# *Ada*MAE: Adaptive Masking for Efficient Spatiotemporal Learning with Masked Autoencoders

Wele Gedara Chaminda Bandara[1], Naman Patel[2], Ali Gholami[2], Mehdi Nikkhah[2], Motilal Agrawal[2], and Vishal M. Patel[1]

[1]Johns Hopkins University [2]Zippin

## Abstract

*Masked Autoencoders (MAEs) learn generalizable representations for image, text, audio, video, etc., by reconstructing masked input data from tokens of the visible data. Current MAE approaches for videos rely on random patch, tube, or frame based masking strategies to select these tokens. This paper proposes AdaMAE, an adaptive masking strategy for MAEs that is end-to-end trainable. Our adaptive masking strategy samples visible tokens based on the semantic context using an auxiliary sampling network. This network estimates a categorical distribution over spacetime-patch tokens. The tokens that increase the expected reconstruction error are rewarded and selected as visible tokens, motivated by the policy gradient algorithm in reinforcement learning. We show that AdaMAE samples more tokens from the high spatiotemporal information regions, thereby allowing us to mask 95% of tokens, resulting in lower memory requirements and faster pre-training. We conduct ablation studies on the Something-Something v2 (SSv2) dataset to demonstrate the efficacy of our adaptive sampling approach and report state-of-the-art results of 70.0% and 81.7% in top-1 accuracy on SSv2 and Kinetics-400 action classification datasets with a ViT-Base backbone and 800 pre-training epochs. Code and pre-trained models are available at:* `https://github.com/wgcban/adamae.git`.

## 1. Introduction

Self-supervised learning (SSL) aims to learn transferable representations from a large collection of unlabeled data for downstream applications (e.g., classification and detection). SSL is conducted in a two-stage framework [21], consisting of pre-training on an unlabeled dataset, and fine-tuning on a downstream task. Pre-training has shown to improve performance [3, 11], convergence speed [11], and robustness [2, 18], and reduce model overfitting [11, 14] on downstream tasks.

Recently, masked autoencoders (MAEs) [6, 11, 16, 17, 20, 48, 57] and contrastive learning [21, 39, 47] approaches are

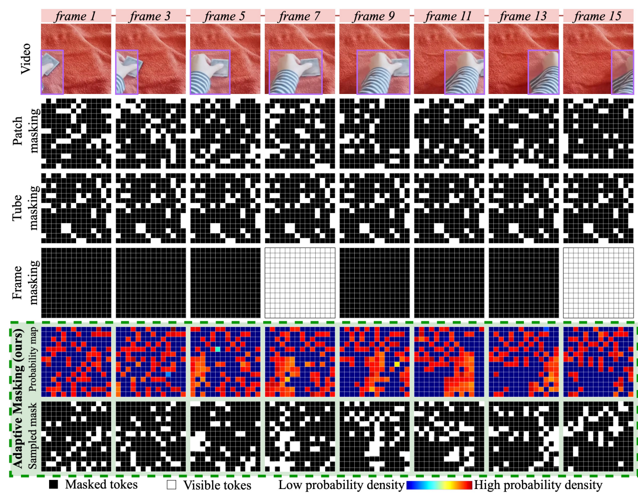



Figure 1. **Comparison of our *adaptive masking* with existing random *patch* [11], *tube* [48, 52], and *frame* [27, 37, 42] masking for masking ratio of 80%.** Our adaptive masking approach selects more tokens from the regions with *high spatiotemporal information* while a small number of tokens from the background.

mainly used for SSL. In *MAEs*, the input (image or video) is patchified and converted into a set of tokens. A small percentage (*e.g.*, 5-10%) of these tokens, namely visible tokens, are passed through a Vision Transformer (ViT) [8]. The resulting token embeddings are then concatenated with a learnable representation for masked tokens, and are fed into a shallow decoder transformer to reconstruct masked patches. On the other hand, *contrastive learning* takes two augmented views of the same input and pulls them together in the embedding space, while embeddings of different inputs are pushed away [39]. MAEs have recently gained more attention over contrastive learning methods due to the inherent use of a high masking ratio, which enables simple and memory-efficient training.

Mask sampling techniques are critical to the success of MAEs [11, 54]. Previous studies have investigated different sampling techniques that include random "patch" [11], "tube" [54], and "frame" [54] masking (see Fig. 1). Random patch sampling has shown to work well compared to its counterparts in some cases [11]. However, since not all tokens have equal information, assuming a uniform probability distribution over all input tokens (for selection of visible tokens) is sub-optimal. In other words, with these random masking strategies, the visible tokens are sampled from redundant or low information regions instead of high information ones, hence resulting in inaccurate reconstructions. This inhibits MAEs from learning meaningful representations, besides requiring a relatively larger number of training iterations compared to contrastive learning methods.

In this paper, we propose an *adaptive sampling* approach that simultaneously optimizes an MAE and an adaptive token sampling network. Our approach selects patches based on their *spatiotemporal information*. Unlike uniform random sampling, we first estimate the categorical distribution over all input tokens using an auxiliary network, and then sample visible tokens from that distribution. Since sampling is a non-differentiable operation, we propose an auxiliary loss for optimizing the adaptive token sampling network. Our solution is motivated by the REINFORCE algorithm [56], which comes under the family of policy gradient algorithms in Reinforcement Learning (RL) [23].

We empirically show that our adaptive token sampling network leads to sampling more tokens from high spatiotemporal information regions compared to random masking techniques as shown in Fig. 1. This efficient token allocation also enables high masking ratios (i.e., 95%) for pre-training MAEs. This ultimately reduces the GPU memory requirements and expedites pre-training while improving accuracy on downstream tasks. In summary, our contributions are:

- We propose *Ada*MAE, a novel, adaptive, and end-to-end trainable token sampling strategy for MAEs that takes into account the spatiotemporal properties of all input tokens to sample fewer but informative tokens.
- We empirically show that *Ada*MAE samples more tokens from high spatiotemporal information regions of the input, resulting in learning meaningful representations for downstream tasks.
- We demonstrate the efficiency of *Ada*MAE in terms of performance and GPU memory against random "patch", "tube", and "frame" sampling by conducting a thorough ablation study on the SSv2 dataset.
- We show that our *Ada*MAE outperforms state-of-the-art (SOTA) by 0.7% and 1.1% (in top-1) improvements on SSv2 and Kinetics-400, respectively.

# 2. Related Work

## 2.1. Masked prediction

Masked prediction has been widely successful in natural language understanding (e.g. Generative Pre-Training (GPT) [40] and bidirectional encoder (BERT) [26]). Many researchers have investigated applying masked prediction to representation learning for images and videos. Generative pre-training from pixels (iGPT) [7] is the first proof of concept in this direction, where masked *pixel* prediction is performed. However, this pixel-based method has a high pre-training computation cost and performs worse compared to ConvNets. With the introduction of the Vision Transformer (ViT) [8], the focus shifted from pixels to patches. The concept of patches enables the ViTs to follow the masked language modeling task in BERT by predicting masked image patches (called iBERT) for visual pre-training.

Although initial ViT experiments for representation learning have shown inferior performance compared to contrastive learning methods [22, 39, 64, 65], subsequent developments such as BERT pre-training including BEiT [4], BEVT [52], and masked auto-encoders (MAEs) including MaskFeat [54], spatiotemporal-learner [17], SimMIM [57], MFM [57], VideoMAE [48], ConvMAE [13], Omni-MAE [14], MIMDET [10], paved the way for superior performance and lower pre-training times.

Since image patches do not have off-the-shelf tokens as words in languages, BEiT [4] and BEVT [52] adopt a two-stage pre-training strategy, where an image/video tokenizer is trained via a discrete variational auto-encoder (dVAE) [41] followed by masked patch prediction using the pre-trained tokenizer [63]. This two-stage training and the dependence on a pre-trained dVAE to generate originally continuous but intentionally discretized target visual tokens leaves room for improving the efficiency. On the other hand, MAEs [4, 10, 14, 17, 24, 28, 48, 57, 61] directly predict the masked patches from the visible tokens for pre-training. Moreover, experiments suggest that a high masking ratio (75% for images [4, 17] and 90% for videos [48, 57]) leads to better fine-tuning performance. The encoder of MAEs only operates on the visible patches and the decoder is lightweight for faster pre-training. This results in MAE pre-training which is three times faster than BEiT [11, 17, 55] and BEVT [52].

## 2.2. Mask sampling

Various ablation experiments of masked autoencoders on both images and videos have shown that the performance on downstream tasks depends on the masking strategy [11, 48, 57]. Various random masking strategies have been proposed including *grid*, *block*, and *patch* for images [17, 58] and *tube*, *frame* and *patch* for videos [11, 27, 37, 42, 57]. Experiments have shown that a mask sampling technique that works well on one dataset might not be ideal for another dataset. For instance, Video-MAE [48] achieves the best action classification performance on the SSv2 [15] dataset with random tube masking, whereas SpatiotemporalMAE [11] achieves its best performance on Kinetics-400 [25] with random patch masking. This could be due to the diversity of underlying scenes in the dataset, different acquisition conditions (*e.g.*, frame rate), and high/low spatiotemporal information regions in videos. Even within a dataset, we observe significant diversity among the videos, where some videos have dynamic scenes, some have a stationary background and moving foreground (see Fig. 1), and some videos have almost stationary scenes (no object is moving). Since the existing masking techniques sample tokens randomly from *space* (tube), *time* (frame), or *spacetime* (patch) and do not depend on the underlying scene, it is possible that the decoder tries to reconstruct high spatiotemporal information regions when most of the visible tokens belong to low spatiotemporal information regions [19]. Consequently, a large number of pre-training iterations are required to learn meaningful representations [11, 48] for downstream tasks. Inspired by this phenomenon, we design an adaptive masking technique based on reinforcement learning that predicts the sampling distribution for a given video.

### 2.3. Reinforcement learning

Reinforcement learning (RL) [23] has become an increasingly popular research area due to its effectiveness in various tasks, including playing Atari games [35], image captioning [59], person re-identification [60], video summarization [43, 66], etc. Policy gradient methods [56] for RL focus on selecting optimal policies that maximize expected return by gradient descent. REINFORCE [44] is commonly seen as the basis for policy gradient methods in RL. However, its application in MAEs has yet to be explored. Our sampling network learns a categorical distribution over all tokens to sample informative visible tokens for pre-training MAEs. We use REINFORCE policy gradient [44] method as an unbiased estimator for calculating gradients to update the sampling network parameters $\theta$.

When the probability density function (estimated by the sampling network $f_\theta$) is differentiable with respect to its parameters $\theta$, we need to sample an action and compute probabilities $p(\cdot)$ to implement policy gradient [44]:

$$\Delta\theta = \alpha \cdot R \cdot \frac{\partial \log p(a)}{\partial \theta}, \tag{1}$$

where $R$ is the return, $p(a)$ is the probability of taking action $a$, and $\alpha$ is the learning rate. In our case, we sample a mask (the *action*) based on the probability distribution estimated by the sampling network, reconstruct masked tokens from the MAE (the *environment*), and then utilize reconstruction error (the *reward*) to compute the expected return.

## 3. Method

### 3.1. Architecture of *Ada*MAE

Fig. 2 shows our *Ada*MAE architecture which consists of four main components, three of which (Tokenizer, Encoder, and Decoder) are standard to MAE along with our proposed Adaptive Token Sampler.

**Tokenizer:** Given an input video $\mathbf{V}$ of size $T \times C \times H \times W$, where $T$ denotes the number of temporal frames, $C$ denotes the input (RGB) channels, and $H \times W$ is the spatial resolution of a frame, we first pass it through a `Tokenizer` (i.e., 3D convolution layer with kernel $K$ of size $(t, C, h, w)$ and stride $S$ of size $(t, h, w)$, and $d$ output channels) to tokenize $\mathbf{V}$ into $N$ tokens of dimension $d$ denoted as $\mathbf{X}$, where $N = \frac{T}{t} \times \frac{H}{h} \times \frac{W}{w}$. Next, we inject the positional information into the tokens by utilizing the fixed 3D periodic positional encoding scheme introduced in [50].

**Adaptive Token Sampler:** Given the tokens $\mathbf{X}$ resulting from the `Tokenizer`, we pass them through a light-weight Multi-Head Attention (MHA) network followed by a `Linear` layer and a `Softmax` activation to obtain the probability scores $\mathbf{P} \in \mathbb{R}^N$ for all tokens as follows:

$$\mathbf{Z} = \mathtt{MHA}(\mathbf{X}); \quad \mathbf{Z} \in \mathbb{R}^{N \times d}, \tag{2}$$

$$\mathbf{P} = \mathtt{Softmax}(\mathtt{Linear}(\mathbf{Z})); \quad \mathbf{P} \in \mathbb{R}^{N}. \tag{3}$$

We then assign an $N$-dimensional categorical distribution [34] over $\mathbf{P}$ ($\boldsymbol{p} \sim \text{Categorical}(N, \mathbf{P})$) and draw (without replacement) a set of visible token indices $\boldsymbol{I}_v$ (hence the set of masked token indices is given by $\boldsymbol{I}_m = U - \boldsymbol{I}_v$, where $U$=$\{1, 2, 3, \cdots, N\}$ is the set of all indices, i.e., $\boldsymbol{I}_m$ is the set complement of $\boldsymbol{I}_v$). The number of sampled visible tokens $N_v$ is computed based on a pre-defined masking ratio $\rho \in (0, 1)$ and equals to $N \times (1 - \rho)$.

**Encoder:** Next, we generate latent representations $\mathbf{F}_v$ by passing sampled *visible* tokens $\mathbf{X}_v$ through the `ViTEncoder`.

**Decoder:** The visible token representations $\mathbf{F}_v$ are concatenated with a fixed learnable representation $f_m$ for masked tokens. Next, positional information is added [39] for both representations, instead of shuffling them back to the original order. Finally, we pass this through a light-weight transformer decoder to obtain the predictions $\widehat{\mathbf{V}}$.

### 3.2. Optimizing *Ada*MAE

**Masked reconstruction loss $\mathcal{L}_R$:** We utilize the mean squared error (MSE) loss between the predicted and *patch normalized*[1] ground-truth RGB values [17] of the masked tokens to optimize the MAE (parameterzied by $\phi$) as follows:

$$\mathcal{L}_{\text{R}}(\phi) = \frac{1}{N - N_v} \sum_{i \in I_m} \| \widehat{\mathbf{V}}_i - \widetilde{\mathbf{V}}_i \|_2, \tag{4}$$

where $\widehat{\mathbf{V}}$ denotes the predicted tokens, $\widetilde{\mathbf{V}}$ denotes the local patch normalized ground-truth RGB values [48].

**Adaptive sampling loss $\mathcal{L}_S$:** We optimize the adaptive token sampling network (parameterized by $\theta$) using sampling loss $\mathcal{L}_{\text{S}}$, which enables gradient update independent of the MAE (parameterized by $\phi$). The formulation of $\mathcal{L}_{\text{S}}$ is

[1]Each patch is normalized based on the patch mean and variance.

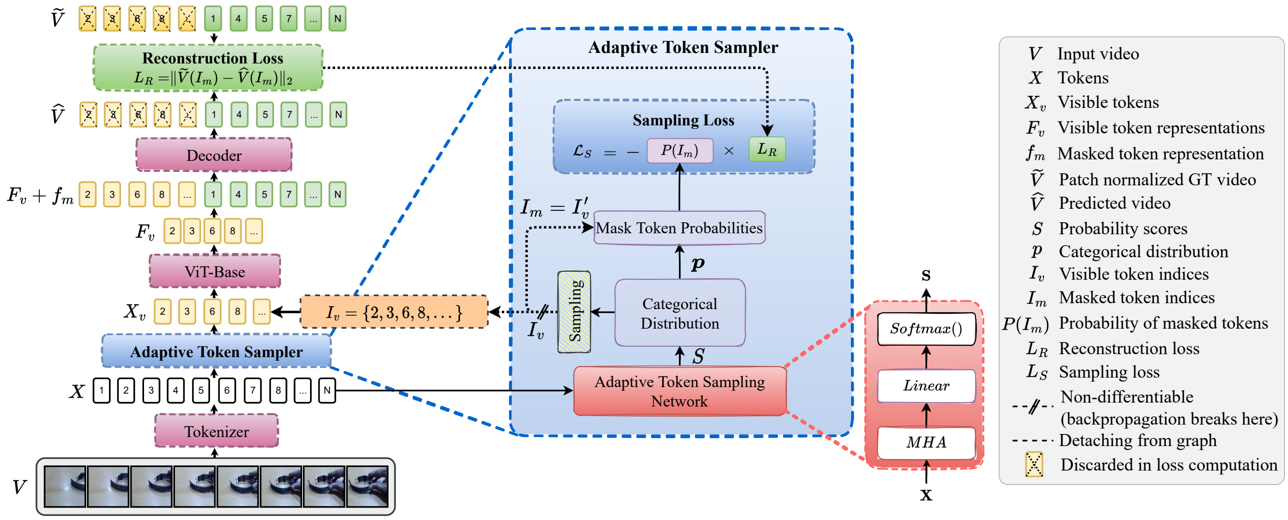



Figure 2. ***Ada*MAE** samples *visible tokens* based on the *categorical distribution* estimated by the *sampling network* and reconstructs the missing ones with a ViT encoder-decoder architecture. Since the sampling process is non-differentiable, we optimize it by ***maximizing*** the *expected reconstruction loss*, which results in the sampling network predicting *high probability* values for tokens that may lead to high reconstruction errors. Our adaptive sampling process samples more tokens from *high spatiotemporal information* regions and fewer tokens from the *low information* or *redundant* regions.

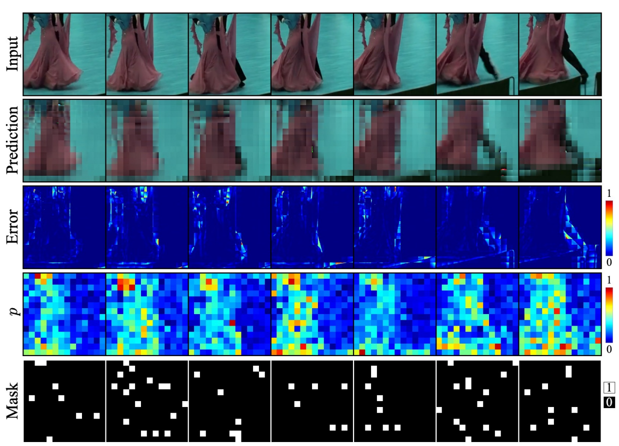



Figure 3. **Understanding the formulation of $L_S$.** We observe high $L_R$ in high activity/information regions (third row). Hence, we propose to *maximize expected reconstruction error* over masked tokens which ultimately results in the sampling network predicting *higher probability scores* for *high activity* regions and *lower probability scores* for *low activity* (i.e., background or redundant) regions. See Appendix for more visualizations.

motivated by the REINFORCE algorithm in RL, where we consider the visible token sampling process as an *action*, the MAE (ViT-backbone and decoders) as an *environment*, and the masked reconstruction loss $\mathcal{L}_R$ as the *return*. Following the expected reward maximization in the REINFORCE algorithm, we propose to optimize the sampling network by **maximizing** the *expected reconstruction error* $\mathbb{E}[\mathcal{L}_R]$.

To elucidate why maximizing $\mathbb{E}[\mathcal{L}_R]$ works for adaptive sampling, we use a visual example shown in Figure 3. Given a video (*first row*) with *high activity/information* (i.e., the dancing couple) and *low activity/information* (i.e., the teal color background) regions, we observe a high reconstruction error (*third row*) around the foreground. Since our objective is to sample more visible tokens from high activity regions and fewer tokens from the background, we optimize the sampling network by maximizing the expected reconstruction error over the masked tokens (denoted by $\mathbb{E}[\mathcal{L}_R]$). When optimized with the above rule, the adaptive token sampling network predicts *high probability scores* for the tokens from the *high activity* regions compared to the tokens from the background as shown in the fourth row. We observe that this adaptive token sampling approach for MAEs closely aligns with non-uniform sampling in compressed sensing, where more samples are assigned to the regions with high activity and fewer to those with low activity. Since our adaptive sampling allocates samples based on the level of spatiotemporal information, it requires fewer tokens to achieve the same reconstruction error compared to random sampling (which is not efficient in token allocation). This also enables using extremely high masking ratios (95%) for MAEs which further reduces the computational burden, thereby accelerating the pre-training process.

The objective function to optimize our adaptive token sampling network can be expressed as:

$$\mathcal{L}_S(\theta) = -\mathbb{E}_\theta[\mathcal{L}_R(\phi)] = -\sum_{i \in \mathbf{I}_m} P_\theta^i \cdot \mathcal{L}_R^i(\phi), \qquad (5)$$

where $P_\theta^i$ is the probability of the mask token at index $i$ inferred from the adaptive token sampling network parameterized by $\theta$ and $\mathcal{L}_R^i$ is the reconstruction error of the $i$-th masked token. Note that $\mathcal{L}_R(\phi)$ is the reconstruction error incurred by MAE with parameters $\phi$. Also, we explicitly prevent gradient updates from $\mathcal{L}_S(\theta)$ to propagate through the MAE (i.e., detach from the computational graph).

Furthermore, we take logarithm of the probabilities to avoid the precision errors caused by small probability values. Note that the negative sign is due to gradient descent optimization, whilst the rule above assumes gradient ascent.

## 4. Experimental Setup

**Datasets.** We evaluate *Ada*MAE on two datasets: Something-Something-v2 (SSv2) [15] and Kinetics-400 [25]. The SSv2 dataset is a relatively large-scale video dataset, consisting of approximately 170k videos for training, and 20k videos for validation, categorized into 174 action classes. Kinetics-400 is also a large-scale dataset which contains 400 action classes, ∼240k training videos, and ∼20k validation videos. We conduct ablation studies on the SSv2 dataset and report results on both SSv2 and Kinetics-400.

**Dataset preprocessing.** We closely follow VideoMAE [48] for dataset preprocessing. For both datasets, we sample 16 RGB frames, each with $224 \times 224$ pixels, from the raw video with a temporal stride of 4, where the starting frame location is randomly selected [11]. As part of the data augmentations for pre-training, we employ random resized crops [45] in spatial domain, random scale $\in [0.5, 1]$, and random horizontal flipping [11].

**Network architecture.** For our experiments we use vanilla ViT [8]. Specifically, we select ViT-Base to save GPU memory and avoid longer pre-training time. We use patch size of $2 \times 3 \times 16 \times 16$ [1, 11, 48], resulting in $(16/2) \times (3/3) \times (224/16) \times (224/16) = 1568$ tokens for an input video of size $16 \times 3 \times 224 \times 224$. See supplementary material for more network architecture details.

**Pre-training setting.** For all experiments, the default number of pre-training epochs is 800 unless otherwise noted. We use `adamw` [33] optimizer with a batch size of 32/GPU and 8 GPUs. See supplementary material for more details.

**Evaluation on action classification.** In order to evaluate the quality of the pre-trained model, we perform end-to-end fine-tuning (instead of linear probing [4, 11, 17, 48]). During inference, we follow the common practice of multi-view testing [36, 53] where $K$ temporal clips (by default $K = 2$ [48] and 7 [11, 48] on SSv2 and Kinetics, respectively) to cover the video length. For each clip, we take 3 spatial views [11, 48] to cover the complete image. The final prediction is the average across all views.

## 5. Results and Discussion

### 5.1. Ablation studies

We perform pre-training using *Ada*MAE on ViT-Base backbone and then fine-tune the encoder with supervision for evaluation on the SSV2 dataset. We present our ablation study results in Table 1. In the following sections, by accuracy we refer to top-1 accuracy unless stated otherwise.

**Adaptive *vs.* random patch, frame, and tube masking.** We compare different masking techniques (as illustrated in Fig. 1) in Table 1a. Random *tube masking* [48], which randomly samples masked tokens in the 2D spatial domain and then extends those along the temporal axis, works reasonably well with a high masking ratio (69.3% accuracy with 90% masking). Random *patch masking* [11], which randomly masks tokens over spacetime, also works well with high masking ratios (68.3% accuracy with 90% masking *vs.* 67.3% accuracy with 75% masking) - note that we observe 1% drop in performance compared to random *tube masking* [48]. However, random *frame masking*, which masks out all tokens in randomly selected frames, performs poorly (61.5% accuracy) compared to random patch and tube masking. Our ***adaptive token sampling***, which samples visible tokens based on a categorical distribution (instead of assuming a fixed distribution) estimated from a neural network by exploiting the spatiotemporal relationship of all tokens, yields the *best* result (70.04% accuracy) – **0.7%** improvement over random patch and tube masking. Note that this improvement is achieved with a higher masking ratio (95%) which requires less memory (14.4 GB *vs.* 14.7 GB) and results in shorter pre-training time.

**Masking ratio.** Table 1b shows the performance of our model while using different masking ratios. Our ***adaptive token sampling*** achieves the **best** performance for masking ratio of **95%** (70.04% accuracy); 5% higher masking ratio compared to random *patch masking* in SpatiotemporalMAE [11] (90% masking) and random *tube masking* in VideoMAE [48] (90% masking). As shown in Fig. 4a and 4b, we sample a relatively higher number of tokens from regions with less redundancy and a small number of tokens from regions with high redundancy. Hence, fewer tokens are required than random sampling to achieve the same reconstruction error. 98% and 90% masking ratios also perform reasonably well (68.85% and 69.55% accuracies, respectively), whereas we observe a significant drop in performance for 85% and 80% masking ratios (68.06% and 66.96% accuracies, respectively). For lower masking ratios, due to large number of redundant visible patches being sampled, the network just copies features from these patches, resulting in poor generalizations and drop in performance.

**Decoder depth.** We compare our performance for different decoder depths (i.e., number of transformer blocks used)

| case | ratio | top-1 | top-5 | memory |
|---|---|---|---|---|
| Tube [48] | 75 | 67.3 | 91.5 | 16.2 GB |
| Tube [48] | 90 | 69.3 | 92.3 | 14.7 GB |
| Random [11] | 75 | 67.9 | 90.6 | 16.2 GB |
| Random [11] | 90 | 68.3 | 91.8 | 14.7 GB |
| Frame [11, 48] | 87.5 | 61.5 | 87.6 | 12.8 GB |
| Adaptive (ours) | 95 | **70.04** | **92.70** | 14.4 GB |

(a) **Mask sampling techniques**. Our *adaptive* masking works better compared to tube, frame, and random, and requires less memory.

| ratio | top-1 | top-5 | memory |
|---|---|---|---|
| 0.98 | 68.85 | 91.94 | 14.1 GB |
| 0.95 | **70.04** | **92.70** | 14.4 GB |
| 0.90 | 69.55 | 92.62 | 15.1 GB |
| 0.85 | 68.06 | 91.38 | 15.9 GB |
| 0.80 | 66.96 | 90.91 | 16.9 GB |

(b) **Different masking ratio** ($\rho$). Our AdaMAE works well with extremely high masking ratio, hence requires less memory.

| blocks | top-1 | top-5 | memory |
|---|---|---|---|
| 1 | 68.86 | 92.07 | 9.2 GB |
| 2 | 69.13 | 92.17 | 11.0 GB |
| 4 | **70.04** | **92.70** | 14.4 GB |
| 8 | 69.97 | 92.67 | 21.3 GB |

(c) **Different decoder depth** ($D$). Our AdaMAE achives the best performance with 4 blocks of decoder.

| case | top-1 | top-5 |
|---|---|---|
| L1 loss (w norm.) | 69.12 | 91.42 |
| L1 loss (w/o norm.) | 68.75 | 91.47 |
| MSE loss (w norm.) | **70.04** | **92.70** |
| MSE loss (w/o norm.) | 68.87 | 91.45 |

(d) **Loss function**. MSE loss with local patch normalization gives the best results.

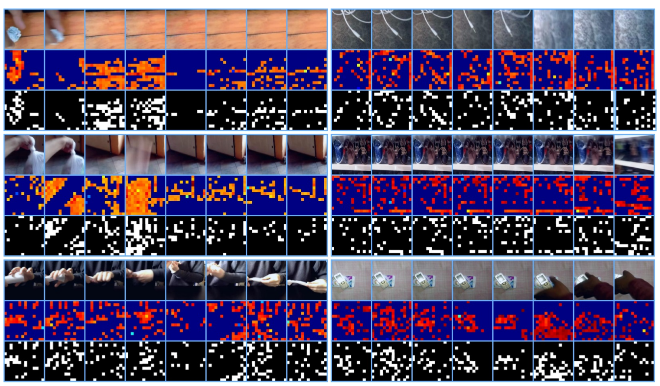
| epochs | top-1 | top-5 |
|---|---|---|
| 200 | 66.42 | 90.59 |
| 500 | 69.20 | 92.50 |
| 600 | 69.47 | 92.52 |
| 800 | **70.04** | **92.70** |

(e) **Pre-training epochs**. Better performance with a greater number of pre-training epochs.

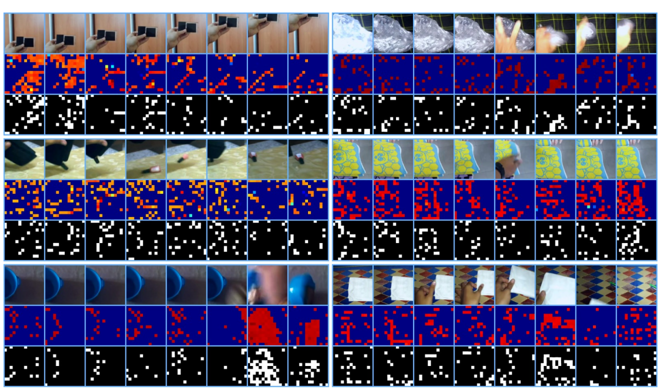
| case | top-1 | top-5 | memory |
|---|---|---|---|
| MLP | 65.41 | 89.22 | 11.8 GB |
| MHA ($D = 1, d = 384$) | 69.56 | 90.89 | 14.0 GB |
| MHA ($D = 1, d = 768$) | **70.04** | **92.70** | 14.4 GB |
| MHA ($D = 2, d = 384$) | 70.03 | 92.74 | 17.1 GB |

(f) **Mask sampling network**. Our AdaMAE works well with a single MHA.

Table 1. **Ablation experiments on SSv2 dataset**. We use ViT-Base [50] as the backbone for all experiments. MHA ($D = 2, d = 384$) denotes our adaptive token sampling network with a depth of two and embedding dimension of 384. All pre-trained models are evaluated based on the evaluation protocol described in Sec. 4. The default choice of our *Ada*MAE is highlighted in **gray** color. The GPU memory consumption is reported for a batch size of 16 on a single GPU.



(a) Masking ratio = 85%.



(b) Masking ratio = 90%.

Figure 4. **Mask visualizations of our *Ada*MAE for different masking ratios on SSV2 dataset [15].** Given a video (*first row*), our *Ada*MAE first predicts the categorical distribution (*second row*), and then samples the mask (*third row*) from that distribution. Refer to the supplementary material for more visualizations.

in Table 1c. Increasing the decoder depth from 1 block to 4 blocks increases the accuracy (from 68.86% to 70.04%); however, this comes with an additional longer pre-training time. Increasing the decoder depth from 4 blocks to 8 blocks, however, results in a slight drop in accuracy (from 70.04% to 69.97%). Hence, we consider 4 transformer blocks as the optimal decoder depth for *Ada*MAE, on par with the observation made in VideoMAE [48] and SpatiotemporalMAE [11].

**Reconstruction target.** Table 1d shows the impact of employing different loss functions and reconstruction targets. We use RGB pixel values as the reconstruction target; instead of utilizing complicated features such as HoG [54], which always comes with additional computational cost. Following an improved performance with per-patch normalized pixel values for images in ImageMAE [17] and also verified later for videos in VideoMAE [48] and SpatiotemporalMAE [11], we consider two scenarios – with and without local patch normalization. Regression over raw RGB pixels with the L1/MSE loss has shown poor performance compared to regression over per-patch normalized pixels. We observe that MSE loss has a slight advantage over L1 loss.

**Pre-training epochs.** Next, we show in Table 1e the effect of the number of pre-training epochs on fine-tuning performance. Increasing the number of pre-training epochs always increases the performance: for 200 epochs - 66.42%, for 500 epochs 69.20%, for 600 epochs - 69.47%, for 700 epochs 69.68, and for 800 epochs we achieve the best result of 70.04%. We observe poor performance when pre-training for fewer than 500 epochs. Hence, considering the trade-off between pre-training time and performance, we recommend pre-training with *Ada*MAE for at least 500 epochs.

**Mask sampling network.** Finally, we investigate the effect of using different network architectures for our adaptive token sampling network in Table 1f. Considering the importance of a computationally lightweight design for an adaptive token sampling network, we first experimented with a simple multi-layer perceptron (MLP) network (see supplementary material). The results of this MLP network were not encouraging, giving similar performance to random patch masking. We visualize the predicted probability map to confirm that it was uncorrelated to the video itself (see supplementary material). This is expected since a simple MLP network cannot model the relationship between the tokens as it operates only on the embedding dimension of each token. We next experimented with an MHA network (see supplementary material). This network can capture the spatiotemporal relationships between tokens through the attention mechanism and is able to predict meaningful probability density maps, as shown in Figs. 4a, and 4b. Increasing the number of MHA blocks results in slight improvements in performance, albeit with more memory and compute requirements. We selected a single MHA block as our adaptive token sampling network for all other experiments to keep it as computationally lightweight as possible. Furthermore, we experimented with different embedding dimensions (i.e., $d$ and $d/2$) and observed improved performance when operating MHA at its original embedding dimension $d$.

### 5.2. Main results and analysis

We compare the performance of *Ada*MAE with previous SOTA results for action classification on the SSv2 [15] and Kinetics-400 [25] datasets. Our results are presented in Table 2 (on SSv2) and Table 3 (on Kinetics-400) for the ViT-Base backbone ($\sim$ 87M parameters).

MAEs (MaskFeat, VideoMAE, SpatioTemporalMAE, OmniMAE, and *Ada*MAE) generally outperform previous supervised representation learning approaches (such as TDN [51], TimeSformer [5], Motionformer [38], Video-Swin [30]), which use ImageNet-1K (IN1K), ImageNet-21K (IN21K), Kinetics-400, and/or Kinetics-600 labels for supervision, by a significant margin on these datasets. This empirically demonstrates the powerful representation learning capability of masked reconstruction methods. Furthermore, they also outperform recent contrastive learning approaches such as BEVT [52] and hybrid approaches (i.e., masked modeling and contrastive learning) such as VIMPAC [46].

We compare *Ada*MAE with recent MAE-based spatiotemporal representation learning approaches: VideoMAE [48], SpatioTemporalMAE [11], and OmniMAE [14] with the same ViT-Base backbone. Among these, VideoMAE performs best when pre-trained with 90% random tube masking (69.3% and 80.9% in accuracies on SSv2 and Kinetics-400, respectively). SpatioTemporalMAE achieves its best performance with 90% random patch masking (68.3% and 81.3% accuracies on SSv2 and Kinetics-400, respectively). OminiMAE shows lower performance than VideoMAE and SpatiotemporalMAE on SSv2 and Kinetics-400 datasets (69.3% and 80.6% accuracies on SSv2 and Kinetics-400, respectively) with 95% random patch masking, even though it uses extra data (IN1K with 90% random patch masking, and Kinetics-400 and SSv2 with 95% random masking). In contrast, our AdaptiveMAE, which utilizes an adaptive masking scheme based on the spatiotemporal information of a given video, achieves the best performance on the SSv2 and Kinetics-400 datasets (70.0% and 81.7% accuracies, respectively) with higher masking ratio (95%). This results in faster pre-training compared to VideoMAE and SpatitemporalMAE and requires less GPU memory.

**Transferability:** We investigate the transferability of *Ada*MAE pre-trained models by evaluating their performance in scenarios where the pre-training and finetuning datasets are different. Our *Ada*MAE *pre-trained on Kinetics-400* achieves SOTA results of **69.9%** (top-1 accuracy) and 92.7% (top-5 accuracy) on *SSv2*. On the other hand, *Ada*MAE *pre-trained on SSv2* achieves **81.2%** (top-1 accuracy) and 94.5% (top-5 accuracy) on *Kinetics-400*, which is better than other frameworks.

*Please refer to the supplementary material document for additional transfer learning results, as well as results for low-budget supervision (i.e., linear probing).*

## 6. Limitations and Future Work

Although we have presented results for ViT-Base backbone, our approach for adaptive token sampling scales-up for larger backbones (e.g., *ViT-Large* and *ViT-Huge*) as well. We expect *Ada*MAE to achieve better results for these settings based on the experiments in previous studies [11, 48], and we plan to conduct similar studies in future. Since *Ada*MAE outputs a categorical distribution on the tokens, it could be used for applications such as saliency detection in videos. Moreover, *Ada*MAE needs only 5% of the video tokens to represent the entire video reasonably well. Hence, it could be applied for efficient video compression and retrieval.

## 7. Conclusion

We propose an adaptive masking technique for MAEs, *Ada*MAE, that efficiently learns meaningful spatiotemporal representations from videos for downstream tasks. Unlike existing masking methods such as random patch, tube, and frame masking, *Ada*MAE samples visible tokens based on a categorical distribution. An auxiliary network is optimized to learn the distribution by maximizing the expected reconstruction error using policy gradients. *Ada*MAE outperforms previous state of the art methods with ViT-Base model on Something-Something v2 and Kinetics-400 datasets and learns better transferable features, while lowering GPU memory requirements and masking pre-training faster.

| Method | Backbone | Extra data | Extra labels | Pre. Epochs | Frames | GFLOPs | Param | Top-1 | Top-5 |
|---|---|---|---|---|---|---|---|---|---|
| TSM$_{two\ stream}$ [29] | ResNet50×2 | IN1K | ✓ | N/A | 16+16 | 130×2×3 | 49 | 66.0 | 90.5 |
| TEINet$_{En}$ [31] | ResNet50×2 | | ✓ | N/A | 8+16 | 99×10×3 | 50 | 66.6 | N/A |
| TANet$_{En}$ [32] | ResNet50×2 | | ✓ | N/A | 8+16 | 99×2×3 | 51 | 66.0 | 90.1 |
| TDN$_{En}$ [51] | ResNet101×2 | | ✓ | N/A | 8+16 | 198×1×3 | 88 | 69.6 | 92.2 |
| SlowFast [12] | ResNet101 | Kinetics-400 | ✓ | N/A | 8+32 | 106×1×3 | 53 | 63.1 | 87.6 |
| MViTv1 [9] | MViTv1-B | | ✓ | N/A | 64 | 455×1×3 | 37 | 67.7 | 90.9 |
| MViTv1 [9] | MViTv1-B | Kinetics-600 | ✓ | N/A | 32 | 170×1×3 | 37 | 67.8 | 91.3 |
| MViTv1 [9] | MViTv1-B-24 | | ✓ | N/A | 32 | 236×1×3 | 53 | 68.7 | 91.5 |
| TimeSformer [5] | ViT-B | IN21K | ✓ | N/A | 8 | 196×1×3 | 121 | 59.5 | N/A |
| TimeSformer [5] | ViT-L | | ✓ | N/A | 64 | 5549×1×3 | 430 | 62.4 | N/A |
| ViViT FE [1] | ViT-L | IN21K+Kinetics-400 | ✓ | N/A | 32 | 995×4×3 | N/A | 65.9 | 89.9 |
| Motionformer [38] | ViT-B | | ✓ | N/A | 16 | 370×1×3 | 109 | 66.5 | 90.1 |
| Motionformer [38] | ViT-L | | ✓ | N/A | 32 | 1185×1×3 | 382 | 68.1 | 91.2 |
| Video Swin [30] | Swin-B | | ✓ | N/A | 32 | 321×1×3 | 88 | 69.6 | 92.7 |
| VIMPAC [46] | ViT-L | HowTo100M+DALLE | ✗ | N/A | 10 | N/A×10×3 | 307 | 68.1 | N/A |
| BEVT-V [52] | Swin-B | Kinetics-400+DALLE | ✗ | N/A | 32 | 321×1×3 | 88 | 67.1 | N/A |
| BEVT [52] | Swin-B | IN-1K+Kinetics-400+DALLE | ✗ | N/A | 32 | 321×1×3 | 88 | 70.6 | N/A |
| MaskFeat↑312 [54] | MViT-L | Kinetics-400 | ✓ | N/A | 40 | 2828×1×3 | 218 | 74.4 | 94.6 |
| MaskFeat↑312 [54] | MViT-L | Kinetics-600 | ✓ | N/A | 40 | 2828×1×3 | 218 | 75.0 | 95.0 |
| MTV (B/2+S/4+Ti/8) [62] | ViT-B | Kinetics-400+I21K | ✓ | N/A | 32 | 384 × 4 × 3 | 310 | 67.6 | 90.1 |
| SpatioTemporalMAE [11] | ViT-B | *no external data* | ✗ | 800 | 16 | 180×2×3 | 87 | 68.3 | 91.8 |
| VideoMAE [48] | ViT-B | *no external data* | ✗ | 800 | 16 | 180×2×3 | 87 | 69.3 | 92.3 |
| OmniMAE [14] | ViT-B | IN1K | ✗ | 800 | 16 | 180×5×3 | 87 | 69.3 | N/A |
| ***Ada*MAE $_{\rho=95\%}$ (ours)** | ViT-B | *no external data* | ✗ | 800 | 16 | 180×2×3 | 87 | **70.0** | **92.7** |

Table 2. **Comparison of our *Ada*MAE with SOTA methods on SSv2** [15]. We report the results for ViT-Base [50] architecture. Our model is pre-trained for the default setting in Table 1. The ✓ in extra labels tab denotes supervised data used for pre-training while ✗ denotes only unlabeled data is used for the pre-training. The N/A denotes these numbers are not available/reported in the paper.

| Method | Backbone | Extra data | Extra labels | Pre. Epochs | Frames | GFLOPs | Param | Top-1 | Top-5 |
|---|---|---|---|---|---|---|---|---|---|
| NL I3D [53] | ResNet101 | IN1K | ✓ | N/A | 128 | 359×10×3 | 62 | 77.3 | 93.3 |
| TANet [32] | ResNet152 | | ✓ | N/A | 16 | 242×4×3 | 59 | 79.3 | 94.1 |
| TDN$_{En}$ [51] | ResNet101$_{\times 2}$ | | ✓ | N/A | 8+16 | 198×10×3 | 88 | 79.4 | 94.4 |
| Video Swin [30] | Swin-B | | ✓ | N/A | 32 | 282×4×3 | 88 | 80.6 | 94.6 |
| TimeSformer [5] | ViT-B | IN21K | ✓ | N/A | 8 | 196×1×3 | 121 | 78.3 | 93.7 |
| TimeSformer [5] | ViT-L | | ✓ | N/A | 96 | 8353×1×3 | 430 | 80.7 | 94.7 |
| ViViT FE [1] | ViT-L | | ✓ | N/A | 128 | 3980×1×3 | N/A | 81.7 | 93.8 |
| Motionformer [38] | ViT-B | | ✓ | N/A | 16 | 370×10×3 | 109 | 79.7 | 94.2 |
| Motionformer [38] | ViT-L | | ✓ | N/A | 32 | 1185×10×3 | 382 | 80.2 | 94.8 |
| Video Swin [30] | Swin-L | | ✓ | N/A | 32 | 604×4×3 | 197 | 83.1 | 95.9 |
| ViViT FE [1] | ViT-L | JFT-300M | ✓ | N/A | 128 | 3980×1×3 | N/A | 83.5 | 94.3 |
| ViViT [1] | ViT-H | JFT-300M | ✓ | N/A | 32 | 3981×4×3 | N/A | 84.9 | 95.8 |
| VIMPAC [46] | ViT-L | HowTo100M+DALLE | ✗ | N/A | 10 | N/A×10×3 | 307 | 77.4 | N/A |
| BEVT [52] | Swin-B | IN1K+DALLE | ✗ | N/A | 32 | 282×4×3 | 88 | 80.6 | N/A |
| ip-CSN [49] | ResNet152 | *no external data* | ✗ | N/A | 32 | 109×10×3 | 33 | 77.8 | 92.8 |
| SlowFast [12] | R101+NL | | ✗ | N/A | 16+64 | 234×10×3 | 60 | 79.8 | 93.9 |
| MViTv1 [9] | MViTv1-B | | ✗ | N/A | 32 | 170×5×1 | 37 | 80.2 | 94.4 |
| MaskFeat [54] | MViT-L | Kinetics-600 | ✗ | N/A | 16 | 377×10×1 | 218 | 84.3 | 96.3 |
| MaskFeat↑352 [54] | MViT-L | | ✗ | N/A | 40 | 3790×4×3 | 218 | 87.0 | 97.4 |
| VideoMAE [48] | ViT-B | *no external data* | ✗ | 1600 | 16 | 180×5×3 | 87 | 80.9 | 94.7 |
| VideoMAE [48] | ViT-L | | ✗ | 1600 | 16 | 597×5×3 | 305 | 84.7 | 96.5 |
| VideoMAE↑320 [48] | ViT-L | | ✗ | 1600 | 16 | 3958×5×3 | 305 | 85.8 | 97.1 |
| SpatioTemporalMAE [11] | ViT-B | *no external data* | ✗ | 1600 | 16 | 180×3×7 | 87 | 81.3 | 94.9 |
| SpatioTemporalMAE [11] | ViT-L | | ✗ | 1600 | 16 | 598×3×7 | 305 | 84.8 | 96.2 |
| SpatioTemporalMAE [11] | ViT-H | | ✗ | 1600 | 16 | 1193×3×7 | 632 | 85.1 | 96.6 |
| OmniMAE [11] | ViT-B | IN1K + SSv2 | ✗ | 1600 | 16 | 180×3×7 | 87 | 80.6 | N/A |
| OmniMAE↓512 [11] | ViT-L | | ✗ | 1600 | 16 | 598×3×7 | 304 | 84.0 | N/A |
| OmniMAE [11] | ViT-H | | ✗ | 1600 | 16 | 1193×3×7 | 632 | 84.8 | N/A |
| ***Ada*MAE $_{\rho=95\%}$ (ours)** | ViT-B | *no external data* | ✗ | 1600 | 16 | 180×5×3 | 87 | **81.7** | **95.2** |

Table 3. **Comparison of our *Ada*MAE with SOTA methods on Kinetics-400** [25]. We report the results for ViT-Base [50] architecture. Our model is pre-trained for the default setting in Table 1. The ✓ in extra labels tab denotes supervised data used for pre-training while ✗ denotes only unlabeled data is used for the pre-training. The N/A denotes these numbers are not available/reported in the paper.

## References


[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 5, 8

[2] Wele Gedara Chaminda Bandara, Nithin Gopalakrishnan Nair, and Vishal M Patel. Ddpm-cd: Remote sensing change detection using denoising diffusion probabilistic models. *arXiv preprint arXiv:2206.11892*, 2022. 1

[3] Wele Gedara Chaminda Bandara and Vishal M Patel. A transformer-based siamese network for change detection. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210. IEEE, 2022. 1

[4] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 2, 5

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 813–824. PMLR, 18–24 Jul 2021. 7, 8

[6] Jun Chen, Ming Hu, Boyang Li, and Mohamed Elhoseiny. Efficient Self-supervised Vision Pretraining with Local Masked Reconstruction, June 2022. arXiv:2206.00790 [cs]. 1

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining From Pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, July 2020. 2

[8] 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 *International Conference on Learning Representations*, 2021. 1, 2, 5

[9] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 8

[10] Yuxin Fang, Shusheng Yang, Shijie Wang, Yixiao Ge, Ying Shan, and Xinggang Wang. Unleashing vanilla vision transformer with masked image modeling for object detection, 2023. 2

[11] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 5, 6, 7, 8

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 8

[13] Peng Gao, Teli Ma, Hongsheng Li, Ziyi Lin, Jifeng Dai, and Yu Qiao. MCMAE: Masked convolution meets masked autoencoders. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2

[14] Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. OmniMAE: Single Model Masked Pretraining on Images and Videos, June 2022. arXiv:2206.08356 [cs, stat]. 1, 2, 7, 8

[15] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 2, 5, 6, 7, 8

[16] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. MaskViT: Masked Visual Pre-Training for Video Prediction, Aug. 2022. arXiv:2206.11894 [cs]. 1

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1, 2, 3, 5, 6

[18] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1

[19] Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. MILAN: Masked Image Pretraining on Language Assisted Representation, Aug. 2022. arXiv:2208.06049 [cs]. 3

[20] Lang Huang, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, and Toshihiko Yamasaki. Green hierarchical vision transformer for masked image modeling. *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. 1

[21] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A Survey on Contrastive Self-Supervised Learning. *Technologies*, 9(1):2, Mar. 2021. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute. 1

[22] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. 2

[23] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237–285, May 1996. 2, 3

[24] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 300–318. Springer, 2022. 2

[25] 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, May 2017. arXiv:1705.06950 [cs]. 2, 5, 7, 8

[26] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, pages 4171–4186, 2019. 2

[27] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020. 1, 2

[28] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34:13165–13176, 2021. 2

[29] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019. 8

[30] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11669–11676, 2020. 7, 8

[31] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11669–11676, 2020. 8

[32] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13708–13718, 2021. 8

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

[34] Francis Henry Marriott, Charles. *A dictionary of statistical terms.* Longman Scientific & Technical, 5-th edition, 1990. 3

[35] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning, Dec. 2013. arXiv:1312.5602 [cs]. 3

[36] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021. 5

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, Jan. 2019. arXiv:1807.03748 [cs, stat]. 1, 2

[38] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506, 2021. 7, 8

[39] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal Contrastive Video Representation Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6960–6970, Nashville, TN, USA, June 2021. IEEE. 1, 2, 3

[40] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. *cs.ubc*, page 12, 2018. 2

[41] 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*, pages 8821–8831. PMLR, 2021. 2

[42] Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell. Loss is its own Reward: Self-Supervision for Reinforcement Learning, Mar. 2017. arXiv:1612.07307 [cs]. 1, 2

[43] Xinhui Song, Ke Chen, Jie Lei, Li Sun, Zhiyuan Wang, Lei Xie, and Mingli Song. Category driven deep recurrent neural network for video summarization. In *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6, July 2016. 3

[44] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. 3

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5

[46] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. VIMPAC: Video Pre-Training via Masked Token Prediction and Contrastive Learning, June 2021. arXiv:2106.11250 [cs]. 7, 8

[47] Chenxin Tao, Xizhou Zhu, Gao Huang, Yu Qiao, Xiaogang Wang, and Jifeng Dai. Siamese Image Modeling for Self-Supervised Vision Representation Learning, July 2022. arXiv:2206.01204 [cs]. 1

[48] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 5, 6, 7, 8

[49] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 8

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 6, 8

[51] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1895–1904, 2021. 7, 8

[52] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14733–14743, 2022. 1, 2, 7, 8

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 5, 8

[54] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022. 1, 2, 6, 8

[55] Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. Should You Mask 15% in Masked Language Modeling?, May 2022. arXiv:2202.08005 [cs]. 2

[56] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. 2, 3

[57] Jiahao Xie, Wei Li, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Masked Frequency Modeling for Self-Supervised Visual Pre-Training, June 2022. arXiv:2206.07706 [cs]. 1, 2

[58] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 2

[59] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015. 3

[60] Shaogang Gong XU LAN, HangXiao Wang and Xiatian Zhu. Deep reinforcement learning attention selection for person re-identification. In Gabriel Brostow Tae-Kyun Kim, Stefanos Zafeiriou and Krystian Mikolajczyk, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 121.1–121.16. BMVA Press, September 2017. 3

[61] Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. Probing inter-modality: Visual parsing with self-attention for vision-and-language pre-training. *Advances in Neural Information Processing Systems*, 34:4514–4528, 2021. 2

[62] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3333–3343, 2022. 8

[63] Chaoning Zhang, Chenshuang Zhang, Junha Song, John Seon Keun Yi, Kang Zhang, and In So Kweon. A Survey on Masked Autoencoder for Self-supervised Learning in Vision and Beyond, July 2022. arXiv:2208.00173 [cs]. 2

[64] Chaoning Zhang, Kang Zhang, Trung X Pham, Axi Niu, Zhinan Qiao, Chang D Yoo, and In So Kweon. Dual temperature helps contrastive learning without many negative samples: Towards understanding and simplifying moco. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14441–14450, 2022. 2

[65] Chaoning Zhang, Kang Zhang, Chenshuang Zhang, Trung X. Pham, Chang D. Yoo, and In So Kweon. How does simsiam avoid collapse without negative samples? a unified understanding with self-supervised contrastive learning. In *International Conference on Learning Representations*, 2022. 2

[66] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3