3B [84] or self-supervised pre-training on IN-22K-ext-70M [44]. VideoMAE [63] and MAE-ST [18] have trained the huge video transformer with millions of parameters. MAE-ST [18] also tried the MAE pre-training on 1M IG-uncurated clips but failed to obtain better performance on Kinetics than small-scale pre-training. We are the first work to train video transformer with billion-level parameters.

3. VideoMAE V2

In this section, we first revisit VideoMAE and analyze its property. Then we present the dual masking strategy for the efficient training of VideoMAE. Finally, we present the scaling details of VideoMAE for large-scale pre-training.

3.1. VideoMAE Revisited

We scale the video masked autoencoder (VideoMAE) due to its simplicity and high performance. VideoMAE processes the downsampled frames $\mathbf{I} \in \mathbb{R}^{C\times T\times H\times W}$ from a clip with stride τ, and uses the cube embedding Φ_{emb} to transform the frames into a sequence of tokens. Then, it designs a customized tube masking strategy to drop tokens with an extremely high ratio ρ (e.g., 90%). Finally, the unmasked tokens are fed into a video autoencoder (Φ_{enc}, Φ_{dec}) for reconstructing the masked pixels. Specifically, VideoMAE is composed of *three core components*: cube embedding, encoder, and decoder. First, cube embedding encodes the local spatiotemporal features and builds the token list: $\mathbf{T} = \Phi_{emb}(\mathbf{I})$, where $\mathbf{T} = \{T_i\}_{i=1}^{N}$ is the token sequence, T_i is the token produced by the embedding layer and then added with positional embedding, and N is the total token number. Then the encoder simply operates on the *unmasked* tokens $\mathbf{T}^u$ with a vanilla ViT of joint space-time attention: $\mathbf{Z} = \Phi_{enc}(\mathbf{T}^u)$, where $\mathbf{T}^u$ represents the unmasked visible tokens $\mathbf{T}^u = \{T_i\}_{i\in(1-\mathbb{M}(\rho))}$, $\mathbb{M}(\rho)$ is the masking map, and its token length N^e is equal to $0.1N$. Finally, the decoder takes the *combined* tokens $\mathbf{Z}^c$ as inputs and performs reconstruction with another ViT: $\hat{\mathbf{I}} = \Phi_{dec}(\mathbf{Z}^c)$, where the combined tokens $\mathbf{Z}^c$ is the concatenated sequence of encoded token features $\mathbf{Z}$ and the learnable masked tokens [MASK] (with position embeddings), and its token length N^d is equal to the original token number N. The loss function is the mean squared error (MSE) loss between the normalized masked pixels and the reconstructed pixels: $\ell = \frac{1}{\rho N}\sum_{i\in\mathbb{M}(\rho)} |\mathbf{I}_i - \hat{\mathbf{I}}_i|^2$.

Computational cost analysis. High efficiency is an important characteristic of masked autoencoder. VideoMAE employs an asymmetric encoder-decoder architecture [22], where token sequence length of encoder is only one-tenth of decoder (i.e. $N^e = 0.1N^d$). This smaller encoder input contributes to more efficient pre-training pipeline compared with other masked autoencoding frameworks [3, 79]. However, when scaling VideoMAE in both depth and width (channels) to a billion-level model, the overall computation and memory consumption is still the bottleneck for the current available GPUs with limited memory. Therefore, the current asymmetric encoder-decoder architecture needs to be further improved for scaling VideoMAE.

3.2. Dual Masking for VideoMAE

To better enable large-scale VideoMAE pre-training under a limited computational budget, we present a dual masking scheme to further improve its pre-training efficiency. As shown in Figure 1, our dual masking scheme generates two masking maps $\mathbb{M}_e = \mathcal{M}_e(\rho^e)$ and $\mathbb{M}_d = \mathcal{M}_d(\rho^d)$ with two different masking generation strategies and masking ratios. These two masking maps $\mathbb{M}_e$ and $\mathbb{M}_d$ are for encoder and decoder, respectively. Like VideoMAE, our encoder operates on the partial and visible tokens under the encoder mask $\mathbb{M}_e$, and maps the observed tokens into latent feature representations. But unlike VideoMAE, our decoder takes

inputs from the encoder visible tokens and *part of the remaining* tokens visible under the decoder mask $\mathbb{M}_d$. In this sense, we use the decoder mask to reduce the decoder input length for high efficiency yet attain similar information to the full reconstruction. Our decoder maps the latent features and the remaining incomplete tokens into the pixel values at the corresponding locations. The supervision only applies to the decoder output tokens invisible to the encoder. We will detail the design next.

Masking decoder. As analyzed in Section 3.1, the decoder of VideoMAE is still inefficient as it needs to process all the cubes in videos. Thus, we further explore the prior of data redundancy in the decoder and propose the strategy of *masking decoder*. Our idea is mainly inspired by the recent efficient action recognition transformer [54], which only uses a small portion of tokens to achieve similar performance. It implies data redundancy exists in inference, which applies for our reconstruction target as well.

Our dual masking strategy is composed of encoder masking $\mathcal{M}_e$ and decoder masking $\mathcal{M}_d$. The encoder masking is the random tube masking with an extremely high ratio, which is the same as the original VideoMAE. For decoder masking, our objective is opposite to encoder masking. The tube masking in encoder tries to relieve the issue of "information leakage" caused by temporal correlation. In contrast, in decoder masking, we need to encourage "information complement" to ensure minimal information loss in this partial reconstruction. In this sense, we need to select as diverse cubes as possible to cover the whole video information. In the implementation, we compare different masking strategies and eventually choose the running cell masking [54]. With this decoder masking map $\mathbb{M}_d$ [1], we reduce the decoder input length to improve efficiency.

VideoMAE with dual masking. Our improved VideoMAE shares the same cube embedding and encoder with the original VideoMAE as described in Section 3.1. For decoder, it processes the combined tokens of encoder output and part the remaining visible tokens under the decoder mask $\mathbb{M}_d$. Specifically, the combined sequence is defined as:

$$\mathbf{Z}^c = \mathbf{Z} \cup \{\mathbf{M}_i\}_{i \in \mathbb{M}_d}, \tag{1}$$

where $\mathbf{Z}$ is the latent representation from encoder, $\mathbf{M}_i$ is the learnable masking token with corresponding positional embedding. With this combined token sequence $\mathbf{Z}^c$, our decoder only reconstructs the visible tokens under the decoder mask. The final MSE loss is computed between the normalized masked pixels $\mathbf{I}$ and the reconstructed ones $\hat{\mathbf{I}}$ over the decoder visible cubes:

$$\ell = \frac{1}{(1-\rho^d)N} \sum_{i \in \mathbb{M}_d \cap \mathbb{M}_e} |\mathbf{I}_i - \hat{\mathbf{I}}_i|^2. \tag{2}$$

[1]For a clear presentation, unlike encoder, we use this masking map to denote the kept and visible tokens in decoder input.

3.3. Scaling VideoMAE

Model scaling. Model scale is the primary force in obtaining excellent performance. Following the original VideoMAE, we use the vanilla ViT [17] as the backbone due to its simplicity. According to the scaling law of ViT [84], we build VideoMAE encoder with backbones of different capacities ranging from ViT-B, ViT-L, ViT-H, to ViT-g. Note that *ViT-g is a large model with billion-level parameters and has never been explored in video domain*. Its performance with masked autoencoding for video representation learning is still unknown to the community. More details on these backbone designs could be referred to [84]. For decoder design, we use relatively shallow and lightweight backbones [22, 63] with fewer layers and channels. In addition, we apply our dual masking strategy to further reduce computational cost and memory consumption. More details on the decoder design could be found in the appendix.

Data scaling. Data scale is another important factor that influences the performance of VideoMAE pre-training. The original VideoMAE simply pre-train the ViT models on relatively small-scale datasets by emphasizing its data efficiency. In addition, they require to pre-train the individual models specific to each dataset (i.e., Something-Something and Kinetics datasets have different pre-trained models). In contrast, we aim to learn a universal pre-trained model that could be transferred to different downstream tasks. To this end, we try to increase the pre-training video samples to a million-level size and aim to understand the data scaling property for VideoMAE pre-training. Data diversity is important for learning general video representations. Therefore, we build an *unlabeled hybrid* video dataset covering videos from General Webs, Youtube, Instagram, Movies, and Manual Recordings. We collect videos from the public datasets of Kinetics, Something-Something, AVA, WebVid, and uncurated videos crawled from Instagram. In total, there are 1.35M clips in our unlabeled mixed dataset. Note that *pre-training video transformer on a such large-scale and diverse dataset is rare in previous works* and *it still remains unknown the influence of data scale and diversity on VideoMAE pre-training*. More details on our dataset could be found in the appendix.

Progressive training. Transferring scheme is an important step to adapt the pre-trained large video transformers to the downstream tasks. The masked autoencoder pre-training is expected to learn some invariant features and can provide a favored initialization for vision transformer fine-tuning [30]. The original VideoMAE directly fine-tunes the pre-trained models on the target dataset only with its corresponding supervision. This direct adapting strategy might fail to fully unleash the power of large pre-trained video transformer due to limited supervision. Instead, in order to relieve the overfitting risk, we argue that *we should lever-*

age the semantic supervision signals from multiple sources in multiple stages to gradually adapt the pre-trained video transformers to downstream tasks. Accordingly, following the intermediate fine-tuning in images [3, 44], we devise a *progressive training* pipeline for the whole training process of billion-level video transformers. First, we conduct unsupervised pre-training with masked autoencoding on the unlabeled hybrid video dataset. Then, we build a *labeled hybrid* dataset by collecting and aligning multiple existing supervised datasets with labels. We perform the supervised post-pre-training stage on this labeled hybrid dataset to incorporate the semantics from multiple sources into the previous pre-trained video transformers. Finally, we perform the *specific fine-tuning* stage on the target dataset to transfer the general semantics to the task-centric knowledge.

Based on the above designs of dual masking, data scaling, and progressive training, we implement a simple and efficient masked autoencoding framework with a billion-level ViT backbone, termed as *VideoMAE V2*. With this new framework, we successfully train the first billion-level video transformer and push the vanilla ViT performance limit on a variety of video downstream tasks, including video action recognition, action detection, and temporal action detection.

4. Experiments

4.1. Implementation and Downstream Tasks

Model. We conduct investigations on the VideoMAE V2 by scaling its model capacity and pre-training data size. We scale the backbone network from the existing huge ViT model (ViT-H) to the giant ViT model (ViT-g) [84]. The ViT-g has a smaller patch size (14), more encoder blocks (40), a higher dimension of cube embedding and self-attention (1408), and more attention heads (16). It has 1,011M parameters. More details could be referred to [84].

Data. To well support the billion-level ViT model pre-training, we build two large-scale video datasets for our proposed progressive training. For self-supervised pre-training of VideoMAE V2, we build a million-level unlabeled video dataset by collecting clips from multiple resources such as Movie, Youtube, Instagram, General Webs, and manual recordings from scripts, and the dataset is termed as *UnlabeledHybrid*. Specifically, our dataset is built by simply selecting videos from the public available datasets of Kinetics [28], Something-Something [20], AVA [21], WebVid2M [2], and our own crawled Instagram dataset. In total, there are around 1.35M clips in our mixed dataset and this is the largest dataset ever used for video masked autoencoding. For supervised post-pre-training, we collect the larger video dataset with human annotations, termed as *LabeledHybrid*. Following [36], we take the union of different versions of Kinetics datasets (K400, K600, K700) by aligning their label semantics and removing the duplicate videos with the validation sets. This labeled hybrid dataset has 710 categories and 0.66M clips. We pre-train our video transformer model on these two datasets and then transfer them to the downstream tasks as detailed next. More details on these pre-training datasets could be found in the appendix.

Tasks. To verify the generalization ability of VideoMAE V2 pre-trained ViTs as video foundation models, we transfer their representations to a variety of downstream tasks.

Video Action Classification. Action classification is the most common task in video understanding. Its objective is to classify each trimmed clip into a predefined action class and evaluated the average accuracy over action classes. According to the original VideoMAE [63], we perform detailed analysis on this task to investigate the property of scaling video masked autoencoding. In experiments, we choose four datasets to report its performance: Kinetics [28], Something-Something [20], UCF101 [57], and HMDB51 [32]. Kinetics and Something-Something are two large-scale action recognition datasets and have their own unique property for action recognition, where Kinetics contains appearance-centric action classes while Something-Something focuses on motion-centric action understanding. UCF101 and HMDB51 are two relatively small datasets and suitable to verify the transfer performance of large pre-trained models as shown in the appendix.

Spatial Action Detection. Action detection is an important task in video understanding, and it aims to recognize all action instances and localize them in space. This task is more challenging than action classification as it deals with more fine-grained action classes and needs to capture detailed structure information to discriminate co-occurring action classes. In experiments, we choose two action detection benchmarks to illustrate the effectiveness of our pre-trained models by VideoMAE V2, namely AVA [21] and AVA-Kinetics [34]. AVA contains the box annotations and their corresponding action labels on keyframes (could be more than one label for each human box). The annotations are done at 1FPS over 80 atomic classes. AVA-Kinetics introduces the AVA style annotations to the Kinetics dataset and a single frame of selected video from Kinetics is annotated with AVA labels. The evaluation metric is frame-level Average Precision (mAP) under the IoU threshold of 0.5.

Temporal Action Detection. Temporal action detection is an important task in long-form video understanding. Its goal is to recognize all action instances in an untrimmed video and localize their temporal extent (starting and ending timestamps). Unlike spatial action detection, temporal action localization aims to focus on precise temporal boundary localization. Intuitively, in addition to capturing semantic information for recognition, the pre-trained models should be able to effectively model the temporal evolution of features to detect action boundaries. In experiments, we choose two temporal action detection benchmarks to

Decoder Masking	ρ^d	Top-1	FLOPs
None	0%	**70.28**	35.48G
Frame	50%	69.76	25.87G
Random	50%	64.87	25.87G
Running cell [1]	50%	66.74	25.87G
Running cell [2]	25%	70.22	31.63G
Running cell [2]	50%	70.15	25.87G
Running cell [2]	75%	70.01	21.06G

Table 1. **Ablation study on the decoder masking strategies.** Experiments are conducted with ViT-B by pre-training on SSv2 with 800 epochs. "None" refers to the original VideoMAE without decoder masking. We use a better fine-tuning setting than the original VideoMAE. [1] Loss computed over all decoder output tokens. [2] Loss computed over only decoder output tokens invisible to encoder. The default setting for VideoMAE v2 is colored in gray.

evaluate the performance of our pre-trained video models: THUMOS14 [43] and FineAction [43]. THUMOS14 is a relatively small and well labeled temporal action detection dataset, that has been widely used by the previous methods. It only includes sports action classes on this dataset. FineAction is a new large-scale temporal action dataset with fine-grained action class definitions. The evaluation metric is the average mAP under different tIoU thresholds.

4.2. Main Results

We first conduct the experimental study on the core designs of our VideoMAE V2 pre-training framework. We report the fine-tuning action recognition accuracy on the datasets of Kinetics-400 and Something-Something (Sth-Sth) V2. Implementation details about the pre-training and fine-tuning could be found in the appendix.

Results on dual masking. We first perform an ablation study on the decoder masking strategy. In this study, we use the ViT-B as the backbone and the pre-training is performed on the Sth-Sth V2 dataset with 800 epochs. The results are evaluated with the fine-tuning accuracy on the Sth-Sth V2 and reported in Table 1. We first re-implement the original VideoMAE without decoder masking and achieve slightly better performance (70.28% vs. 69.6% Top-1 acc.). Then, we try two masking alternatives in decoder: frame masking and random masking. For frame masking, we only reconstruct half of the frames in the decoder, and for random masking, we stochastically drop half of the cubes in the decoder for reconstruction. These two alternatives perform worse than the original VideoMAE. Finally, we apply the running cell masking to select a subset of representative cubes for reconstruction. In this setting, we apply loss functions computed over all decoder output tokens or only encoder-invisible tokens. Agreed with the results in MAE and VideoMAE, the loss on all tokens performs worse partially due to information leakage of these visible tokens from encoder. The running cell masking scheme performs on par with the original result (70.28% vs. 70.15% top-1 acc.). We also ablates the decoder masking ratio and 50% keeps a good trade-off between accuracy and efficiency.

In Table 2, we report the computational cost (FLOPs), memory consumption (Mems), and per-epoch running time (Time) of dual masking, and compare with the original encoder-only masking in VideoMAE. In this comparison, we use a lightweight decoder only with 4 transformer blocks (channel 384 for ViT-B and 512 for ViT-g). Our dual masking can further improve the computational efficiency of the original asymmetric encoder-decoder architecture in MAE [22]. For memory consumption, we can reduce almost half of the overall memory of feature maps, and this is particularly important for pre-training billion-level video transformer under the available GPUs of limited memory.

Results on data scaling. We study the influence of pre-training data size on the VideoMAE V2 pre-training. In this experiment, we pre-train the video models with backbones from ViT-B, ViT-L, ViT-H, to ViT-g on our built Unlabeled-Hybrid dataset with around 1.35M videos. The fine-tuning accuracy is shown in Table 3 on the Kinetics-400 and Table 4 on the Sth-Sth V2. We first compare our performance with the original VideoMAE pre-training. We find that for all backbones, our large-scale pre-training obtains better performance than the original VideoMAE pre-trained on the small-scale datasets of Kinetics-400 or Sth-Sth V2. Meanwhile, we see that the performance gap between two pre-training data datasets becomes more evident as the modal capacity scales up. In particular, on the Sth-Sth V2 dataset, our pre-trained ViT-H outperforms the original VideoMAE by 2.0%. It implies that data scale is also important for video masked autoencoding. Meanwhile, we compare with MAE-ST of IG-uncurated pre-training (1M clips). With the same ViT-L backbone, our pre-trained model outperforms it by 1% on the Kinetics-400 dataset. This result shows that data quality and diversity might be another important factor.

Results on model scaling. We study the performance trend with different model capacities. We compare the fine-tuning performance of pre-trained models with ViT-B, ViT-L, ViT-H, and ViT-g as shown in Table 3 and Table 4. Vit-g is the first billion-level model pre-trained in video domain. We obtain consistent performance improvement with increasing model capacity. For all compared pre-training methods, the performance improvement from ViT-B to ViT-L is more obvious, while the improvement from ViT-L to ViT-H is much smaller. We further scale up the model capacity to ViT-g architecture. We can still boost the fine-tuning performance further on these two benchmarks with a smaller improvement. We also notice that the performance gap between huge and giant model is very small (0.1%-0.2%) in images [80]. We analyze the performance seems to saturate around 87.0 on the Kinetics-400 and 77.0 on the Sth-Sth V2 for methods without using any extra labeled data.

Masking	Backbone	pre-training dataset	FLOPs	Mems	Time	Speedup	Top-1
Encoder masking	ViT-B	Something-Something V2	35.48G	631M	28.4h	-	70.28
Dual masking	ViT-B	Something-Something V2	25.87G	328M	15.9h	**1.79**×	70.15
Encoder masking	ViT-g	UnlabeledHybrid	263.93G	1753M	356h[1]	-	-
Dual masking	ViT-g	UnlabeledHybrid	241.61G	1050M	241h	**1.48**×	77.00

Table 2. **Comparison between dual masking and encoder-only masking**. We report the computational cost, memory consumption, and running time for comparison. We perform experiments with backbones (ViT-B and ViT-g) and pre-training on two scales of datasets (Sth-Sth V2 and UnlabeledHybrid). Time is for 1200 epochs on 64 GPUs. [1] is estimated by training 5 epochs.

method	pre-train data	data size	epoch	ViT-B	ViT-L	ViT-H	ViT-g
MAE-ST [18]	Kinetics400	0.24M	1600	81.3	84.8	85.1	-
MAE-ST [18]	IG-uncurated	1M	1600	-	84.4	-	-
VideoMAE V1 [63]	Kinetics400	0.24M	1600	**81.5**	85.2	86.6	-
VideoMAE V2	UnlabeledHybrid	1.35M	1200	**81.5** (77.0)	**85.4** (81.3)	**86.9** (83.2)	**87.2** (83.9)
$\Delta Acc.$ with V1	-	-	-	***+ 0%***	***+ 0.2%***	***+ 0.3%***	-

Table 3. **Results on the Kinetics-400 dataset**. We scale the pre-training of VideoMAE V2 to billion-level ViT-g model with million-level data size. We report the fine-tuning accuracy of multiple view fusion (5×3) and single view results in the bracket. All models are pre-trained and fine-tuned at the input of 16×224×224 and sampling stride $\tau = 4$.

method	pre-train data	data size	epoch	ViT-B	ViT-L	ViT-H	ViT-g
MAE-ST [18]	Kinetics400	0.24M	1600	-	72.1	74.1	-
MAE-ST [18]	Kinetics700	0.55M	1600	-	73.6	75.5	-
VideoMAE V1 [63]	Something-Something V2	0.17M	2400	70.8	74.3	74.8	-
VideoMAE V2	UnlabeledHybrid	1.35M	1200	**71.2** (69.5)	**75.7** (74.00)	**76.8** (75.5)	**77.0** (75.7)
$\Delta Acc.$ with V1	-	-	-	***+ 0.4%***	***+ 1.4%***	***+ 2.0%***	-

Table 4. **Results on the Something-Something V2 dataset**. We scale the pre-training of VideoMAE V2 to billion-level ViT-g model with million-level data size. We report the fine-tuning accuracy of multiple view fusion (2×3) and single view results in the brackets. All models are pre-trained at input of 16×224×224 and sampling stride $\tau = 2$. Fine-tuning is on the same size as TSN [71] sampling.

method	extra supervision	ViT-H	ViT-g
MAE-ST [18]	K600	86.8	-
VideoMAE V1 [63]	K710	88.1 (84.6)	-
VideoMAE V2	-	86.9 (83.2)	87.2 (83.9)
VideoMAE V2	K710	**88.6** (85.0)	**88.5** (85.6)
$\Delta Acc.$ with V1	K710	***+ 0.5%***	-

Table 5. **Study on progressive pre-training**. We report the fine-tuning accuracy of multiple view fusion (5×3) and single view results in the bracket on the Kinetics-400 dataset. The implementation detail is the same with Table 3.

Results on progressive training. We study the influence of the post-pre-training step in our progressive training scheme. To mitigate over-fitting risk and integrate more human supervision into our pre-trained video models, we merge different versions of Kinetics for post-pre-training (intermediate fine-tuning) and evaluate on the Kinetics-400 dataset. The results are reported in Table 5. We observe that the post-pre-training boosts the performance of large-scale pre-trained models for both ViT-H and ViT-g. This result agrees with the findings in image domain [3, 44]. We also apply this technique to VideoMAE V1 pre-trained model and it (ViT-H) achieves worse performance (88.1% vs. 88.6%), demonstrating the effectiveness of large-scale unsupervised pre-training. We also compare with the intermediate fine-tuning of MAE-ST and our performance is better by 1.8% with the same ViT-H backbone. This superior performance might be ascribed to the larger unsupervised pre-training dataset and larger intermediate fine-tuning dataset. We also try this post-pre-training on Sth-Sth V2 dataset but obtain worse performance.

4.3. Performance on Downstream Tasks

To demonstrate the generalization ability of our pre-trained models, we transfer their representations to a variety of downstream tasks. In total, we study three kinds of tasks and report performance on ten mainstream benchmarks.

Action classification. We compare with previous state-of-the-art methods on four action recognition benchmarks Kinetics-400/600 and Something-Something V1/V2. On the Kinetics datasets, our VideoMAE V2 achieves the best performance among these methods without using extra in-house labeled data, and quite competitive performance to the leading performance MTV [82] trained with extra 60M labeled videos. On the Something-Something datasets, our models significantly outperform the previous best performance, and in particular on Something-Something V1, the performance improvement is above 7%. The fine-tuning results on UCF101 and HMDB51 are shown in the appendix.

Spatial action detection. We perform a comparison with the previous action detection methods on two datasets: AVA and AVA Kinetics. We follow the same detection pipeline with the original VideoMAE [63]. On the AVA dataset, our pre-trained model is significantly better than the previous state-of-the-art action detector of TubeR [88] and outperforms the previous masked modeling methods by 3.1%. On the AVA-Kinetics, we compare the previous challenge winner methods with the model ensemble, and our single model is better than the best performance by 3.4%.

Temporal action detection. We investigate the transfer

(a) Kinetics 400

Method	Top 1	Top 5	Views	TFLOPs
I3D NL [74]	77.7	93.3	10 × 3	10.77
TDN [70]	79.4	94.4	10 × 3	5.94
SlowFast R101-NL [19]	79.8	93.9	10 × 3	7.02
TimeSformer-L [4]	80.7	94.7	1 × 3	7.14
MTV-B (320^2) [82]	82.4	95.2	4 × 3	11.16
Video Swin-L (384^2) [47]	84.9	96.7	10 × 5	105.35
ViViT-L FE [1]	81.7	93.8	1 × 3	11.94
MViTv2-L (312^2) [38]	86.1	97.0	40 × 3	42.42
MaskFeat [76]	87.0	97.4	4 × 3	45.48
MAE-ST [18]	86.8	97.2	4 × 3	25.05
VideoMAE [63]	86.6	97.1	5 × 3	17.88
VideoMAE V2-H	88.6	97.9	5 × 3	17.88
VideoMAE V2-g	88.5	98.1	5 × 3	38.16
VideoMAE V2-g (64×266^2)	**90.0**	**98.4**	2 × 3	160.30
Methods using in-house labeled data				
CoVeR (JFT-3B) [85]	87.2	–	1 × 3	-
MTV-H (WTS 280^2) [82]	89.9	98.3	4 × 3	73.57

(b) Kinetics 600

Method	Top 1	Top 5	Views	TFLOPs
SlowFast R101-NL [19]	81.8	95.1	10 × 3	7.02
TimeSformer-L [4]	82.2	95.6	1 × 3	7.14
MTV-B (320^2) [82]	84.0	96.2	4 × 3	11.16
ViViT-L FE [1]	82.9	94.6	1 × 3	11.94
MViTv2-L (352^2) [38]	87.9	97.9	40 × 3	45.48
MaskFeat [76]	86.4	97.4	1 × 10	3.77
VideoMAE V2-H	88.3	98.1	5 × 3	17.88
VideoMAE V2-g	88.8	98.2	5 × 3	38.16
VideoMAE V2-g (64×266^2)	**89.9**	**98.5**	2 × 3	160.30
Methods using in-house labeled data				
CoVeR (JFT-3B) [85]	87.9	97.8	1 × 3	-
MTV-H (WTS 280^2)	**90.3**	**98.5**	4 × 3	73.57

(c) Something-Something V2

Method	Top 1	Top 5
SlowFast [19]	63.1	87.6
TEINet [46]	66.5	-
TEA [37]	65.1	89.9
TDN [70]	69.6	92.2
TimeSformer-L [4]	62.4	-
MFormer-HR [53]	68.1	91.2
ViViT-L FE [1]	65.9	89.9
Video Swin-B [47]	69.6	92.7
MViTv2-B [38]	72.1	93.4
MTV-B [82]	67.6	90.1
BEVT [72]	70.6	-
VIMPAC [60]	68.1	-
UniFormer [35]	71.2	92.8
MaskFeat [76]	75.0	95.0
MAE-ST [18]	75.5	95.0
VideoMAE [63]	75.4	95.2
VideoMAE V2-H	76.8	95.8
VideoMAE V2-g	**77.0**	**95.9**

(d) Something-Something V1

Method	Top 1	Top 5
I3D [7]	41.6	72.2
NL I3D+GCN [75]	46.1	76.8
TSM [40]	49.7	78.5
V4D [87]	50.4	-
TANet [48]	50.6	79.3
TEINet [46]	52.5	-
TEA [37]	51.9	80.3
CorrNet [68]	53.3	-
GSM [58]	55.2	-
TDN [70]	56.8	84.1
UniFormer [35]	61.0	87.6
VideoMAE V2-H	66.6	90.8
VideoMAE V2-g	**68.7**	**91.9**

(e) AVA

Method	Long Feature	mAP
SlowFast [19]	✗	29.0
TubeR [88]	✓	33.4
MaskFeat [76]	✗	38.8
MAE-ST [18]	✗	39.0
VideoMAE [63]	✗	39.5
VideoMAE V2	✗	**42.6**

(f) AVA Kinetics

Method	Ensembled	mAP
AIA++ [78]	✓	29.0
MSF [89]	✓	33.4
ACAR [51]	✓	40.5
VideoMAE V2	✗	**43.9**

(g) THUMOS14

Method	Optical Flow	mAP
RTD-Net [61]	✓	43.6
DaoTAD [67]	✗	50.0
AFSD [39]	✓	52.0
DCAN [8]	✓	52.3
TadTR [42]	✓	54.2
TALLFormer [12]	✗	59.2
BasicTAD [83]	✗	59.6
ActionFormer [86]	✓	66.8
VideoMAE V2	✗	**69.6**

(h) FineAction

Method	Optical Flow	mAP
BMN [41]	✓	9.25
G-TAD [81]	✓	9.06
BasicTAD [83]	✗	12.2
ActionFormer [86]	✗	13.2
VideoMAE V2	✗	**18.2**

Table 6. **Systematic study on the transfer performance of VideoMAE V2 pre-trained models.** We use them as the video foundation models and transfer them to three kinds of downstream tasks: action classification, action detection, and temporal action detection, covering eight mainstream video action benchmarks. Entries using extra in-house labeled data for training are in gray. "-": numbers not available.

performance of our model to the task of temporal action detection. This is an important task yet not to be tested by previous masked pre-training methods. We report performance on two benchmarks: THUMOS14 and FineAction. We use the ActionFormer [86] detection pipeline as our baseline method and replace its I3D feature with our pre-trained representation. On the THUMOS14 dataset, our model outperforms all previous results even with optical flow as input. On the large-scale FineAction dataset, our model is significantly better than a previous best performance by 5%.

5. Conclusion and Discussion

Building foundation model has turned out to be an effective paradigm to improve the performance of tasks in AI. Simple and scalable algorithms are the core of building powerful foundation models. In this paper, we have presented a simple and efficient way to scale VideoMAE to billion-level model on the million-level pre-training set. Thanks to our core design of dual masking, we are able to successfully the first billion-level video transformer, and demonstrate its effectiveness on a variety of video downstream tasks. Our work shows that video masked autoencoders are general and scalable representation learners for video action understanding. We hope our pre-trained models could provide effective representations for more video understanding tasks in the future.

In spite of these excellent results, challenge still remains. We observe that the performance improvement is smaller when we scale VideoMAE from ViT-H to ViT-g, partially because of performance saturation on these video benchmarks. However, the data scale we have explored for VideoMAE is still several orders of magnitudes smaller than the Image [44, 84] and NLP [6, 16]. How to train VideoMAE on billions of videos is still extremely challenging for the current software and hardware. We need to come up with more efficient video pre-training framework and hope our work can inspire future work on scaling video pre-training.

Acknowledgements. This work is supported by the National Key R&D Program of China (No. 2022ZD0160900, No.2022ZD0160100), the National Natural Science Foundation of China (No. 62076119, No. 61921006), Shanghai Committee of Science and Technology (Grant No. 21DZ1100100), the Youth Innovation Promotion Association of Chinese Academy of Sciences (No. 2020355).

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6816–6826, 2021. 8

[2] 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. 5

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: BERT pre-training of image transformers. In *ICLR*, 2022. 1, 2, 3, 5, 7

[4] 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. 2, 8

[5] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. 1, 2

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. 1, 2, 3, 8

[7] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733, 2017. 2, 8

[8] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. DCAN: improving temporal action detection via dual context aggregation. In *AAAI*, pages 248–257, 2022. 8

[9] 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

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 2

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pages 15750–15758, 2021. 2

[12] Feng Cheng and Gedas Bertasius. Tallformer: Temporal action localization with a long-memory transformer. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *ECCV*, pages 503–521, 2022. 8

[13] Dading Chong, Helin Wang, Peilin Zhou, and Qingcheng Zeng. Masked spectrogram prediction for self-supervised audio pre-training. *CoRR*, abs/2204.12768, 2022. 1

[14] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 3

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *NAACL-HLT*, pages 4171–4186, 2019. 1, 2, 3, 8

[17] 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. 2, 3, 4

[18] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *NeurIPS*, 2022. 1, 3, 7, 8

[19] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6201–6210, 2019. 2, 8

[20] 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. 2, 5

[21] 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. 2, 5

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 15979–15988, 2022. 1, 2, 3, 4, 6

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735, 2020. 2

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 3

[25] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Xu Chen, HyoukJoong Lee, Jiquan

Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, pages 103–112, 2019. 3

[26] 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. 2

[27] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916, 2021. 1, 2

[28] 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. 2, 5

[29] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, pages 491–507, 2020. 3

[30] Xiangwen Kong and Xiangyu Zhang. Understanding masked image modeling via learning occlusion invariant feature. *CoRR*, abs/2208.04164, 2022. 2, 4

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 2

[32] 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

[33] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. 2

[34] Ang Li, Meghana Thotakuri, David A. Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *CoRR*, abs/2005.00214, 2020. 2, 5

[35] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *ICLR*, 2022. 3, 8

[36] 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. 5

[37] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. TEA: temporal excitation and aggregation for action recognition. In *CVPR*, pages 906–915, 2020. 8

[38] 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. 8

[39] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *CVPR*, pages 3320–3329, 2021. 8

[40] Ji Lin, Chuang Gan, and Song Han. TSM: temporal shift module for efficient video understanding. In *ICCV*, pages 7082–7092, 2019. 2, 8

[41] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3888–3897, 2019. 8

[42] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *IEEE Trans. Image Process.*, 31:5427–5441, 2022. 8

[43] 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. 2, 6

[44] 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. In *CVPR*, pages 11999–12009, 2022. 1, 2, 3, 5, 7, 8

[45] 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 *ICCV*, pages 9992–10002, 2021. 2

[46] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *AAAI*, pages 11669–11676, 2020. 8

[47] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3192–3201, 2022. 2, 8

[48] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. TAM: temporal adaptive module for video recognition. In *ICCV*, pages 13688–13698, 2021. 2, 8

[49] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, pages 185–201, 2018. 3

[50] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation. *CoRR*, abs/2204.12260, 2022. 1

[51] 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*, pages 464–474, 2021. 8

[52] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 3

[53] Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea

Vedaldi, and João F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *NeurIPS*, pages 12493–12506, 2021. 8

[54] Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Xiang Wang, Yuehuan Wang, Yiliang Lv, Changxin Gao, and Nong Sang. MAR: masked autoencoders for efficient action recognition. *CoRR*, abs/2207.11660, 2022. 4

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 1, 2

[56] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 3

[57] 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

[58] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *CVPR*, pages 1099–1108, 2020. 8

[59] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 3

[60] 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. 3, 8

[61] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, pages 13506–13515, 2021. 8

[62] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *ICML*, pages 6105–6114, 2019. 3

[63] 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. 1, 2, 3, 4, 5, 7, 8

[64] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 2

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 1, 2

[66] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008. 3

[67] Chenhao Wang, Hongxiang Cai, Yuxin Zou, and Yichao Xiong. RGB stream is enough for temporal action detection. *CoRR*, abs/2107.04362, 2021. 8

[68] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *CVPR*, pages 349–358, 2020. 8

[69] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, pages 1430–1439, 2018. 2

[70] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. TDN: temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021. 2, 8

[71] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016. 2, 7

[72] 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*, pages 14713–14723, 2022. 3, 8

[73] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 548–558, 2021. 2

[74] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 8

[75] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, pages 413–431, 2018. 8

[76] 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. 1, 3, 8

[77] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 2

[78] Jin Xia, Wei Li, Jie Shao, Zehuan Yuan, Jiajun Tang, Cewu Lu, and Changhu Wang. Multiple attempts for ava-kinetics challenge 2020 https://static.googleusercontent.com/media/research.google.com/es//ava/2020/ByteDance-SJTU_AVA_report_2020.pdf, 2020. 8

[79] 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 *CVPR*, pages 9643–9653, 2022. 1, 2, 3

[80] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Yixuan Wei, Qi Dai, and Han Hu. On data scaling in masked image modeling. *CoRR*, abs/2206.04664, 2022. 2, 6

[81] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali K. Thabet, and Bernard Ghanem. G-TAD: sub-graph localization for temporal action detection. In *CVPR*, pages 10153–10162, 2020. 8

[82] 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. 7, 8

[83] Min Yang, Guo Chen, Yin-Dong Zheng, Tong Lu, and Limin Wang. Basictad: an astounding rgb-only baseline for temporal action detection. *CoRR*, abs/2205.02717, 2022. 8

[84] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, pages 1204–1213, 2022. 2, 3, 4, 5, 8

[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. *CoRR*, abs/2112.07175, 2021. 8

[86] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *ECCV*, pages 492–510, 2022. 8

[87] Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R. Scott, and Limin Wang. V4D: 4d convolutional neural networks for video-level representation learning. In *ICLR*, 2020. 8

[88] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, Ivan Marsic, Cees G. M. Snoek, and Joseph Tighe. Tuber: Tubelet transformer for video action detection. In *CVPR*, pages 13588–13597, 2022. 7, 8

[89] Xiantan Zhu, Xuan Tao, Lu Shi, Shaoqi Chen, Rui Yin, Lan Ding, Yuya Obinata, Takuma Yamamoto, and Zhiming Tan. Multi-scale spatiotemporal features for action localization. https://static.googleusercontent.com/media/research.google.com/es//ava/2020/Multi-scale_Spatiotemporal_Features_for_Action_Localization.pdf, 2020. 8