Data-Efficient Masked Video Modeling for Self-supervised Action Recognition

Qiankun Li
HFIPS, Chinese Academy of Sciences
University of Science and Technology of China
Hefei, China
qklee@mail.ustc.edu.cn

Xiaolong Huang
School of Artificial Intelligence, Chongqing University of Technology
Chongqing, China
3579628328@2019.cqut.edu.cn

Zhifan Wan
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
wanzhifan21s@ict.ac.cn

Lanqing Hu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
lanqing.hu@vipl.ict.ac.cn

Shuzhe Wu
Beijing Huawei Cloud Computing Technologies Co., Ltd
Beijing, China
wushuzhe2@huawei.com

Jie Zhang
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
zhangjie@ict.ac.cn

Shiguang Shan
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
sgshan@ict.ac.cn

Zengfu Wang*
HFIPS, Chinese Academy of Sciences
University of Science and Technology of China
Hefei, China
zfwang@ustc.edu.cn

ABSTRACT

Recently, self-supervised video representation learning based on Masked Video Modeling (MVM) has demonstrated promising results for action recognition. However, existing methods face two significant challenges: (1) video actions involve a crucial temporal dimension, yet current masking strategies adopt inefficient random approaches that undermine low-density dynamic motion clues in videos; (2) pre-training requires large-scale datasets and significant computing resources (including large batch sizes and enormous iterations). To address these issues, we propose a novel method named Data-Efficient Masked Video Modeling (DEMVM) for self-supervised action recognition. Specifically, a novel masking strategy named Flow-Guided Dense Masking (FGDM) is proposed to facilitate efficient learning by focusing more on the action-related temporal clues, which applies dense masking to dynamic regions based on optical flow priors, while sparse masking to background regions. Furthermore, DEMVM introduces a 3D video tokenizer to enhance the modeling of temporal clues. Finally, Progressive Masking Ratio (PMR) and 2D initialization strategies are presented to enable the model to adapt to the characteristics of the MVM paradigm during different training stages. Extensive experiments on multiple benchmarks, UCF101, HMDB51, and Mimetics, demonstrate that our method achieves state-of-the-art performance in the downstream action recognition task with both efficient data and low computational cost. More interestingly, the few-shot experiment on the Mimetics dataset shows that DEMVM can accurately recognize actions even in the presence of context bias.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**.

KEYWORDS

flow-guided mask; data-efficient; self-supervised action recognition

ACM Reference Format:
Qiankun Li, Xiaolong Huang, Zhifan Wan, Lanqing Hu, Shuzhe Wu, Jie Zhang, Shiguang Shan, and Zengfu Wang. 2023. Data-Efficient Masked Video Modeling for Self-supervised Action Recognition. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3581783.3612496

1 INTRODUCTION

Action recognition plays a vital role in various applications, including video comprehension [14, 55], human-computer interaction [35], sports content analysis [39], and intelligent surveillance [53]. Since the arduous nature of manual-based video-level annotation, self-supervised learning from unlabeled videos for downstream action recognition tasks has gained significant attention in the field of computer vision [9, 33, 43].

Inspired by the remarkable achievements of BERT[13] in natural language processing, several vision studies, including BEiT [4],

*Corresponding author.

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612496

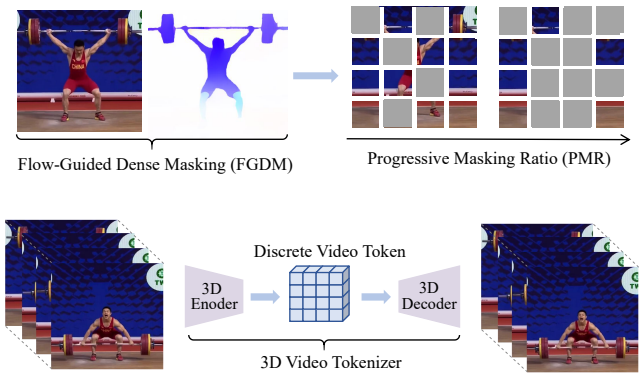

Figure 1: Overview of the main components of DEMVM. The FGDM employs optical flow to guide dense masks for dynamic motion patches, effectively utilizing action-related clues. PMR gradually increases the masing ratio during the training process to achieve efficient MVM pre-training, and the 3D video tokenizer efficiently incorporates spatio-temporal information into the token.

MAE [22], and MaskFeat [49], have demonstrated the efficacy of the Masked Image Modeling (MIM) paradigm for self-supervised representation learning. The MIM paradigm randomly masks some patches at the input image, and the pre-training task is to reconstruct the original image by predicting these masked patches. After pre-training, task-specific heads can be integrated into the encoder to fine-tune downstream tasks.

Following the MIM paradigm, some works [16, 18, 43, 48] have extended it to the field of self-supervised video representation learning, forming a series of Masked Video Modeling (MVM) methods. All these methods have demonstrated remarkable performance for various video understanding tasks, like action recognition. However, they usually suffer from two significant issues. First, the temporal information introduced by videos has not been effectively utilized. The current video masking strategies merely expand the random masking area of 2D images into 3D videos or use spacetime-agnosti [16] random masking for videos, which undermines the importance of motion clues in temporal. Additionally, the discrete visual tokens in MVM are generated independently for each frame using the 2D image tokenizer, such as DALL-E [38] or VQ-VAE [3], which overlooks the continuous frame temporal information in the video. Second, pre-training of MVM necessitates large-scale datasets and significant computational resources, including large batch sizes and vast training iterations. Specifically, VideoMAE [43] entails training for 1600 epochs with batch size of 1024 on Kinetics-400 [7] (about 240k video clips), while OmniMAE [18] demands training for 1600 epochs with batch size of 2048 on Something-Something-V2 [19] (about 169k video clips). The training of these methods necessitates the use of 64 or even more than 128 NVIDIA A100 (80GB) or NVIDIA V100 (32GB) GPUs.

In this paper, we propose a Data-Efficient Masked Video Modeling (DEMVM) for self-supervised action recognition (as shown in Figure 1). To address the inefficiency of existing MVM methods in utilizing temporal clues, DEMVM introduces a novel video masking strategy called Flow-Guided Dense Masking (FGDM). Unlike random masking strategies, FGDM leverages optical flow [5] priors to distinguish dynamic motion patches from static background patches in videos, and conduct dense masks on motion patches while sparse masks on background patches. Since more motion patches are masked, DEMVM is encouraged to exploit temporal clues to well reconstruct the original videos during the pre-training procedure, which is beneficial to downstream action recognition. In addition, DEMVM introduces a 3D video tokenizer from VideoGPT [51]. This 3D tokenizer upgrades the previous 2D VQ-VAE tokenizer [3] through a 3D encoder [44] and axial attention mechanism [26], which helps it model spatio-temporal information when encoding video discrete visual tokens.

To further enhance the efficiency of MVM pre-training, we propose a new Progressive Masking Ratio (PMR) and 2D initialization strategies. The PMR strategy gradually increases the masking ratio during the pre-training procedure, enabling the DEMVM to adapt the MVM paradigm from easy to difficult. This strategy helps the encoder quickly learn compelling features at the high masking ratio. Since the encoder is based on a 3D Swin Transformer [32], the 2D initialization strategy inflates the 2D Swin Transformer [31] to the encoder to obtain a representative initial feature space.

By effectively utilizing action-related temporal clues and carefully designing training strategies, our DEMVM achieves state-of-the-art results on downstream action recognition task. DEMVM pre-trained on UCF101 [41] (only 9.5K videos) and HMDB51 [28] (only 3.5K videos) achieves Top-1 accuracy of 92.2% and 63.9% on their downstream using only pre-training 10 epochs with the batch size of 24. The entire training process only needs 4 NVIDIA 3090 (24G) GPUs and takes only 2 hours. More interestingly, we design a few-shot experiment on the Mimetics dataset [50] without background bias to verify the effectiveness of DEMVM in learning the action-related temporal information. Our DEMVM outperforms the random mask method with an accuracy improvement of up to 6%.

Our main contributions can be summarized as follows:

- We propose FGDM to use optical flow to guide the dense masking on motion patches, thereby efficiently exploring video action-related clues that are usually underestimated by current MVM methods.
- We introduce a 3D video tokenizer to enable MVM to incorporate spatio-temporal information when encoding video-level discrete visual tokens.
- We devise PMR and 2D initialization strategies to combine FGDM and 3D video tokenizer in pre-training, enabling data efficiency for our DEMVM.

2 RELATED WORK

2.1 Self-supervised Video Representation Learning

Existing self-supervised techniques for video representation learning commonly utilize three types of pretext tasks: context-based learning methods, contrastive learning methods, and generative MVM [20]. The earlier context-based methods concentrate on developing pretext tasks that exploit contextual relationships within videos, such as speed prediction [24], pace prediction [46], and jigsaw solving [27]. Despite promising results in utilizing unlabeled data, these methods often fail to model complex temporal action

relationships. Contrastive learning methods [6, 37, 47] have more recently focused on constructing positive and negative examples without annotations and conducting contrastive learning to bring positive examples closer to each other than negative ones. However, since the learning supervision based on contrastive learning is applied on global representations, it cannot effectively model the local relationship in action recognition. Some contrastive methods [1, 40, 45] employ multi-modality information to improve the representation in visual encoders, with the objective of extracting more action-specific video features. For instance, P-HLVC [40] contrasts skeleton pose features with video features to comprehend fine-grained human actions, whereas the method [45] includes optical flow features to enhance temporal correlations for the action recognition task. Nevertheless, these approaches depend on large batch sizes to introduce more diversity into each batch, allowing the model to contrast with a wider range of video samples. Furthermore, multi-modal methods require sub-stream encoders to comprehend different modalities, resulting in large model parameters. Consequently, contrastive learning methods that aim to enhance action recognition generally rely on large-memory computation with additional fusion frameworks. Given the success of visual transformers and their natural compatibility with masked prediction tasks, the MVM methods are beginning to demonstrate their potential for video representation learning, which are detailedly illustrated in the following section.

2.2 Masked Video Modeling

Masked Language Modeling (MLM) plays a leading role in the NLP field, among which BERT [13] and RoBERTa [30] have become baseline methods for various NLP tasks. Inspired by this, many efforts [4, 22, 49] are devoted to Masked Image Modeling (MIM) for self-supervised representation learning. BEiT [4] uses the 2D tokenizer Dall-E [38] to get the discrete visual tokens of the random mask input and learn to restore it. MAE [22] considers MIM as a denoising pixel-level reconstruction task. MaskFeat [49] chooses to restore the Histograms of Oriented Gradients (HOG) [10] of the masked regions. The encoders pre-trained by these methods finally achieve satisfactory performance on downstream image recognition tasks.

Recently, novel approaches [16, 18, 43, 48] to the MVM paradigm have emerged in the realm of video self-supervision through direct extensions to the MIM paradigm. These methods have exhibited outstanding performance in self-supervised action recognition. OmniMAE [18] employs a fully random masking scheme for patches in videos. BEVT [48] utilizes a spatial random block-wise masking initially and then directly expands it along the video dimension into a tube masking. VideoMAE [43] accounts for the video's information redundancy while employing tube masking, resulting in an increased mask ratio. MAE-ST [16] suggests using space-time-agnostic random masking for video. Nonetheless, these random variant masking strategies are a simple yet inefficient extension of the random masking strategy in images to video, which underestimates low-density action clues in the temporal dimension. Unlike the above-mentioned masking strategies, our DEMVM employs optical flow [5] to guide dense motion patch masking, thereby effectively exploring video action clues. In addition, most MVM methods [16, 18, 43, 49] extend image pixel-level prediction to video, but video pixel-level reconstruction requires vast data, and is difficult to learn holistic spatio-temporal information. BEVT [48] predicts the discrete visual tokens by generating them through the image VQ-VAE tokenizer [3]. The tokenizer encodes each frame of the video clip independently into a 2D token, which is then merged. However, this encoding procedure overlooks the temporal information of consecutive frames in the video. To tackle this issue, our DEMVM introduces a 3D video tokenizer [51] to replace the previous 2D image tokenizer, which can encode video-level discrete visual tokens with temporal information. Ultimately, existing MVM methods rely on large-scale datasets [7, 19, 34] and significant computing resources (large batch sizes and vast iterations). Due to the efficient combination of temporal action clues and training strategy, our DEMVM achieves exceptional results on several popular small datasets for action recognition using only 4 NVIDIA 3090 (24G) GPUs.

3 METHOD

In this section, we firstly present an overview of our DEMVM, which encompasses the MVM pre-training procedure and downstream action recognition. Furthermore, we introduce the details of our Flow-Guided Dense Masking (FGDM), which efficiently guides MVM to mine temporal action clues. We then illustrate a 3D tokenizer, which enables MVM to incorporate temporal information when encoding video-level discrete visual tokens. Lastly, we describe our efficient training strategies, including Progressive Masking Ratio (PMR) and 2D initialization.

3.1 Framework Overview

The structure of the DEMVM is presented in Fig 2. DEMVM is pretrained by the MVM task in a self-supervised learning manner.

Flow-Guided Dense Masking. Given an input video clip $X \in \mathbb{R}^{T\times H\times W\times 3}$, where T represents the length of the video, H and W denote the height and width of the video, and 3 represents the number of channels (RGB). To enhance the learning of temporal action information, Flow-Guided Dense Masking (FGDM) is proposed to mask more on patches of dynamic motions while mask fewer on background patches. Specifically, optical flow points are utilized to locate the regions of dynamic motions. Since more information is masked around dynamic regions, DEMVM is enforced to learn more temporal action features to well reconstruct the original videos during pre-training procedure. Subsequently, FGDM randomly selects frames to be masked in the temporal dimension. In addition, the Progressive Masking Ratio (PMR) strategy is proposed to gradually increase the temporal and spatial masking ratio according to the number of iterations during training.

DEMVM encoder. The masked video clip is then fed into a Video Swin Transformer [32] encoder, which is a hierarchical architecture consisting of four stage swith feature maps $[F_1, F_2, F_3, F_4]$. Each stage performs spatial downsampling using patch merging layers, which concatenate the features of each group of 2×2 spatially neighboring patches. A linear layer maps the concatenated tokens' features to half of their dimension. A series of swin attention blocks are then applied to transform the features. The encoder outputs the

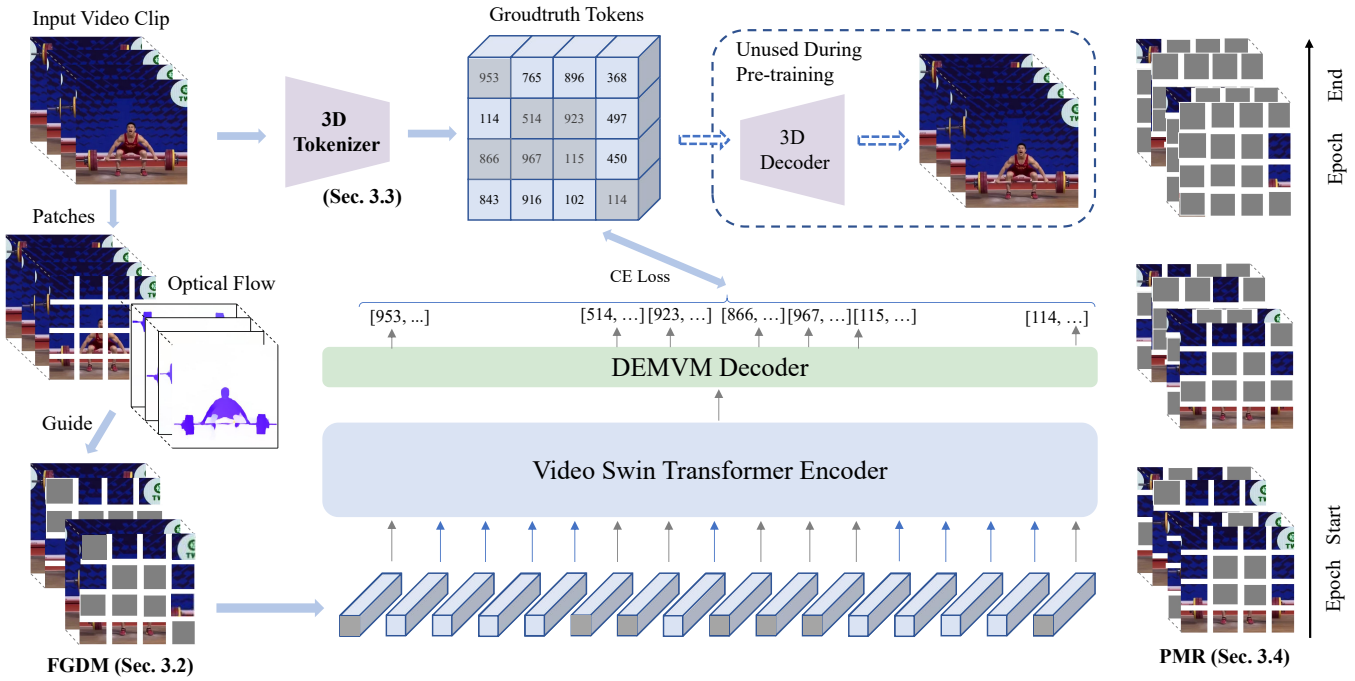

Figure 2: The overall architecture of the Data-Efficient Masked Video Modeling (DEMVM). DEMVM is pre-trained by the Masked Video Modeling (MVM) task that aims to predict the visual tokens based on the corrupted video in a self-supervised learning manner.

stage-4 feature map F_4 with a size of $\frac{T}{2} \times \frac{H}{32} \times \frac{W}{32} \times 8C$, where C is the number of channels.

3D video tokenizer. Inspired by the VideoGPT [51], we use the video tokens generated by a pretrained 3D VQ-VAE tokenizer [3] as the ground truth tokens. This tokenizer is characterized by a 2×4×4 stride and a 2048-dimensional codebook. However, since the encoding stride of the tokenizer is smaller than that of the DEMVM encoder, the input video clip X is first downsampled two times on spatial before being passed through the tokenizer. This process produces a token map output y that has a size of $\frac{T}{2} \times \frac{H}{8} \times \frac{W}{8}$.

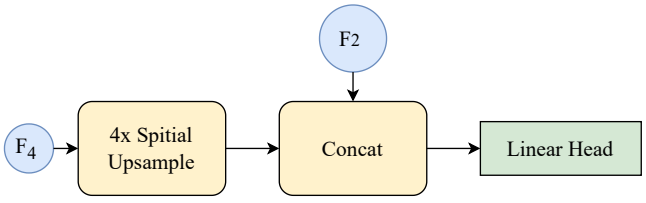

Figure 3: Structure of the DEMVM decoder. The stage-4 feature F_4 output by the encoder can be upsampled and fused to the stage-2 feature F_2 in the encoder to match the dimension of token map y.

DEMVM decoder. During upstream pre-training, it is necessary for the encoder to align the feature dimension with the ground truth token dimension and predict the token. To improve performance during downstream fine-tuning, pre-trained encoders are more critical. Therefore, we design a lightweight decoder that is decoupled from the encoder, thereby compelling the encoder to acquire a more profound comprehension of latent feature representations. As shown in Fig 3, the decoder first spatially upsamples the stage-4 feature map F_4 by 4× using bilinear interpolation. The upsampled stage-4 feature map $\hat{F}4$ is then concatenated with the stage-2 feature map F_2 to get feature F. A softmax-based classifier is used to predict the tokens for the masked patches based on the feature vector f_{mask} from F. The above operation is denoted mathematically as follows:

$$\hat{F}_4 = SpatialBilinearUpsample(F_4, 4), \tag{1}$$

$$F = Concat(\hat{F}_4, F_2), \tag{2}$$

$$\hat{y}_{mask} = softmax(W f_{mask} + b), \tag{3}$$

where $\hat{y}_{mask}$ is the predicted visual tokens of masked patches, W and b are the weight and bias of a linear layer, and f_{mask} is the feature vector for the masked patches from F.

Loss function of pre-training. Denote the y_{mask} as the true tokens labels of the masked patches, the cross-entropy loss function can be defined as:

$$L_{CE} = -\sum_{i=1}^{N_{mask}} y_{mask,i} \log(\hat{y}_i) + (1 - y_{mask,i}) \log(1 - \hat{y}_i), \tag{4}$$

where, N_{mask} is the number of masked patches. y_{mask}, i and $\hat{y}_{mask}$ are the true token label and predicted token label of the i-th patch, respectively. The final loss function can be defined as the average

of the loss functions of all masked patches:

$$L = \frac{1}{N_{mask}} L_{CE}. \tag{5}$$

Downstream fine-tuning. Once the DEMVM model has been pre-trained, the encoder is expected to learn effective video representations. It is straightforward to leverage the pre-training-then-fine-tuning paradigm on other vision tasks with DEMVM. As an illustration, considering the action recognition task, it is sufficient to append a classification head to the encoder. In this study, we employ an I3D head [8], which is a classic head that has proven its effectiveness in extensive works [2, 15, 17].

3.2 Flow-Guided Dense Masking (FGDM)

Video action recognition relies heavily on temporal action information, which is usually overshadowed by the background content. However, current masking strategies typically transfer 2D random masking to 3D, resulting in pre-training models reconstructing more background patches. To address this issue, we propose a novel Flow-Guided Dense Masking (FGDM), which applies dense masking to dynamic motion regions based on optical flow priors, while sparse masking is applied to static background regions.

Specifically, FGDM uses the classic Kanade-Lucas-Tomasi (KLT) algorithm [23] to estimate optical flow points between two consecutive frames, X_t and X_{t+1}. This algorithm calculates pixel differences between images to determine motion direction and size. FGDM obtains the motion vector for optical flow points of each frame in the video clip (excluding the first frame). To align the masked patches to tokens, FGDM upsample the result V_t of optical flow estimation by 8×. FGDM denotes the optical flow masked region of frame X_t with size of $H_{mask} \times W_{mask} \times 1$ as $V_{p,t}$, where $1 \leq t < T$. The above operation can be formalized as:

$$V_t = KLT\left(X_t, X_{t+1}\right), \tag{6}$$

$$V_{p,t} = SpatialBilinearUpsample(V_t, 8), \tag{7}$$

Let P_{den} and P_{spr} be the masking ratio of dense masked patches and sparse masked patches, respectively. The number of dense and sparse masked patches are defined as:

$$N_{den} = P_{den} P_{spa} H_{mask} W_{mask}, \tag{8}$$

$$N_{spr} = P_{spr} P_{spa} H_{mask} W_{mask}, \tag{9}$$

where P_{spa} is the masking ratio of the total masked patches in frame spatial, and $P_{den} + P_{spr} = 1$. The masking ratio $P_{act,t}$ and $P_{bg,t}$ of the motion and background regions in frame X_t, respectively, can be expressed as:

$$P_{act,t} = \frac{N_{den}}{\sum_{i=1}^{H_{mask}} \sum_{j=1}^{W_{mask}} V_{p,t}}, \tag{10}$$

$$P_{bg,t} = \frac{N_{spr}}{\sum_{i=1}^{H_{mask}} \sum_{j=1}^{W_{mask}} (1 - V_{p,t})}. \tag{11}$$

If the limited number of patches causes an exception (the masking ratio is greater than 1), the frame uses a random masking strategy with $P_{act,t} = P_{bg,t} = P_{spa}$. Finally, the masked patches consist of dense masked patches and sparse masked patches together as follows:

$$M_{den,t} = (rand(H_{mask}, W_{mask}) \leq P_{act,t}) \odot V_{p,t}, \tag{12}$$

$$M_{spr,t} = (rand(H_{mask}, W_{mask}) \leq P_{bg,t}) \odot (1 - V_{p,t}), \tag{13}$$

$$M_{mask,t} = M_{den,t} + M_{spr,t}, \tag{14}$$

where $rand(H, W)$ is a $H \times W$ random matrix, and each element is randomly sampled from the uniform distribution $[0, 1]$. $\odot$ represents element-wise multiplication. Whether the $M_{mask,t}$ is applied to the frame-t, it is randomly selected with P_{tmp} as the ratio on the time frame sequence. An example masking result of the FGDM strategy performed on the "Ice Dance" video action clip is shown in Figure 4.

Figure 4: Masked results of the FGDM implementation in the "Ice Dancing" video action clip. The first row shows the computed optical flow points. The second row depicts the scenario where $P_{den} = P_{spr}$, causing the FGDM to degenerate to the normal random mask. The third row shows FGDM under the influence of $P_{den} = 0.7$ and $P_{spr} = 0.3$, making the motion region densely masked. In this example, both spatial masking ratio P_{spa} and temporal masking ratio P_{tmp} are 0.5.

3.3 3D Video Tokenizer

Most token prediction methods in MVM still rely on a 2D image tokenizer [3, 38], which independently encodes tokens on a video frame-by-frame and then combines them afterward. However, this approach fails to consider the temporal information when encoding.

To address this issue, we propose using a 3D video tokenizer for the first time in the MVM method. This 3D tokenizer incorporates spatio-temporal information into token encoding. Specifically, we adopt the 3D tokenizer from VideoGPT [51], which follows the structure of the 2D VQ-VAE by replacing the 2D convolutional layers with 3D convolutional layers. Additionally, it introduces attention residual blocks [26] after a sequence of convolutional layer groups to improve spatio-temporal feature extraction and compression. We use the official release configuration of the 3D tokenizer with an encoding step size of $2 \times 4 \times 4$ and a codebook dimension of 2048.

3.4 Training Strategies

Progressive Masking Ratio (PMR). The MVM enables the model to learn video representations when predicting tokens of masked

patches. A high masking ratio has been found to enhance the performance of the MVM-based methods in downstream tasks [16, 22, 43]. This is because it provides more training data for the task of restoring the masked patches, while also increasing the difficulty of learning, which helps the model learn visual patterns and long-term dependencies in videos.

However, a high masking ratio can also challenge the model, especially in the early training stage and limited-data situations. To address this problem, we propose a Progressive Masking Ratio (PMR) strategy. PMR sets lower masking ratios for spatial and temporal masking ratios (P_{spa}, P_{tmp}) in FGDM during the initial stages of training to help the model quickly adapt to the MVM task. In the later stages, the higher masking ratios are used to enhance the representation learning ability of the model. Throughout the training process, the masking ratio is gradually increased by the iterations. Specifically, denote ep as the current epoch and Ep as the total number of epochs. The spatial masking ratio P_{spa} and temporal masking ratio P_{tmp} can be calculated as:

$$P_{spa} = P_{spa,s} + \frac{\left(P_{spa,e} - P_{spa,s}\right)\sqrt{ep}}{\sqrt{Ep}}, \tag{15}$$

$$P_{tmp} = P_{tmp,s} + \frac{\left(P_{tmp,e} - P_{tmp,s}\right)\sqrt{ep}}{\sqrt{Ep}}, \tag{16}$$

where $P_{spa,s}$, $P_{spa,e}$, $P_{tmp,s}$, and $P_{tmp,e}$ represent the starting and ending masking ratios in the spatial and temporal dimensions.

2D initialization. To enhance the efficiency of DEMVM, we upgrade its video encoder from the standard image encoder. Additionally, we further leverage the corresponding 2D encoder Swin Transformer [31] to inflate its pre-trained weights on ImageNet [12] to Video Swin Transformer [32] for initialization. This enables DEMVM to efficiently complete the MVM task on the limited dataset. However, as the initialized weights of the video encoder already have image-level representation learning capabilities, there exists a discrepancy with the masked patches reconstruction goal task of MVM. To address this issue, we adopt a stratified learning rate [25] for the encoder during training, which smoothly transits the goal and alleviates the possible model collapse. The stratified learning rate is set to 5e-3 times the initial learning rate.

4 EXPERIMENT

4.1 Experiment Configuration

Datasets and evaluation metric. We evaluate our method on three video action recognition datasets: UCF101 [41], HMDB51 [28], and Mimetics [50]. The UCF101 contains around 9.5k training videos and 3.5k validation videos from Youtube, and the videos are manually labeled into 101 action categories. HMDB51 is collected from various sources such as movies, YouTube, and Google videos. The dataset contains 6849 clips divided into 51 action categories, with around 3.5k training videos and 1.5k validation videos. Compared with large-scale video datasets, these two limited datasets are more suitable for verifying the effectiveness of our method, as training ViT with MVM is more challenging on limited datasets. Moreover, another Mimetics dataset contains 713 video clips from YouTube of mimed actions of 50 classes. Interestingly, this dataset does not have background biases related to actions. This makes it ideal for demonstrating the improvement of our method in temporal information utilization. Considering the tiny scale of Mimetics, we randomly select two samples from each class to build the training set for few-shot learning scenarios.

Table 1: Comparisons with the State-of-the-art methods on UCF101 dataset.

Method	Backbone	Pre-train	frames	Top-1
VCP [33]	R(2+1)D	UCF101	16	66.3
Playback [52]	R3D-18	UCF101	16	69.0
CoCLR [21]	S3D	UCF101	32	81.4
TCLR [11]	R3D-18	UCF101	16	82.4
MSCL [36]	R3D-18	UCF101	16	86.7
ViCC [42]	R(2+1)D	UCF101	16	82.8
ViCC [42]	S3D	UCF101	32	84.3
MoCo v3 [9]	ViT-B	UCF101	16	81.7
ESViT [29]	Swin-T	UCF101	16	81.1
VideoMAE [43]	ViT-B	UCF101	16	91.3
DEMVM	Swin-T	UCF101	16	88.9
DEMVM	Swin-B	UCF101	16	**92.2**

Implementation details. We pre-training the model for 10 epochs with the batch size of 24 on UCF101 and HMDB51. We follow previous trails [9, 43] setting the default resolution as 224×224 and the video length T is 16. We use Video Swin-Base and Video Swin-Tiny as the encoder in experiments throughout the paper. Dense masking ratio P_{den} of motion regions and sparse masking ratio P_{spr} of background regions in FGDM are set to 0.7 and 0.3, respectively. Starting and ending masking ratios of spatial and temporal dimensions in PMR $P_{spa,s}$, $P_{spa,e}$, $P_{tmp,s}$, and $P_{tmp,e}$ are set to 0.3, 0.8, 0.3, 0.9, respectively, and the total number of epochs Ep is set to 100. We use Adam optimizer with an initial learning rate of 5e-5 and a weight decay of 1e-5. For downstream tasks, we finetune the pretrained model for 30 epochs on UCF101, HMDB51, and Mimetics. Other configurations are consistent with pre-training. All of the experiments are implemented only on 4 Nvidia GeForce RTX 3090 GPUs with 24 GB memory, and the pretraining of 10 epochs on UCF101 takes only about 2 hours, which is extremely efficient.

4.2 Main Results and Analysis

Comparisons with the state-of-the-art methods. We compare DEMVM with the current state-of-the-art methods on UCF101 and HMDB51, and the training settings are self-supervised standards of pre-training-then-fine-tuning. Table 1 lists the results obtained on the UCF101 dataset, where DEMVM achieves a new state-of-the-art accuracy of 92.2% under ucf101 pre-training. Our method significantly outperform other comparative learning methods, whether they are CNN based or Vi-T based, with an accuracy improvement ranging from 5.5% to 25.9%. Regarding the MVM method, our accuracy is slightly better than that of VideoMAE [43]. Furthermore, the training efficiency of our DEMVM is superior to VideoMAE, as shown in Table 3. Specifically, our method only

Table 2: Comparisons with the State-of-the-art methods on HMDB51 dataset.

Method	Backbone	Pre-train	frames	Top-1
VCP [33]	R(2+1)D	UCF101	16	32.2
Playback [52]	R3D-18	UCF101	16	33.7
CoCLR [21]	S3D	UCF101	32	52.1
TCLR [11]	R3D-18	UCF101	16	52.9
MSCL [36]	R3D-18	UCF101	16	58.9
ViCC [42]	R(2+1)D	UCF101	16	52.4
ViCC [42]	S3D	UCF101	32	47.9
MoCo v3 [9]	ViT-B	HMDB51	16	39.2
VideoMAE [43]	ViT-B	HMDB51	16	62.6
DEMVM	Swin-B	UCF101	16	**62.3**
DEMVM	Swin-B	HMDB51	16	**63.9**

requires 10 training epoch with 24 batch size, whereas VideoMAE necessitates 3200 epochs with 192 batch size to achieve comparable performance. Table 3 also lists other MVM methods that require more epochs (950 to 1600) with large batch sizes (256 to 2048) on large-scale data (169k to 240k). These results demonstrate that our DEMVM achieves high accuracy with significantly fewer training resources, making it highly data-efficient and algorithm effective for self-supervised action recognition.

For HMDB51, our DEMVM also achieves a new state-of-the-art accuracy of 63.9%, as listed in Table 2. Specifically, our method outperform MoCo v3 and VideoMAE by 24.7% and 1.3%, respectively. Despite the small size of the HMDB51 dataset (3.5k), our DEMVM only requires 10 epochs with 24 batch size to achieve this result, whereas VideoMAE requires 4800 epochs with 192 batch size to achieve comparable performance (as listed in Table 3). Moreover, since many self-supervised methods prefer to pre-train on relatively large UCF101 (7.5k) and then fine-tune on HMDB51, we also compare the accuracy of our DEMVM under this setting. The results in Table 2 demonstrate that our DEMVM achieves improvements up to 3.4% to 30.1% accuracy improvement compared to other methods.

Few-shot learning on Mimetics. We aim to evaluate the efficiency of DEMVM in utilizing temporal information on the Mimetics dataset, which is a tiny dataset without background bias. To achieve this, we conduct a few-shot learning experiment in which we randomly select two samples from each category of Mimetics to form a few-shot training set. Subsequently, we employ linear probing [54] evaluation on this training set after pre-training DEMVM on UCF101. To better analyze the efficiency of DEMVM in utilizing temporal information, we use the random mask and the FGDM mask in the pre-training stage, respectively. The results listed in Table 4 show that under the effect of FGDM, the accuracy of DEMVM increases by 5.8%. These results demonstrate the effectiveness of our DEMVM in data-efficient and temporal information utilization.

4.3 Ablation Study

Feature transferability. To evaluate the feature transferability of DEMVM, we cross-conduct pre-training-then-fine-tuning experiments on UCF101 and HMDB51 datasets. The results listed in Table 5 demonstrate that DEMVM performs better when pre-trained and fine-tuned on the same dataset. Interestingly, we also observe that the results of DEMVM under cross-domain training are comparable to that under intra-domain. In particular, we are able to transfer the learned features from HMDB51 to UCF101 and achieve an accuracy of 90.9%, which is even better than most methods that directly pre-train on the UCF101 dataset, as shown in Table 1. Overall, these results demonstrate that DEMVM has satisfactory feature transferability and can effectively leverage the learned knowledge across domains.

Effect of masking ratio in FGDM. The details of the FGDM masking strategy are detailed in Section 3.2, where P_{den} and P_{spr} are used to adjust the masking ratio of the motion region and the background region. In this ablation, we conduct experiments with different masking ratios of P_{den} and P_{spr} on UCF101, and the results are listed in Table 6. The baseline accuracy of 87.6% is achieved when $P_{den} = P_{spr}$, as FGDM degenerates to the normal random mask. When $P_{den} < P_{spr}$, the FGDM is equivalent to applying the dense mask to the background region, whose results are generally worse than the baseline. However, when $P_{den} > P_{spr}$, the FGDM begins to play a positive role. Specifically, when $P_{den} = 0.7$ and $P_{spr} = 0.3$, it yields a 1.3% accuracy gain. These results demonstrate the importance of leveraging temporal action information. It is worth noting that the accuracy gain brought by FGDM decreases when the masking ratio of the background region is extremely low ($P_{den} = 0.9, P_{spr} = 0.1$), which demonstrates that action-related background information is also helpful.

Effect of each component. In this section, we investigate the effectiveness of a 3D tokenizer, lightweight decoder (LW-D), stratified learning rate (S-LR), Progressive Masking Ratio(PMR), and Flow-Guided Dense Masking (FGDM) in our DEMVM.

Taking Swin-T as the backbone, the ablation studies results on the UCF101 dataset are listed in Table 7. To establish a baseline, we compare our method to an approach that directly transforms MIM to video using a 2D tokenizer, random tube mask, and complex decoder. The complex decoder refers to applying 2× upsampling twice and fusing the features in each upsampling stage. Subsequently, we introduce different components to the baseline in various rows of the table.

Comparing the first two rows, it can be seen that the 3D tokenizer plays a vital role in the DEMVM. The significant performance improvement indicates that the 3D tokenizer effectively captures spatio-temporal information. The inclusion of LW-D results in an accuracy increase to 86.5%, confirming our initial design objective of using a lightweight and uncomplicated decoder to motivate the encoder to acquire a more profound understanding of latent feature representations. The addition of S-LR further consolidates the performance to 86.9%. Comparing the results of the 4th, the 5th, and the 6th row, PMR and FGDM still have a great positive impact based on higher performance, especially FGDM, which significantly improves performance with an accuracy of 88.2%. This again demonstrates that FGDM is capable of effectively leveraging temporal action information. A noteworthy outcome is observed in the last row, where the accuracy is further improved following the combination of PMR and FGDM. This finding highlights the efficacy of PMR in enabling DEMVM to effectively accomplish the

Table 3: Comparisons of computational resources and model efficiency with existing MIM/MVM methods. Due to the distinct computation methods of GFlops for the image correlation technique and the video, solely the GFlops for the video are listed.

Method	Backbone	Dataset	Data Size	frames	Epoch	Batch Size	GFlops	Params	Paradigm
BEiT [4]	Transformer-B	ImageNet-1k	128M	1	800	2000	-	307	MIM
MAE [22]	ViT-L	ImageNet-1k	128M	1	800	4096	-	304	MIM
MaskFeat [49]	ViT-L	ImageNet-21k	1400M	1	1600	2048	-	304	MIM
BEVT-I [48]	Swin-B	ImageNet-1k	128M	1	800	2048	-	88	MIM
MaskFeat [49]	MViT-L	Kinetics-400	240k	16	800	512	377×10	218	MVM
BEVT-V [48]	Swin-B	Kinetics-400	240k	16	150	256	282×12	88	MVM
OmniMAE [18]	ViT-B	SSv2	169k	16	1600	2048	180×15	87	MVM
MAE-ST [16]	ViT-B	Kinetics-400	240k	16	1600	512	180×21	87	MVM
VideoMAE [43]	ViT-B	UCF101	7.5k	16	3200	192	180×15	87	MVM
VideoMAE [43]	ViT-B	HMDB51	3.5k	16	4800	192	180×15	87	MVM
DEMVM	Swin-T	UCF101	**7.5k**	16	**10**	**24**	$\mathbf{88 \times 12}$	**28**	MVM
DEMVM	Swin-B	UCF101	**7.5k**	16	**10**	**24**	282×12	88	MVM
DEMVM	Swin-B	HMDB51	**3.5k**	16	**10**	**24**	282×12	88	MVM

Table 4: Few-shot Learing reuslts on Mimetics.

Method	Backbone	Pre-train	Mask	Top-1
DEMVM	Swin-T	UCF101	Random	8.3
DEMVM	Swin-T	UCF101	**FGDM**	**14.1**

Table 5: Comparisons with the feature transferability.

Method	Backbone	Pre-train	UCF101	HMDB51
DEMVM	Swin-B	UCF101	**92.2**	62.3
DEMVM	Swin-B	HMDB51	90.9	**63.9**

Table 6: Ablation study on the masking ratio of motion and background regions.

Backbone	motion (P_{den})	Background (P_{spr})	Top-1
Swin-T	0.9	0.1	88.2
Swin-T	0.8	0.2	88.4
Swin-T	**0.7**	**0.3**	**88.9**
Swin-T	0.6	0.4	88.7
Swin-T	0.5	0.5	87.6
Swin-T	0.4	0.6	87.8
Swin-T	0.3	0.7	87.6
Swin-T	0.2	0.8	87.4
Swin-T	0.1	0.9	87.4

challenging task of dense motion mask pre-training. Our ablation study demonstrates that each proposed component plays a crucial role in improving the performance of our DEMVM, and the combination of all components achieves the best results of 88.9%.

Table 7: Ablation study of proposed components for DEMVM.

Baseline	3D Tokenizer	LW-D	S-LR	PMR	FGDM	Top-1
✓						82.1
	✓					85.8
	✓	✓				86.5
	✓	✓	✓			86.9
	✓	✓	✓	✓		87.6
	✓	✓	✓		✓	88.2
	✓	✓	✓	✓	✓	**88.9**

5 CONCLUSION

In this paper, we propose a novel method named Data-Efficient Masked Video Modeling (DEMVM) for self-supervised action recognition. Our DEMVM addresses a key limitation of existing Masked Video Modeling (MVM) methods by designing a Flow-Guided Dense Masking (FGDM) strategy that effectively captures the temporal action information in videos. Additionally, we introduce a 3D video tokenizer that better represents the spatio-temporal features of videos. To adapt to the MVM paradigm during different training stages, we present the Progressive Masking Ratio (PMR) and 2D initialization techniques. These enable our approach to efficiently complete pre-training with both efficient data and computational cost, significantly improving upon existing MVM methods that require large-scale datasets and significant computing resources. Extensive experiments on three public action recognition datasets demonstrate the state-of-the-art performance of our method.

ACKNOWLEDGMENTS

This work is partially supported by National Key R&D Program of China (No. 2021YFC3310100), and National Natural Science Foundation of China (Nos. 62176251, 61976219).

REFERENCES

[1] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. 2020. Self-supervised multimodal versatile networks. *Advances in Neural Information Processing Systems* 33 (2020), 25–37.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6836–6846.

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1728–1738.

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

[5] Steven S. Beauchemin and John L. Barron. 1995. The computation of optical flow. *ACM Computing Surveys (CSUR)* 27, 3 (1995), 433–466.

[6] Nadine Behrmann, Mohsen Fayyaz, Juergen Gall, and Mehdi Noroozi. 2021. Long short view feature decomposition via contrastive video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9244–9253.

[7] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[8] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[9] Xinlei Chen, Saining Xie, and Kaiming He. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9640–9649.

[10] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. Ieee, 886–893.

[11] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. 2022. TCLR: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding* 219 (2022), 103406.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 248–255.

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

[14] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. 2022. PYSKL: Towards good practices for skeleton action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*. 7351–7354.

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

[16] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. 2022. Masked autoencoders as spatiotemporal learners. *Advances in Neural Information Processing Systems* 35 (2022), 35946–35958.

[17] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. 2019. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 244–253.

[18] Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2022. OmniMAE: Single model masked pretraining on images and videos. *arXiv preprint arXiv:2206.08356* (2022).

[19] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*. 5842–5850.

[20] Jie Gui, Tuo Chen, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. 2023. A Survey of Self-Supervised Learning from Multiple Perspectives: Algorithms, Theory, Applications and Future Trends. *arXiv preprint arXiv:2301.05712* (2023).

[21] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Self-supervised co-training for video representation learning. *Advances in Neural Information Processing Systems* 33 (2020), 5679–5690.

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

[23] Duan Hong. 2004. Tracking facial feature points using kanade-lucas-tomasi approach. *Journal of Computer Aided Design and Computer Graphics* 16, 3 (2004), 279–283.

[24] Lianghua Huang, Yu Liu, Bin Wang, Pan Pan, Yinghui Xu, and Rong Jin. 2021. Self-supervised video representation learning by context and motion decoupling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13886–13895.

[25] Xiaolong Huang and QianKun Li. 2022. Runner-Up Solution to Google Universal Image Embedding Competition 2022. *arXiv preprint arXiv:2210.08735* (2022).

[26] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. 2019. CCNet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 603–612.

[27] Yuqi Huo, Mingyu Ding, Haoyu Lu, Zhiwu Lu, Tao Xiang, Ji-Rong Wen, Ziyuan Huang, Jianwen Jiang, Shiwei Zhang, Mingqian Tang, et al. 2021. Self-supervised video representation learning with constrained spatiotemporal jigsaw. (2021).

[28] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. 2011. HMDB: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*. IEEE, 2556–2563.

[29] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. 2021. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785* (2021).

[30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10012–10022.

[32] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2022. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3202–3211.

[33] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. 2020. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11701–11708.

[34] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2630–2640.

[35] Jianyuan Ni, Anne HH Ngu, and Yan Yan. 2022. Progressive Cross-Modal Knowledge Distillation for Human Action Recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*. 5903–5912.

[36] Jingcheng Ni, Nan Zhou, Jie Qin, Qian Wu, Junqi Liu, Boxun Li, and Di Huang. 2022. Motion Sensitive Contrastive Learning for Self-supervised Video Representation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*. Springer, 457–474.

[37] Rui Qian, Yuxi Li, Huabin Liu, John See, Shuangrui Ding, Xian Liu, Dian Li, and Weiyao Lin. 2021. Enhancing self-supervised video representation learning via multi-level feature optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7990–8001.

[38] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*. PMLR, 8821–8831.

[39] Hideo Saito, Thomas B Moeslund, and Rainer Lienhart. 2022. MMSports' 22: 5th International ACM Workshop on Multimedia Content Analysis in Sports. In *Proceedings of the 30th ACM International Conference on Multimedia*. 7386–7388.

[40] David Schneider, Saquib Sarfraz, Alina Roitberg, and Rainer Stiefelhagen. 2022. Pose-based contrastive learning for domain agnostic activity representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3433–3443.

[41] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[42] Martine Toering, Ioannis Gatopoulos, Maarten Stol, and Vincent Tao Hu. 2022. Self-supervised video representation learning with cross-stream prototypical contrasting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 108–118.

[43] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602* (2022).

[44] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 4489–4497.

[45] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. 2019. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4006–4015.

[46] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. 2020. Self-supervised video representation learning by pace prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer, 504–521.

[47] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. 2020. Self-supervised video representation learning by pace prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16.*

Springer, 504–521.
[48] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. 2022. BEVT: BERT pretraining of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14733–14743.
[49] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. 2022. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14668–14678.
[50] Philippe Weinzaepfel and Grégory Rogez. 2021. Mimetics: Towards understanding human actions out of context. *International Journal of Computer Vision* 129, 5 (2021), 1675–1690.
[51] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. VideoGPT: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157* (2021).
[52] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. 2020. Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6548–6557.
[53] Dingwen Zhang, Chaowei Fang, Wu Liu, Xinchen Liu, Jingkuan Song, Hongyuan Zhu, Wenbing Huang, and John Smith. 2022. HCMA'22: 3rd International Workshop on Human-Centric Multimedia Analysis. In *Proceedings of the 30th ACM International Conference on Multimedia*. 7407–7409.
[54] Richard Zhang, Phillip Isola, and Alexei A Efros. 2017. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1058–1067.
[55] Tao Zhuo, Zhiyong Cheng, Peng Zhang, Yongkang Wong, and Mohan Kankanhalli. 2019. Explainable video action reasoning via prior knowledge and state transitions. In *Proceedings of the 27th ACM International Conference on Multimedia*. 521–529.

A APPENDIX

A.1 Visualization of Reconstructing Masked Video.

Figure 5 shows the visualization of DEMVM in the pre-training stage for reconstructing masked video. The visualization configurations of DEMVM include masking ratios of 30% and 80%, as well as a motion-to-background region ratio of 0.7:0.3. It can be observed that under the low masking ratio of 30%, the reconstructed videos exhibit favorable quality, with both motion and background accurately restored. Under the high masking ratio of 80%, the videos lack visible information, yet DEMVM is still capable of reconstructing the overall motion, while the relatively sparse and simple background region is comprehensively reconstructed. These results demonstrate that DEMVM can learn high-quality video representations in the pre-training stage and enhance its focus on the motion region.

A.2 Visualization of Mimetics Datasets

Figure 6 shows several video clips from the Mimetics dataset. It can be observed that the video data in Mimetics exhibit no background bias. For instance, actions such as indoor surfing in a waterless environment, and shooting basketball without a basketball in non-basketball court locations.

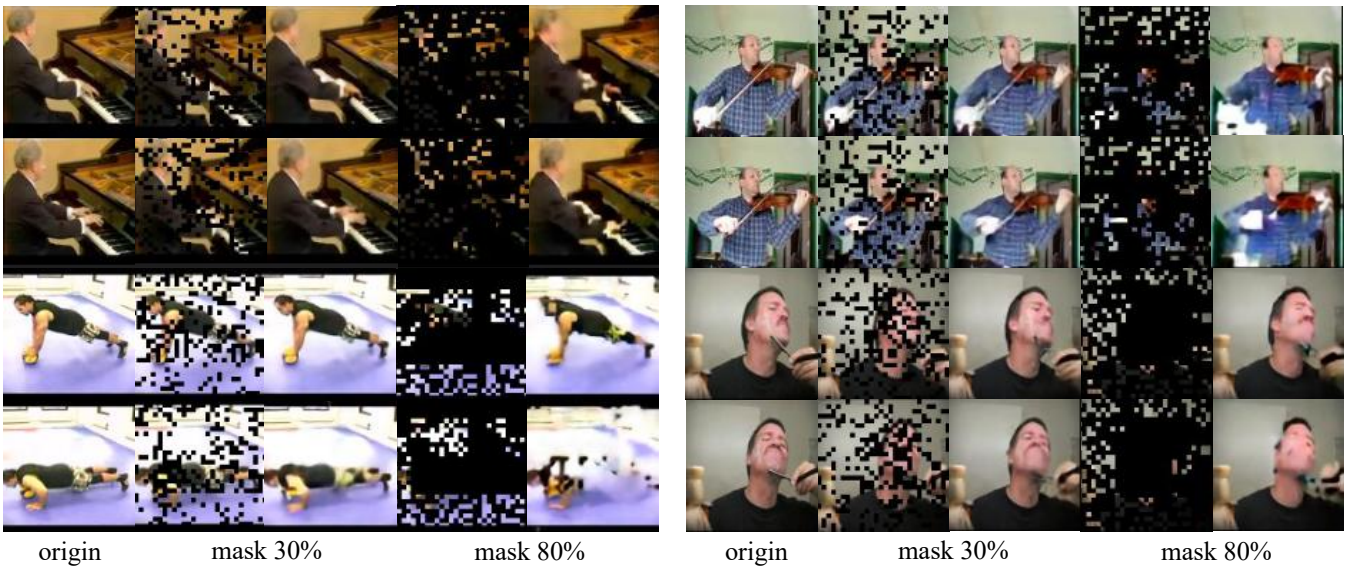

Figure 5: Reconstruction results using DEMVM with 30%, 80% masking ratio, and 0.7:0.3 ratio of motion and background regions.

Figure 6: Examples of the Mimetics dataset. Indoor demo of surfing (left) and no basketball shooting (right).