Efficient Video Representation Learning via Masked Video Modeling with Motion-centric Token Selection

Sunil Hwang 1* Jaehong Yoon 1* Youngwan Lee 1 2 Sung Ju Hwang 1
1KAIST 2ETRI
{sunilhoho, jaehong.yoon, ywlee88, sjhwang82}@kaist.ac.kr

Abstract

Self-supervised Video Representation Learning (VRL) aims to learn transferrable representations from uncurated, unlabeled video streams that could be utilized for diverse downstream tasks. With recent advances in Masked Image Modeling (MIM), in which the model learns to predict randomly masked regions in the images given only the visible patches, MIM-based VRL methods have emerged and demonstrated their potential by significantly outperforming previous VRL methods. However, they require an excessive amount of computations due to the added temporal dimension. This is because existing MIM-based VRL methods overlook spatial and temporal inequality of information density among the patches in arriving videos by resorting to random masking strategies, thereby wasting computations on predicting uninformative tokens/frames. To tackle these limitations of Masked Video Modeling, we propose a new token selection method that masks our more important tokens according to the object’s motions in an online manner, which we refer to as Motion-centric Token Selection. Further, we present a dynamic frame selection strategy that allows the model to focus on informative and causal frames with minimal redundancy. We validate our method over multiple benchmark and Ego4D datasets, showing that the pre-trained model using our proposed method significantly outperforms state-of-the-art VRL methods on downstream tasks, such as action recognition and object state change classification while largely reducing memory requirements during pre-training and fine-tuning.

1. Introduction

Massive amounts of video data flood the web and media every single day with the rapid growth of portable devices equipped with cameras, such as google glasses, smartphones, UAVs, and robots. However, direct utilization of user-generated video data for solving target task problems is nontrivial as the annotation process is time-consuming and expensive. One potentially reasonable approach to tackle this problem is to learn generic representations from unlabeled video data streams that can transfer to downstream video-based tasks. Unsupervised Video Representation Learning [13,17,30,32,33] methods allow learning spatial and temporal features from input video frames in a self-supervised manner without any human-annotated clues. A caveat to such pre-training for video tasks is that, unlike representation learning on image-based tasks capturing static information of objects in instances, video-based tasks involve temporal causality, where successive frames are strongly correlated in their semantics.

Recently, a nascent self-supervised learning paradigm, Masked Image Modeling (MIM) [18,42], significantly outperforms the previous representation learning methods on various downstream tasks. MIM aims to predict masked regions in input images by solving a pixel regression problem, in which the model splits each image into small patches and generates masks to zero out a subset of patches. Most of these approaches use Vision transformer backbones [9,25] due to their compatibility with patchwise operations.
Such a simple and intuitive strategy of MIMs was also exploited in video representation learning. Masked Video Modeling (MVM) [12, 38], which learns to reconstruct randomly masked spatiotemporal areas in video clips at each minibatch, has recently shown impressive performance on a variety of downstream tasks. Yet, critical challenges still remain in efficiently utilizing spatiotemporal information from real-world videos: (1) Tokens (a pair of two temporally successive patches in the same space) from videos are not equally valuable to reconstruct, as their informativeness depends on adjacent frames, unlike in image representation learning. (2) Training on the entire video is almost infeasible without access to a large amount of compute. Prior MVM approaches that reconstruct the whole video heavily rely on numerous GPUs without considering each object’s changes and redundancy. For example, VideoMAE [38] takes about 27 hours to pre-train for 800 epochs with a ViT-B backbone using 64 NVIDIA V100 (32GB) GPUs, and MAE [12] takes about 35.8 hours to pre-train for 800 epochs with a ViT-L backbone using 128 NVIDIA A100 (80GB) GPUs.

To tackle these inefficiencies for efficient video representation learning, we propose a token-selective training algorithm, which we refer to as Motion-centric Token Selection (VideoMS). We aim to selectively find helpful visual tokens based on the distance across adjacent tokens in the embedding space to detect significant changes in objects or status while discarding tokens for redundant frames and meaningless backgrounds as shown in Figure 2(c). Thus, VideoMS drastically reduces memory usage by back-propagating only to a few selected tokens of high importance as shown in Figure 1. Our method uses 5.3 times smaller memory (81% ↓) with the effective batch size of 256 and allows to use up to 4 times larger batch size using NVIDIA A100 GPUs compared to VideoMAE [38] (Please see Table 1). Further, most existing VRL methods load video frames within uniform distances in each minibatch clip, neglecting high diversity over frames and noise in uncurated videos. To enhance frame selection, we extend our tokenwise masking strategy to the frame level, constructing informative minibatch video clips by inducing to gather frames that contain various changes over time.

• We propose an efficient video self-supervised learning method that promptly selects the most informative tokens based on the objects’ motion, to avoid wasteful training on uninformative spatiotemporal regions in the given videos.

• We demonstrate that our proposed method outperforms existing SoTA VRL methods, achieving remarkable performance improvements while significantly reducing the computational costs and memory requirement.

• We further suggest an online video clip selection to instantly and adaptively discard noisy and redundant frames from incoming videos, leading the model to help learn objects’ dynamic movements.

![Figure 2. Illustration of Motion-centric Token Selection. (a-b) MVM methods forward a few patches in arriving video frames into the encoder based on fully random and time-only random selection, and then the decoder reconstructs entire frames. (c) Unlike prior works, our proposed motion-centric masking strategy only recovers key tokens related to moving objects (visible). To do this, we use encoder features extracted from randomly selected tokens among them (visible and colored).](image-url)
2. Related Work

A video stream consists of a massive amount of correlated frames exhibiting spatial and temporal locality. Annotating video data requires significant human efforts, and thus it is impractical to fully label all the frames due to extremely high cost from both time and financial standpoints. To capture spatiotemporal representation from video datasets without any given labels, many recent works have focused on the video representation learning problem from diverse perspectives.

Contrastive Video Representation Learning One of the main research directions in video representation learning is contrastive learning with various constraints [13, 16, 17, 33, 39]. Odd-one-out learning [13] learns to distinguish odd video subsequences which are composed of permuted order of frames against normal video streams in a self-supervised fashion. Han et al. [16] introduce the Dense Predictive Coding (DFC) framework for video self-supervised learning that predicts future frame representation from current spatiotemporal features from 3D-Resnet. They adopt a modified contrastive loss, considering pairs with semantic, spatial, and temporal negatives. Motivated by an observation that the human visual system can rapidly and precisely distinguish different motions, Wang et al. [39] propose a new video representation learning method that model learns to predict video pace given a few clustered speed candidates. CoCLR [17] and CVRL [33] tackle video representation learning by proposing modified InfoNCE [29] losses. On the other hand, CoCLR collaboratively trains RGB and optical flow models with shared positives, and CVRL carefully utilizes various augmentation technologies, which is crucial for capturing spatiotemporal representation from video data.

Masked Video Modeling Self-supervised learning with Masked Image Modeling (MIM) [18, 20, 42] has recently become a popular alternative to contrastive learning for its ability to learn rich representations without having to define negative examples. MIM aims to learn image representations by solving pixel regression problems in regions of an image that are zeroed out through random [18, 42] or attention-based [20] masking strategies. Inspired by MIM, several recent works on video representation learning [12, 38, 41, 46] present spatiotemporal masking strategies for representation learning given video streams. To capture spatial representation and temporal dynamics for unsupervised video streams, MAE [12] and VideoMAE [38] extend a masked image autoencoder to mask partial regions in arriving video clips via random and space-only random sampling, respectively. They find that spatiotemporal inductive bias in video clips helps a decoder predict input pixels in masked regions, allowing a higher masking ratio (∼90%) than MIM (∼60% [42] and ∼75% [18]) on image self-supervised learning. BEVT [41] proposes to jointly train an image-level masked autoencoder and a masked video modeling method by sharing weights of the encoder, formulated with Video Swin [26]. They resort to random sampling given spatiotemporal inductive bias, which can be a good approximator with stochasticity during data-driven training. Nevertheless, selecting random spatiotemporal tokens to reconstruct for masked video modeling is inefficient since embedded tokens in video frames are not equally important, especially since the informativeness of each token is affected by the adjacent frames.

Input Selective Training As training benchmark datasets often have massive scales and contain a lot of redundant and less meaningful instances, various approaches [3, 11, 31, 45] have been developed to utilize only the important instances among entire datasets. A few works focus on instance-level selection for training. Borsos et al. [3] presents a bilevel optimization method to select coreset from arriving stream data. OCS [45] proposes a gradient-based coreset selection criterion that selects only the beneficial instances for a task among incoming inputs, dropping redundant and noisy instances in an online fashion. Several recent works have discussed patch selection techniques for efficient image recognition with Vision Transformers (ViT). AdaVit [28] introduces parametric decision networks for patch/head/block selection for training ViT. ATS [11] samples meaningful tokens based on the significant score, computed by multiplication of attentions and value matrices. AttMask [20] and A-Vit [44] adopt parameter-free token masking strategies utilizing the average of multi-head attentions. However, those methods have no means to capture the temporal relatedness across frames in video tasks. Very recently, Wang et al. [40] propose an efficient fine-tuning framework for video tasks that sequentially perform temporal and spatial token selection given pre-trained ViT backbones. They introduce a lightweight scorer network to learn the importance score at each token and select the most informative regions in incoming videos. Park et al. [31] suggest the greedy K-center search that iteratively finds the most distant patches in the geometrical space from video clips. These approaches limit their selection processes to fine-tuning and are not suitable for pre-training. On the other hand, our proposed VideoMS drastically reduces the required computational cost and memory from both stages through motion-centric token selection.

3. Video Representation Learning via Masked Video Modeling

In this section, we first introduce Masked Image Modeling (MIM) for image representation learning, and then extend it to our video representation learning scheme, Masked Video Modeling (MVM) (Sections 3.1 and 3.2). Then we discuss several crucial challenges in recently proposed masked video modeling methods in Section 3.3.
3.1. Masked Image Modeling

Learning to reconstruct intentionally corrupted data with masking is broadly utilized as means of representation learning in Natural Language Processing [7, 15, 34, 35] and has demonstrated its efficacy and power in broad research problems. In vision tasks, Masked Image Modeling [18, 42] aims to learn representations of the input images by solving the regression problem in which a model predicts RGB pixel values in randomly zeroed patch regions in images. The model first splits a high-resolution input image into equally-sized square patches and fills the values of a few patches to be zeros according to a pre-defined masking ratio. A transformer encoder extracts visual features from unmasked patch tokens with positional information, and a decoder tries to reconstruct the original input images by predicting the RGB values of missing image patches from encoded features. Let a self-supervised model be formulated into an encoder-decoder framework . Given the dataset , where is an arbitrary random masking function depending on the specific policy, for example, random (or agnostic) and space-only (or tube) masking techniques, illustrated in Figure 2 (a) and (b).

Unlike the samples from the image dataset, which are permutation-invariant as they are independent of each other, consecutive frames from the video stream inherently have a strong correlation and redundancy. Thus, masked video modeling can enjoy more spatiotemporal inductive bias from other adjacent frames in the input clip, achieving good reconstruction quality even with lessened hints (i.e., a proportion of unmasked tokens). Indeed, MVM allows a much higher masking ratio per frame against MIM methods. This property is advantageous because masked modeling with a higher masking ratio significantly reduces computations when training the encoder-decoder framework.

3.3. Challenges in Masked Video Modeling

Recently proposed masked video modeling methods capture meaningful representations from pre-training videos via randomly masking spatiotemporal patches from input clips. Random sampling-based masking strategies are reasonable for curated and distributionally-stable video datasets. Yet, there is plenty of room for further development to make the model much more robust and computation-efficient.

Here, we summarize two major limitations in the random sampling-based Masked Video Modeling scenario: (1) Tokenized image patches from a video clip are not equally important. At each iteration, MVM methods determine which patches to mask according to specific random selection strategies (e.g., random, space-only, time-only, ...). Yet, the relative amount of information in each patch highly depends on the position of the informative objects and the correlation across patches within adjacent frames, which renders most of the patches highly uninformative or redundant. (2) MVMs uniformly collect discretized frames from video streams to construct minibatch clips. Real-world videos may include noisy and highly redundant frame sequences, often uninformative or even detrimental in representing temporal causal relationships and features of moving objects. However, the uniform sampling of video frames results in a dissipation of computational and memory resources in video representation learning. They consume massive training budgets in memory occupancy and suffer from slow convergence speed (e.g., training 3, 200 and 4, 800 epochs on UCF101 [36] and
Video tokens, $v_{[2t]} \in \mathbb{R}^{2T \times C \times H \times W}$, where an index $0 \leq i < \tau$. When $k_{i+1,j}$ indicates the $j^{th}$ token embedding of $k_{i+1}$, we compute the importance $I_{i+1,j}$ of the token $k_{i+1,j}$ by computing the distance from the token embedding of the same spatial position in the previous time step, $k_{i,j}$:

$$I_{i+1,j} = S(k_{i+1,j}, k_{i,j}),$$

where $k_i = \text{Conv3d}(v_{[2i:2i+1]}; w)$, (3)

$S(\cdot, \cdot)$ indicates a distance function, such as Euclidian distance, negative Cosine Similarity, and negative Centered Kernel Alignment (CKA) [5]. Throughout this paper, we use the $\ell_2$ norm for the distance function, which is simple yet empirically performs well. After computing the change in spatial information which means the informative tokens as the video progresses, we perform a two-step mask sampling. Given $v$, the model determines the token embedding vectors with the highest importance ratio of $\rho_{\text{pre}}$ based on Equation 3. This process is called motion-centric masking generation (Depicted in Figure 3) and we use $30\%$ ($\rho_{\text{pre}} = 0.3$) of the total of $J \times \tau$ token embeddings for all experiments. Since only the selected embedding vectors $\hat{k}$ go through the encoder-decoder frameworks, this will drastically reduce the computational cost during training (Please see Figure 2 (c) and 'Sparsified Video' in Figure 3). Second, we generate binary masks via random sampling for $\hat{k}$ and the model forwards drawn embedding vectors with the ratio of $\rho_{\text{post}}$ into the encoder. A corresponding decoder predicts RGB pixels on all embedding vectors $\hat{k}_i \in \hat{k}$ from the encoder output features. We set a high masking ratio in general, $\rho_{\text{pre}} \cdot \rho_{\text{post}} = 0.1$, to follow the settings of our MVM baselines. To this end, the objective function of our VideoMS is
We dynamically sample the given video by utilizing our video-based learning methods construct minibatch clips by videos and only processes them for solving downstream task training iteration, a model selects a random frame in each sampling evenly spaced frames from arriving videos. At each iteration since longer clips require massive computations and memory. Therefore, we remark that adaptive frame selection is crucial to better capture causality in the arrival video, as the model can observe longer video fragments while avoiding redundant frames.

5. Experiments

Experimental Settings We validate our VideoMS method on multiple video datasets: UCF101 [36], HMDB51 [22], and Ego4d [14]. UCF101 and HMDB51 are mainly used datasets for video action recognition, which consist of 9.5k/3.5k and 3.5k/1.5k videos, respectively. We evaluate our adaptive frame selection strategy on the Ego4d dataset, which consists of egocentric videos containing raw and uncurated people’s daily lives, by applying it during pre-training. We validate our methods through the Object State Change Classification (OSCC) 1 task from Ego4d. Given 8-second videos, OSCC classifies whether the object’s state has changed due to interaction with a camera wearer.

Implementation details We follow the implementation details and experimental settings from prior works [12, 38] for fair comparisons. We set the size of an input clip \( \tau = 16 \), tokenization size \( 2 \times 16 \times 16 \), and the fixed sampling stride of 4 for UCF101, and 2 for HMDB51, respectively. For our proposed method, VideoMS, \( \rho_{pre} \) is set to 0.3 and 0.6 for pre-training and fine-tuning, respectively, and we simply adopt Joint Space-Time Attention [2]. Please see the supplementary file for more details.

---

1 https://github.com/EGO4D/hands-and-objects/tree/main/state-change-localization-classification
Figure 5. Examples of Motion-centric Token selection. We show the original video frames (left), Motion-centric masking results (middle), and obtained importance heatmaps (right) on UCF101 [36] ((a) and (b), \( \rho_{pre} = 0.3 \)) and Ego4d [14] ((c), \( \rho_{pre} = 0.5 \)) datasets. Curated fixed-view (third-person view) videos, (a) and (b), show narrow and concentrated motion information as shown in their heatmaps, whereas motion cues in egocentric videos (c) are distributed and contain multi objects.

Table 2. Comparison to state-of-the-art methods on UCF101 and HMDB51. We outperform the previous method, including the masking-based video model, without using the pre-training step on a large-scale dataset. These results are drawn from [8] and [38].

Quantitative Evaluation

We first extensively compare our proposed method against strong VRL baselines. We pre-train the VRL methods over benchmark datasets without labels and report the fine-tuning performance on the same datasets in Table 2. Our masked video modeling with motion-centric token selection achieves superior fine-tuning performance against baselines across all benchmark tasks. Specifically, it outperforms the best baseline by 2.1% \( \uparrow \) on UCF101 and 3.2% \( \uparrow \) on HMDB51, which demonstrates the effectiveness of focusing on the objects’ motions for video representation learning. For the UCF101 dataset, we visualize the convergence plot of VideoMS in Figure 1 Left, which shows that it reaches a higher fine-tuning performance than VideoMAE [38] by pre-training at only 800 epochs. We provide masked input examples with their importance heatmaps through our proposed motion-centric masking in Figure 5. Our masking strategy enables the model to capture the most critical tokens containing static and dynamic moving of objects (e.g., doing wall push up (Figure 5 (a)) and playing ping pong (Figure 5 (b))). Furthermore, our masking strategy captures informative objects even with the rapid change of the view in the first-person videos, as it masks out objects which are crucial for understanding the camera wearers’ attention (Figure 5 (c)), while not attending to backgrounds such as walls and floors.

The Impact of Motion-centric Token Selection

One of our major contributions is the significantly enhanced computational efficiency during VRL, as we process a few latent vectors in the decoder to reconstruct only the motion-activated tokens in the given videos according to the motion-centric masking ratio \( \rho_{pre} \) (Please see Section 4.1). We set \( \rho_{pre} \) to 0.3 so that our VideoMS reconstructs the 30% of the essential spatiotemporal regions focusing on objects’ movements and behaviors, from a sparsified video clip, which reduces 65.5% of GFLOPs at the pre-training phase (Please see Figure 1 Right and Table 3 Left). Additionally, as reported in Table 3 Right, our approach can reduce the computational overhead at the fine-tuning phase, which is practically useful when transferring the learned representation learning model to downstream video tasks.
We process only 60% of tokens ($\rho_{VRL}$ aims to capture spatiotemporal features in incoming videos. But when real-world videos include many redundant scenes over a long time, the model may waste time and computations by learning on meaningless frames, which also can lead to poor local optima due to bias and catastrophic forgetting. To overcome the limitation, we adopt the ratio of motion-centric tokens $\alpha$ to focus on the frames

| Method          | Motion-centric Masking | Fine-tuning Acc. top-1 |
|-----------------|------------------------|------------------------|
| VideoMS (Ours)  |                        |                        |
| ascending       | ascending              | 60.93                  |
| descending      |                        | 75.60                  |
| descending      |                        | 91.56                  |

Table 4. The impact of informative-token selection. It shows better performances both in pre-training and fine-tuning when the model prioritizes selection with the most far-distance (descending) embedded tokens rather than near-distance (ascending) tokens.

Next, we emphasize that our motion-centric masking is also applicable when solving video-based downstream tasks. We process only 60% of tokens ($\rho_{pre} = 0.6$) of the given video during fine-tuning. Surprisingly, our motion-centric masking for supervised downstream tasks gains increased performance than our variant without masking on the fine-tuning tasks ($\rho_{pre} = 1.0$) using only about 55% of GFLOPs, as shown in Table 3 Right. The results support our hypothesis that video data often contain redundant information and our selective video learning using the proposed masking strategy successfully captures essential space-time regions in video inputs, allowing us to focus more on learning spatiotemporally meaningful features. The fine-tuning masking ratio is higher than the ratio at the pre-training stage, showing that the model uses more information to fully exploit the task-relevant cues from the given videos compared to pre-training, which aims to obtain general information.

Also, we examine the impact of the informativeness of the tokens in Table 4. As explained in Section 4.1, we selectively learn informative tokens by calculating the distance between tokens in adjacent frames in the embedding space, as the tokens farther away from adjacent frames contain less redundant information. When we reversely select near-distance tokens, it drastically decreases the accuracy in pre-training and fine-tuning, obtaining the worst performance when reversely selecting informative tokens in both phases.

Adaptive Frame Selection for Uncurated Video Streams

VRL aims to capture spatiotemporal features in incoming videos. But when real-world videos include many redundant scenes over a long time, the model may waste time and computations by learning on meaningless frames, which also can lead to poor local optima due to bias and catastrophic forgetting. To overcome the limitation, we adopt the ratio of motion-centric tokens $\alpha$ to focus on the frames with larger motions across frames. As shown in Table 5 Left, downstream task performance with adaptive frame selection results in superior performance to baselines. By constructing the given videos to have fluent motion information, our model focuses more on learning the core part of the video selected by our motion-centric masking ratio $\rho_{pre}$. Furthermore, we compare our method to recent SoTA methods over OSCC dataset in Table 5 Right. Our model pre-trains OSCC without labels and fine-tunes it to classify whether the object’s state has changed. With only visual information, we outperform the previous SoTA method Egocentric VLP [24], which use visual and text information, by 2.3% $p \uparrow$.

6. Conclusion

In this paper, we propose a simple yet efficient parameter-free token/frame selection method for masked video representation learning. Our method is based on the intuition that not all patches in the given video are equally informative. We adaptively select the most useful spatiotemporal informative patches based on objects’ moving and train only crucial motion-centric tokens with the sparsified video clips, drastically reducing memory allocation and computational cost. In addition, we propose a frame selection technique to construct input video data by sampling incoming frames with a probability proportional to the degree of the occupancy of motion-centric tokens. This strategy can be useful for video representation learning with uncurated videos such as egocentric videos from first-person cameras. The experimental results show that our method is significantly more efficient in computations and memory, reaching the target performance with a much smaller number of training epochs than the baselines and achieving substantially higher performance when trained for the same number of epochs.

| $\alpha$     | Acc. (%) | Method     | Modality | Acc. (%) |
|--------------|----------|------------|----------|----------|
| -            | 73.17    | Egocentric VLP [24] | V+T      | 73.9     |
| 1.3          | 73.54    | SViT [10]  | V        | 69.8     |
| 1.5          | 73.85    | TarHeels [19] | V        | 70.8     |
| 1.8          | 73.79    |            |          |          |
| 2.0          | 73.80    | VideoMS (Ours) | V        | 76.2 ($+5.4\% p$) |

Table 5. Effect of Motion-centric Sampling ratios (Left) and Comparison to public SoTA methods for OSCC on val set (Right). We set the default motion-centric sampling rate ($\alpha = 1.5$) and report fine-tuning accuracy. Pre-trained data modalities from the OSCC dataset ‘V’ and ‘T’ refers to visual and text.
References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 12

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all we need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 6

[3] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 12

[5] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 2012. 5, 13

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 12

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4

[8] Ali Diba, Vivek Sharma, Reza Safdari, Dariush Lotfi, Saqib Sarfarz, Rainer Stiefelhagen, and Luc Van Gool. Vi2clr: Video and image for visual contrastive learning of representation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[10] Maria Escobar, Laura Daza, Cristina González, Jordi Pont-Tuset, and Pablo Arbeláez. Video twin transformers for egocentric video understanding@ ego4d challenges 2022. *arXiv preprint arXiv:2207.11329*, 2022. 8

[11] Mohsen Fayyaz, Soroush Abbasi Kouhpayegani, Farnoush Rezaei Jafari, Eric Sommerlade, Hamid Reza Vaezi Joze, Hamed Pirsivash, and Juergen Gall. Ats: Adaptive token sampling for efficient vision transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[12] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2, 3, 5, 6, 7, 12, 13

[13] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 3

[14] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 6, 7, 12

[15] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasuapat, and Mingwei Chang. Retrieval augmented language model pre-training. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 4

[16] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019. 3

[17] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 3, 7

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3, 4

[19] Md Mohaiminul Islam and Gedas Bertasius. Object state change classification in egocentric videos using the divided space-time attention mechanism. *arXiv preprint arXiv:2207.11814*, 2022. 8

[20] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Pomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[21] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The
kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017. 6

[22] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomas Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In Proceedings of the International Conference on Computer Vision (ICCV), 2011. 5, 6, 12

[23] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In Proceedings of the International Conference on Computer Vision (ICCV), 2017. 7

[24] Kevin Qinghong Lin, Alex Jinfeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenhao Jiang, Weijie Kong, et al. Egocentric video-language pretraining. arXiv preprint arXiv:2206.01670, 2022. 8

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 1

[26] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 3

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, 2016. 12

[28] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 3

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018. 3

[30] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 1

[31] Seong Hyeon Park, Jihoon Tack, Byeongho Heo, Jung-Woo Ha, and Jinwoo Shin. K-centered patch sampling for efficient video recognition. In Proceedings of the European Conference on Computer Vision (ECCV), 2022. 3

[32] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 1

[33] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 1, 3

[34] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. arXiv preprint arXiv:1905.02450, 2019. 4

[35] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 4

[36] 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. 1, 4, 6, 7, 12, 14

[37] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 12

[38] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In Advances in Neural Information Processing Systems (NeurIPS), 2022. 1, 2, 3, 4, 5, 6, 7, 12, 13

[39] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In Proceedings of the European Conference on Computer Vision (ECCV), 2020. 3

[40] Junke Wang, Xitong Yang, Hengduo Li, Liu Li, Zuxuan Wu, and Yu-Gang Jiang. Efficient video transformers with spatial-temporal token selection. In Proceedings of the European Conference on Computer Vision (ECCV), 2022. 3

[41] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiayi Dai, Mengchen Liu, Yu-Gang Jiang, Luyue Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 3

[42] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Sim-
mim: A simple framework for masked image modeling. arXiv preprint arXiv:2111.09886, 2021. 1, 3, 4

[43] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 7

[44] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 3

[45] Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. In Proceedings of the International Conference on Learning Representations (ICLR), 2022. 3

[46] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 3
Supplementary Materials

Organization The supplementary file is organized as follows: Firstly, we explain the implementation details on datasets and tasks in Appendix A and perform the distance function ablation study for motion-centric masking during pre-training in Appendix B. Then we provide the additional experimental results for our VideoMS in Appendix C. Finally, we visualize additional examples from VideoMS in Appendix D.

A. Implementation Details

|                      | UCF101 | HDMB51 | OSCC |
|----------------------|--------|--------|------|
| optimizer            | AdamW  |        |      |
| optimizer momentum   | β₁, β₂ = 0.9, 0.95 [4] |        |      |
| base learning rate   | 1e-3   |        |      |
| weight decay         | 0.05   |        |      |
| learning rate schedule | Cosine decay [27] |        |      |
| flip augmentation    | yes    |        |      |
| augmentation         | MultiScaleCrop |        |      |
| batch size           | 192    | 192    | 256  |
| warmup epochs        | 40     | 40     | 20   |
| ρ_pre                | 0.3    | 0.3    | 0.5  |
| sampling stride      | 4      | 2      | 4    |
| total epochs         | 3200   | 4800   | 400  |

Table 6. Pre-training settings for UCF101, HDMB51 and OSCC.

|                      | UCF101 | HDMB51 | OSCC |
|----------------------|--------|--------|------|
| optimizer            | AdamW  |        |      |
| optimizer momentum   | β₁, β₂ = 0.9, 0.999 |        |      |
| weight decay         | 0.05   |        |      |
| learning rate schedule | Cosine decay [27] |        |      |
| warmup epochs        | 5      |        |      |
| laer-wise lr decay   | 0.75 [1] |        |      |
| flip augmentation    | yes    |        |      |
| RandAug              | (9, 0.5) [6] |        |      |
| label smoothing      | 0.1 [37] |        |      |
| drop path            | 0.1    |        |      |
| base learning rate   | 1e-3   | 1e-3   | 1e-4 |
| batch size           | 128    | 128    | 32   |
| ρ_pre                | 0.6    | 0.6    | -    |
| sampling stride      | 4      | 2      | 10   |
| total epochs         | 100    | 50     | 30   |
| multi-view           | 5×3    | 10×3   | 2×3  |

Table 7. Fine-tuning settings for UCF101, HDMB51 and OSCC.

As we mentioned in Section 5 of the main paper, we validate our VideoMS on three video datasets: UCF101 [36], HMDB51 [22], and Ego4d [14]. We provide the hyperparameter setup for pre-training and fine-tuning in Table 6 and Table 7, respectively. As we mentioned in Section 4.1 of the main paper, we adopt the same masking ratio with VideoMAE [38] and MAE [12] for masking input video, but we only reconstruct (ρ_pre × 100)% of input video tokens during pre-training. And in fine-tuning, our model only takes (ρ_pre × 100)% of input video tokens. We follow the linear learning rate scheduling of [38] and [12], \[ lr = base\text{learning}\_rate \times \frac{batch\_size}{256}. \]

Video Action Recognition UCF101 and HDMB51 are used to evaluate our method for the video action recognition task. We pre-train ViT-B on the training set of each targeted downstream task. During the inference, we adapt common multi-view testing, T clips × three crops, which takes T temporal clips with three spatial crops to cover the overall length and space of the video, and the final decision is made by averaging the results of each view.

Object State Change Classification(OSCC) The OSCC dataset is the subset of the Ego4d dataset, consisting of 41.1k/21.2k train/val 8-second videos. As shown in Table 5 (Left) of the main paper, we evaluate our adaptive frame selection strategy during pre-training by comparing the downstream tasks’ accuracy, varying the ratio of α from 1.0, which is equal to not using α denoted as hyphen, to 2.0. Note that, as the Ego4d dataset shows the characteristic of having motion cues that are distributed and containing multi objects in Figure 5 (c) of the main paper, we adopt ρ_pre = 0.5. To see the effect of the frame selection range α, in Table 5 (Left), we train ViT-B on the train set of OSCC for 100 and 10 epochs in pre-training and fine-tuning, respectively. On the other hand, in Table 5 (Right), we train 400 and 30 epochs in pre-training and fine-tuning phases, respectively, using α = 1.5 only in pre-training. We adopt a fixed sampling ratio of 4 and 10 in pre-training and fine-tuning, respectively.

B. Comparing Functions for Computing Patch Embedding Distance

| Distance Function   | accuracy(%) |
|---------------------|-------------|
| negative cosine     | 33.53       |
| negative CKA        | 34.58       |
| L1                  | 42.22       |
| L2                  | 42.81       |

Table 8. Fine-tuning results on HMDB51 measured by varying the distance function in pre-training.

https://github.com/MCG-NJU/VideoMAE
As mentioned in Section 4.1 of the main paper, we can adapt other distance functions: L1 and L2 distance, negative cosine similarity, and negative CKA [5]. We train ViT-B on HMDB51 for 200 and 50 epochs, respectively, only varying the distance function in pre-training as shown in Table 8. L1 and L2 show the superior accuracy against the others. From these results, we use the L2 distance function as our default function for computing patch embedding distance.

C. Additional Experimental Results

As shown in Table 1 and 5 of the main paper, we save the computational cost drastically in pre-training and fine-tuning. Therefore, we conduct a further experiment in Table 9 by increasing the number of input frames from 16 to 24 in pre-training and fine-tuning to observe the results in case of having similar computational costs with VideoMAE. We achieve 94.42% accuracy on UCF101 while still having relatively lower GFLOPs than VideoMAE [38].

Here, we show the practical results comparing the GPU memory usage and corresponding batch size of each masking-base video models in Table 10. From this result, we strongly believe that our method would be the efficient implementation baseline for those who don’t have many GPU resources in contrast to the previous works [12, 38].

| Method   | # of frames | GFLOPs | accuracy(%) |
|----------|-------------|--------|-------------|
|          | pre  | fine | pre  | fine |          |
| VideoMAE | 16   | 16   | 35.48 | 180.5 | 90.80    |
| VideoMS  | 16   | 16   | 19.81 | 98.1  | 94.42    |
| VideoMS  | 24   | 24   | 30.65 | 159.4 | 93.39    |

Table 9. Fine-tuning results on UCF101 measured by varying the number of frames in pre-training and fine-tuning.

| Method   | Pre-train | Fine-tune |
|----------|-----------|-----------|
|          | Batch size | GPU(G)    | Batch size | GPU(G)    |
| MAE [12] | 16        | 23.2      | 4           | 20.3      |
| VideoMAE† [38] | 24        | 23.6      | 8           | 18.7      |
| Ours†    |           |           |             |           |
|          | 16        | 7.2       | 4           | 6.5       |
|          | 24        | 9.2       | 8           | 9.5       |
|          | 36        | 12.1      | 12          | 14.0      |
|          | 72        | 20.6      | 16          | 17.1      |

Table 10. GPU memory usage comparison of ViT-B between previous Masking-based Video models with respect to batch size.† use Deepspeed frameworks during fine-tuning. These experiments are conducted on the same NVIDIA GeForce RTX 3090 GPU.

D. Visualization of motion-centric masking and sampling

As shown in Figure 5 of the main paper, our masking method successfully captures the most informative space of each frame in various conditions. In this section, we further visualize our masking results in Figure 7 for deeper understanding. The first sample is similar in Figure 5 (b) but changes the $\rho_{pre}$ from 0.5 to 0.15. The masked result shows that our masking strategy could be better and save more computational costs if the view is fixed and the moving object is few. In Figure 8, the masked results in frames of the first to fourth show that our masking strategy wrongly captured the blue line of each frame’s upper regions as informative tokens in the given video. As the heatmaps show that those tokens are relatively low informative than the two skaters in the video, it still makes sense that our masking strategy can successfully capture the informative tokens in the given video. Figure 9 shows that our sampling method takes various frames and ignores redundant frames in the given video during pre-training.
Figure 7. **More Examples of Motion-centric Token selection.** We show the original video frames in the first row, Motion-centric masking results in the second, third, and fourth row by varying the $\rho_{pre}$ to 0.5, 0.25, and 0.15, respectively, and obtained importance heatmaps in the last row on UCF101 [36] dataset.

Figure 8. **More Examples of Motion-centric Token selection.** We show the original video frames in the first row, Motion-centric masking results in the second row, and obtained importance heatmaps in the last row on UCF101 [36] dataset. $\rho_{pre}$ is set to 0.3 in this example.
Figure 9. More Examples of Motion-centric Sampling results. Twenty-four frames of the first row in each sample are the given video, and blue-boxed frames are sampled with our sampling method. Red-boxed frames highly similar to each other would be sampled without our sampling method.