MAR: Masked Autoencoders for Efficient Action Recognition

Zhiwu Qing®, Shiwei Zhang®, Ziyuan Huang®, Xiang Wang®, Yuchuan Wang®, Yiliang Lv,
Changxin Gao®, Member, IEEE, and Nong Sang®, Member, IEEE

Abstract—Standard approaches for video action recognition usually operate on full input videos, which is inefficient due to the widespread spatio-temporal redundancy in videos. The recent progress in masked video modelling, specifically VideoMAE, has shown the ability of vanilla Vision Transformers (ViT) to complement spatio-temporal contexts using limited visual content. Inspired by this, we propose Masked Action Recognition (MAR), which reduces redundant computation by discarding a proportion of patches and operating only on a portion of the videos. MAR includes two essential components: cell running masking and bridging classifier. Specifically, to enable the ViT to perceive the details beyond the visible patches, cell running masking is used to preserve the spatio-temporal correlations in videos. This ensures that the patches at the same spatial location can be observed in turn for easy reconstructions. Additionally, we notice that, although the partially observed features can reconstruct semantically explicit invisible patches, they fail to achieve accurate classification. To address this issue, we propose a bridging classifier that can help fill the semantic gap between the ViT encoded features used for reconstruction and the specialized features used for classification. Our proposed MAR can reduce the computational cost of ViT by 53%. Extensive experiments have demonstrated that MAR consistently outperforms existing ViT models by a notable margin. Notably, we found that a ViT-Large model fine-tuned by MAR achieves comparable performance to a ViT-Huge model fine-tuned by standard training methods on both Kinetics-400 and Something-Something v2 datasets. Moreover, the computation overhead of our ViT-Large model is only 14.5% of that of the ViT-Huge model.

Index Terms—Efficient action recognition, masked autoencoders, spatio-temporal redundancy, vision transformer.

I. INTRODUCTION

In recent years, deep neural networks [1], [2], [3], [4], [5], [6] have achieved impressive performance for action recognition on several large-scale video datasets [7], [8]. These methods typically rely on full video frames to understand the visual content. While this approach yields decent performances, computation over full videos is highly redundant due to the excessive and widely present spatio-temporal redundancy of visual information [9], [10], [11], [12], [13] in videos. Consequently, some previous works have proposed reducing spatio-temporal redundancy by training an additional model to focus on relevant frames [12], [13], [14], [15], [16], [17] or spatio-temporal regions [10], [11], which can significantly reduce the computation cost. However, these methods often require complicated operations, such as reinforcement learning and multi-stage training.

In self-supervised video representation learning, Masked Autoencoders (MAE) [9], [18] discard a high proportion of visual patches to yield a non-trivial and meaningful self-supervised reconstruction task. The simple masking strategy with vanilla Vision Transformers (ViT) can achieve realistic reconstruction results [9], [18], implying that the masked visual information in videos can be complemented from limited visible contents, and the ViT shows this capability. This is possible because the spatio-temporal redundancy in videos empowers the model to derive the visual semantics of the invisible parts from the visible contexts. After pre-training, the models are fine-tuned on the downstream action recognition task following a standard scheme, which feeds all the details of videos into ViT.

In this work, we argue that given the strong ability of ViT to reconstruct visual semantics with limited visual content and the high spatio-temporal redundancy in videos, the standard action recognition scheme that operates on full video frames is highly inefficient. To this end, we propose a simple and computationally efficient end-to-end scheme for the action recognition task, termed as Masked Action Recognition (MAR). The core idea of MAR is to discard a subset of video patches to reduce the encoded tokens of ViT, thus avoiding redundant computation to some extent. We investigate this from two perspectives: (i) designing an appropriate input masking map with strong spatio-temporal correlations for ViTs; (ii) increasing the abstraction level for the features output by ViTs.

For the first perspective, we aim to determine a reasonable mask form for action recognition. The existing VideoMAE
methods [9], [18] adopt a large proportion (i.e., 90%) of tube masking or random masking, which are mainly designed to avoid “information leakage” caused by spatio-temporal redundancy and correlation in videos to improve the difficulty of the self-supervised reconstruction task. On the contrary, detailed contexts are crucial for action recognition. Therefore, training an accurate action recognizer with masks requires the model to easily complement the disappeared details. To this end, we present cell running masking, which encourages “spatio-temporal information leakage” to enhance the detail perception of encoders for the lost patches. Specifically, we first design a running masking strategy to move the masks frame by frame, allowing the patches in the same spatial location to be observed in turn. To generate more flexible masking maps for training, we divide the running masking into multiple small running cells and place a specified proportion of masks in these cells. Executing the running strategy in the small cells can present multiple states to provide various spatio-temporally interleaved masks. Overall, the cell running masking preserves spatio-temporal correlations [9], which can be easily exploited by the encoder to perceive missing details for accurate action recognition. Fig. 7 shows that the semantically explicit reconstructed videos can be observed with this masking strategy.

Despite this, we also observe that the models with realistically reconstructed videos still fail to achieve on-par accuracy compared to using the full patches. Additionally, MAE in the image domain [20] reveals a semantic gap between the low-level features required for reconstruction and the abstract features required for recognition. The literature [21] has also found that MAE pre-trained models extract features at a lower level of abstraction. To solve this issue, we further propose bridging classifier with a similar structure to the reconstruction decoder in MAE. However, unlike the reconstruction decoder, the bridging classifier is designed to bridge the semantic gaps further and make the encoded features more specialized for the classification task. Conversely, the reconstruction decoder is used to decode low-level pixel-wise information.

In this way, compared to the standard action recognition scheme, MAR is shown to reduce the computation by up to 53% while achieving on-par or better performances, as shown in Fig. 1. In particular, when training a large model on Kinetics-400 [7], our ViT-Large costs only 14.5% of FLOPs and surpasses the performance of the ViT-Huge by 0.2%.

In a nutshell, our contributions can be summarized as:

- We explore the masked autoencoders for efficient action recognition, achieving better performance and a 2x wall-clock speedup in training and testing.
- We present a novel MAR framework, consisting of cell running masking and bridging classifier to better exploit the spatio-temporal correlations and increase the abstract level of encoded features, respectively.
- Extensive experiments on different datasets show that MAR requires only 47% of the computation to consistently outperform standard action recognition with the same pre-trained models. Especially under comparable computational costs, models trained by MAR significantly outperform existing state-of-the-art methods.

II. RELATED WORKS

A. Action Recognition

Action recognition is a fundamental task for video analysis and understanding. Recent video networks can be divided into two types, i.e., convolution-based and Transformer-based. The dominant convolution-based methods build upon 3D Convolutional Neural Networks (CNNs) [22], [23], [24]. Inspired by the two-stream CNNs, separately modelling spatial appearance and temporal relations by 3D convolutions [1], [25], [26] have also shown decent accuracy. However, they suffer from a large computation burden. To reduce complexity, some approaches attempt to disentangle 3D convolution into spatial and temporal convolution [2], [27], [28], [29]. Other works employ 2D convolutions for spatial modelling [30], [31] and introduce additional temporal operations, such as calibrating the convolution weights by temporal context [32], designing spatio-temporal attention [33], [34], generating adaptive temporal kernels [35], mining temporal cross layer correlation [36], fusing appearance and motion [37], capturing temporal difference for motion modelling [3], [38], [39], integrating multi-level predictions [40], and shifting part of the channels along the temporal dimension [41], etc. Limited by the small receptive field of the convolution operator, convolution-based methods struggle to model long-range spatio-temporal dependencies. In recent years, the great success of Transformers in the image domain [19], [42], [43] has led to the exploration of Transformer-based video networks. VTN [44] adopts ViT [19] to extract spatial features, followed by a Longformer [45] to capture temporal relationships. Both TimeSformer [46] and ViViT [6] factorize different spatial and temporal attentions for Transformer encoders, suggesting that factorized spatial and temporal attention can achieve better performance. MViT [5], [47] studies hierarchical Transformers with several channel-resolution scale stages, and proposes pooling attention to reduce computation. In comparison, Video Swin Transformer [4] introduces an inductive bias of locality for videos and achieves a better speed-accuracy trade-off. And Uniformer [48] captures local spatio-temporal context and
global token dependency by convolution in shallow layers and Transformer in deep layers, respectively. All these studies are based on variants of Transformer structure. In this work, we adopt ViTs [19] pre-trained by VideoMAE [9] as our encoder. We investigate the input and output of Transformer encoders and empirically demonstrate that the proposed MAR has advantages in both efficiency and performance.

B. Spatio-Temporal Redundancy

Reducing spatio-temporal redundancy for efficient video analysis has been a popular research topic. Mainstream approaches typically involve training an additional lightweight network to achieve the following: (i) adaptive frame selection [12], [13], [14], [16], [49], which involve dynamically determining the relevant frames for recognition networks; (ii) adaptive frame resolution [12], which involves learning an optimal resolution for each frame online; (iii) early stopping [50], which involves terminating the inference process before observing all frames; (iv) adaptive spatio-temporal regions [10], [11], which involve localizing the most task-relevant spatio-temporal regions; (v) adaptive network architectures [15], [16], [51], which involves adjusting the network architecture to save computation on less informative features. Another line is to manually define low redundant sampling rules, such as MGSampler [52], which selects frames containing rich motion information based on cumulative motion distribution. However, in this work, the ViTs trained by MAR adopt only a proportion of video patches for efficient action recognition, taking advantage of the powerful completion capabilities of Transformers.

C. Masked Autoencoders

Masked autoencoder, as a form of denoising autoencoding [53], is a general methodology for learning effective representations by reconstructing the uncorrupted inputs from corrupted inputs. In Natural Language Processing (NLP), the masked language modelling task proposed in BERT [54] is one of the most successful explorations of masked autoencoding. Various variants [55], [56], [57], [58] based on BERT have also further improved the performance of language Transformer pre-training. Recently, in the image domain, a series of masked autoencoding methods have been seeking a framework for vision and language unification based on Transformer architectures [59], and progress has been made. iGPT [60] first proposed training a Transformer to predict pixels from a sequence of low-resolution pixels for unsupervised representation learning. Then ViT [19] takes image patches as tokens and performs masked patch prediction to mimic the masked language modelling in BERT [54]. SimMIM [61] suggests that using a large masked patch size for pixel predictions can create a strong pre-text task. Image MAE [20] investigates an asymmetric encoder-decoder structure, where the heavy encoder only operates on a small proportion (25%) of visible patches, while a lightweight decoder is used to reconstruct pixels. Besides pixel prediction, another research line also proposes to reconstruct other targets, such as pre-trained dVAE [62], [63] of BEiT [64], and HoG [65] of MaskFeat [66]. In the video domain, two MAE-based methods [9], [18] find that an extremely high proportion of masks yields decent performance due to the large spatio-temporal redundancy in videos. Instead of predicting pixels, BEVT [67] and VIMPAC [68] also attempt to learn spatio-temporal representations by predicting features derived from a tokenizer. All the masked autoencoder based works discussed above focus on learning an effective self-supervised visual representation. We are inspired by the powerful completion ability of ViT in reconstruction tasks [9], [18], [20] and propose to adopt this idea to empower supervised action recognition models in terms of both efficiency and performance.

III. APPROACH

As shown in Fig. 2, instead of full video frames, the proposed MAR takes masked videos as input. Specifically, the MAR encoder only operates over the visible patches, allowing it to process videos with a fraction of computation and memory cost. The encoded visible tokens are then fed into two branches: the reconstruction branch for pixel-level reconstruction, and the classification branch for efficient action recognition. Note that the auxiliary reconstruction branch is only used during training and removed during inference. Besides the general framework, we also present two key designs in MAR: (i) To facilitate the perception of the MAR encoder over invisible patches, we propose a cell running masking strategy (in Section III-B) that generates a masking map ensuring strong spatio-temporal correlations; (ii) To bridge the semantic gaps between the encoded visible tokens and features more suitable for the action classification task, we introduce a bridging classifier (in Section III-C) in the classification branch that increases the abstraction level of tokens. In this section, we first briefly revisit masked video modelling and then present the ideas of cell running masking and bridging classifier.

A. Masked Video Modelling

The existing masked video modelling approaches, with encouraging performances [9], [18], are extended from image MAE. They first divide an input video into several non-overlapped spatio-temporal patches, and then only a small proportion of patches (i.e., 10%) with positional embeddings are randomly selected as input for the ViT encoders [19]. A lightweight decoder is then adopted to reconstruct the full video from the encoded latent representations of visible patches. VideoMAE [9] identifies two crucial characteristics of video data, i.e., temporal redundancy and temporal correlation. The former means that the semantics vary slowly in the temporal dimension, and the spatio-temporal information is highly redundant, indicating that retaining all spatio-temporal patches for training and inference is inefficient and unnecessary. While the latter emphasizes the strong inherent correlation between adjacent frames, which could cause “information leakage” between frames, thus reducing the reconstruction difficulty of VideoMAE. Hence, a large proportion (i.e., 90%) of mask ratio and tube masking are proposed for the video reconstruction task. The pre-trained models can still achieve satisfactory reconstruction
Fig. 2. Overview of Masked Action Recognition (MAR). A specified proportion (e.g., 50% here) of patches is first discarded by cell running masking (C.R.M.), which retains sufficient spatio-temporal correlations. Next, the remaining visible patches are fed into the encoder to extract their spatio-temporal features (i.e., visible tokens in the figure). Finally, the reconstruction decoder receives mask tokens and visible tokens to reconstruct the masked patches. In contrast, the bridging classifier in the classification branch receives only visible tokens for action classification. Note that the reconstruction branch is only performed during training, which is used to preserve the completion capability of the encoders for invisible patches.

Fig. 3. Different schemes for running masking. (a) randomly removing spatial patches in the first frame; (b) removing a large spatial block in the first frame; (c) uniformly distributing the masks in the first frame; (d) performing running masking in cells; (e) the masks run circularly frame by frame in a running cell.

videos with limited visible contents, which implies the powerful spatio-temporal association ability of ViTs. However, in the downstream action recognition task, all patches are still fed into the encoder. Since the number of tokens in a video clip is much larger than that of an image (e.g., 1568 vs. 196), and the attention matrix size is quadratic to the number of tokens, which causes a heavy computational burden. In this work, we draw inspiration from the powerful capability of complementing invisible contexts shown by MAE pre-trained ViTs and propose MAR to achieve efficient action recognition.

B. Cell Running Masking

1) Running Masking: When performing the action recognition task with masks, a key issue is that the valuable details in masked patches are removed together, which is bound to damage the accuracy. To this end, instead of avoiding “information leakage” between the adjacent frames like VideoMAE [9], we encourage “information leakage” caused by temporal correlation to lower the difficulty of the reconstruction task. This is because easier reconstruction means that richer details can be redrawn to improve recognition performance.

Specifically, we first propose running masking strategy where the masks run frame by frame, and the patches at different spatial locations are discarded in turn. It can be formulated as:

\[ M_t = s(M_{t-1}|M_{t-2}, \ldots, M_1), \]

where \( M_t \) is the masking map for frame \( t \), and the function \( s(\cdot|\cdot) \) indices that the masking map of frame \( t \) is transformed from frame \( t-1 \) circularly. Fig. 3(e) also shows the masks in a small running cell between adjacent frames. With this idea, running masking ensures that the same spatial locations in consecutive frames can quickly observe the visible patches from adjacent frames, which exploits spatio-temporal visual redundancy to mitigate information loss.

However, it is still challenging to get an efficient and effective implementation of running masking. As shown in Fig. 3(a), one potential option is to create random masks on the first frame and then use the function \( s(\cdot|\cdot) \) to transform these masks frame by frame. This straightforward idea suffers from the randomness of masks, and patches in mask-dense spatial locations may not be displayed, which still cannot effectively exploit video visual correlations. Similar flaws arise in the block-wise running masking depicted in Fig. 3(b), where dense masks can destroy chunks of spatio-temporal context. Fig. 3(c) shows a different uniform running masking. It arranges uniform grid masks on the first frame to create spatial dislocations that help the model...
deduce spatial semantics for the masked patches by using the image’s spatial redundancy. These grid masks are then shifted in subsequent frames to form temporal intersections, enabling the model to retrieve detailed information about the masked parts by leveraging temporal correlations in the video. Therefore, the uniform running masking can efficiently exploit spatio-temporal redundancy. However, it is worth noting that adopting the rigid “uniform masking” can easily lead to overfitting during training since the diversity of uniform masking maps is limited.

2) Running Cell: For this reason, we propose decomposing the running masking into multiple small repeated units, which we call running cells. These simple units can then be combined to implement complex masks, known as cell running masking, as shown in Figs. 3(d), (e) and 4(a), (b). We define the spatial size of a running cell as \( r \times q \) (e.g., \( r = q = 2 \) in Fig. 3(e)). When the spatial size is small, there are only a few patches in the running cell, making the arrangement of masks clear. For example, in Fig. 3(e), only two masks need to be placed when the mask ratio is set to 50%. We simply put the two masks in the first two patches and define this as state \( A \). Driven by the function \( s(\cdot) \), this cell has four different states, i.e., \( A \to B \to C \to D \). The cell starts with state \( A \) and performs state switching with a period of 4 in the temporal dimension to realize a spatio-temporal uniform masking map. This design ensures that each patch has an equal chance of being visible in four temporal patches, thus providing enough temporal correspondences for other masked patches with the same spatial locations.

We also observe that a running cell with a spatial size of \( 2 \times 2 \) can only provide three different mask ratios, i.e., \( 25\% \), \( 50\% \), and \( 75\% \). These ratios already meet most needs. However, using running cells with a large spatial size can produce more mask ratios. Nevertheless, the diversity of mask combinations is weakened because there are fewer cells. For instance, when the size of the running cell is equal to the spatial size of the ViT-encoded tokens, i.e., \( 14 \times 14 \), it degenerates to uniform running masking.

3) Augmentations of Cell Running Masking: Fig. 3(e) shows that the illustrated running cell has four different states in turn, i.e., \( A \to B \to C \to D \), in the illustrated running cell. However, the fixed sequence and positions may cause overfitting during training, similar to the uniform running masking in Fig. 3(c). To address this issue, we design the running cell to allow for multiple different cells in the space, with each cell free to choose the starting state. As shown in Fig. 4(a), each running cell selects the starting state randomly, and their combination is close to random masking, but the regulation of running masking is followed within each small cell. Figs. 3(d) and 4(b) spatially repeat running cells, but their starting states are selected randomly as \( A \) and \( B \), respectively, resulting in visually different masking maps. Both the spatially random and spatially repeated masking in Fig. 4 can be used as a data augmentation during training. For inference, we use spatially repeated running masking for evaluation. The experiments in Table V(b) demonstrate that the model is robust to the different starting states of running cells during inference.

4) Implementations: Algorithm 1 shows a PyTorch-like pseudo-code of the proposed cell running masking, which can be divided into two steps: cell initialization and masking generation. Specifically, in initialization, we first make the masks evenly distributed in a cell (the size is \( r \times q \)), and then shift this cell frame-by-frame to form an running cell. In generation, the running cell is first repeated in the spatial dimension to obtain the desired masking size. Then, shuffling the temporal dimension is only performed in training for augmentation.

C. Bridging Classifier

The design of cell running masking preserves the spatio-temporal correlations between visible patches and invisible patches, effectively reducing the difficulty of reconstruction and further improving the reconstruction quality. For example, when the mask ratio is set to 50%, the reconstruction branch can already achieve sufficiently satisfactory reconstructed videos, as shown in Fig. 7. However, when using a fully connected layer as a classifier in Table VI, the recognition accuracy with 50% of the masked patches still struggles to reach the performance without masking. Our bridging classifier sets out to close this performance gap. Image MAE [20] mentioned that the pixel-level reconstruction and the recognition tasks require latent representations

---

**Algorithm 1**: Pseudo-Code of Cell Running Masking in a PyTorch-Like Style.

```python
def init_running_cell(pT, r, q, mask_ratio):
    # Initialize the running cell.
    super().__init__(self, pT, r, q, mask_ratio)
    self.r, self.q = r, q
    cell = torch.ones(pT, r, q).flatten(1, 2)
    mask_num_per_cell = int(mask_ratio * r * q)
    stride = (r + q) / mask_num_per_cell
    for i in range(pT):
        for j in range(mask_num_per_cell):
            if (i + j) % (r + q) == 0:
                self.cell[i, j] = 0
                self.cell_view = (i, j, T)

def generate_running_mask(self, x):
    # Generate running mask for an input video
    mask = self.cell.repeat(1, x.size(2))/self.r, x.size(3)/self.q
    if self.training:
        # Temporally shuffled
        mask = shuffle_temporal_dim(mask)
    return mask
```

---

Fig. 4. Different combinations of running cells in space. (a) each running cell randomly selects a starting state; (b) all running cells share the same random starting state.
at different abstract levels. Specifically, representations with higher-level semantics can lead to better recognition accuracy but are not specialized for reconstruction. In this work, the satisfactory reconstructions prove that the low-level semantic information in features is sufficient. Hence, we speculate that the weak classification accuracy is actually because the linear classifier cannot fully exploit low-level semantic information.

To this end, we propose bridging classifier consisting of a series of Transformer blocks, like the reconstruction decoder, to bridge the semantic gaps between the encoded features and the classification features:

\[ p = \phi(h(F_v)), \]

where \( F_v \in \mathbb{R}^{N_v \times D} \) is the encoded visible tokens output by the encoder, and \( \phi(\cdot) \) and \( h(\cdot) \) denote the average pooling operation and proposed bridging classifier, respectively. \( N_v \) is the number of visible tokens, and \( D \) is the channel. \( p \in \mathbb{R}^C \) represents the model prediction for \( C \) classes. The reconstruction decoder infers the encoded \( F_v \) to pixels, while the bridging classifier extracts the more concise classification features in \( F_v \). Therefore, the bridging classifier is designed to be more lightweight. Additionally, since the bridging classifier only processes visible tokens, compared with the linear classifier using 50% of visible tokens, the additional FLOPs introduced by the bridging classifier are only 8%.

D. Loss Function

MAR contains both the reconstruction and classification branches, with their loss functions denoted as \( L_r \) and \( L_c \), respectively. The reconstruction loss \( L_r \) is a pixel-wise mean-squared loss, following [9]. \( L_c \) is a widely used cross entropy loss in classification tasks. The training objective function can be written as:

\[
L_r = \frac{1}{\Omega(x_M)} \| x_M - y_M \|_2^2, \\
L_c = -\sum_{i=1}^{C} z_i \log(p_i), \\
L = \lambda L_r + L_c,
\]

where \( x \) and \( y \) are the input RGB pixel values and the predicted pixel values, respectively; \( M \) denotes the masked pixels; \( \Omega(\cdot) \) calculates the number of masked pixels; \( \| \cdot \|_2 \) refers to \( \ell_2 \) loss; \( z \) is a one-hot label vector for classification; \( \lambda \) is the balance parameter for the reconstruction branch.

E. Discussion

To alleviate the destruction of spatio-temporal details caused by discarding visual patches, the proposed cell running masking leverages the information redundancy in video data to encourage “information leakage” between frames. As shown in Fig. 5, the cell running masking approach yields smaller reconstruction errors and better classification accuracy. Additionally, to reduce the semantic gap between the VideoMAE [9] pre-trained features and the classification tasks, we propose a bridging classifier. Our experiments in Table VI demonstrates its effectiveness, and the CKA similarity [69] results in Fig. 6 indicate that the bridging classifier plays a vital role in improving the abstraction level of features. Finally, as shown in Fig. 9, compared with the existing advanced efficient action recognition scheme, i.e., AdaFocusV2 [11], MAR does not introduce any additional training strategies, such as reinforcement learning, while outperforms AdaFocusV2 by at least 2.7% with similar computation costs. This suggests that MAR may represent a more concise and efficient approach to action recognition.

IV. Experiments

A. Implementation

1) Datasets: We evaluate the performance of MAR on four widely-used action recognition datasets: Kinetics-400 [7], Something-Something v2 [8], HMDB51 [80], and UCF101 [81]. Kinetics-400 is a large-scale benchmark comprising approximately 240 k training videos and 20 k validation videos from 400 different action categories. Something-Something v2 is a
temporal-related video dataset with 174 action classes, containing 169 k videos for training and 20 k videos for validation. HMDB51 and UCF101 are two smaller datasets for action recognition, each with 3.5 k/1.5 k train/val videos and 9.5 k/3.5 k train/val videos, respectively.

2) Architecture: Following VideoMAE [9] and MAE [18], we utilize ViT [19] with the joint spatio-temporal attention. The attention mechanism with quadratic complexity can lead to computational bottlenecks, while MAR can effectively alleviate this problem. We set the same spatio-temporal patch size (i.e., \(2 \times 16 \times 16\)) as VideoMAE for both ViT-Base and ViT-Large to leverage its pre-trained models conveniently. The spatio-temporal resolution of input videos is \(16 \times 224 \times 224\) for both training and inference. The embedding tokens output by encoders are \(8 \times 14 \times 14 = 1568\). When the mask ratio is set to 50%, the encoders only operate on 784 tokens.

3) Data Pre-Processing and Training Settings: Our MAR training configurations follow the training settings in VideoMAE [9]. We use the same training augmentations as VideoMAE, which are presented in Table I.

| Config                | Sth-Sth v2 | Kinetics-400 |
|-----------------------|------------|--------------|
| Optimizer             | AdamW [70] |              |
| Momentum              | \(\beta_1 = 0.9, \beta_2 = 0.999\) |              |
| Weight Decay          | 0.5        |              |
| Base LR\(^1\)         | 1e-3       |              |
| Batch Size            | 512(B),128(L) |            |
| LR Schedule           | cosine decay [71] |           |
| Layer-wise Decay      | 0.75       |              |
| Filp Augmentation     | yes        |              |
| RandAugment [72]      | (9, 0.5)   |              |
| Mixup [73]            | 0.8        |              |
| Cutmix [74]           | 1.0        |              |
| Label Smoothing [75]  | 0.1        |              |
| DropPath [76]         | 0.1(B), 0.2(L) |       |
| Dropout [77]          | 0.1        |              |
| Warmup Epochs [78]    | 5          |              |
| Training Epochs       | 40         | 100(B),60(L) |
| Repeated Sampling [79] | 1          | 2            |

\(^1\) is linear learning rate scaling rule [78]: ActualLR = BaseLR \times BatchSize/256.

B. Ablation Studies

In this section, we present ablation studies for the in-depth analysis of our proposed complements (i.e., cell running masking and bridging classifier) in MAR. The default encoder is ViT-Base [19] with 16 frames, and other default settings are marked in gray in tables. The notation \(\rho\) in tables means the mask ratio. If not specific, the mask ratio is set to 50% by default.

1) Overall Ablation Studies: We first evaluate the effectiveness of the cell running masking and the bridging classifier on two widely used action recognition datasets, namely Kinetics-400 and Something-Something v2. As in Table II, cell running masking and bridging classifier can achieve stable improvements of approximately 0.6% and about 1.0%, respectively, under various scenarios. These results demonstrate that both techniques can effectively boost the performance of action recognition while only utilizing half of the visible patches. (i.e., the mask ratio is 50%). For efficiency purposes, we use Something-Something v2 as our evaluation benchmark for the subsequent in-depth analysis experiments, unless otherwise specified.

2) Different Mask Sampling Strategies: We replace the cell running masking in MAR with other mask sampling methodologies under different mask ratios \(\rho\). Table III summarizes several observations:

i) Our “cell running masking” provides stable accuracy improvements compared to “random standard masking”. Particularly, larger mask ratios yield greater gains; for example, the gains are 0.18%, 0.62% and 0.67% for mask ratios of 25%, 50% and 75%, respectively. This can be explained by the fewer destroyed contexts with small mask ratios, and cell running masking can better show its advantages with a large proportion of masks;

ii) “block standard masking” and “frame standard masking” respectively indicate the block-wise and frame-wise masking.
masking mentioned in [18]. Both remove chunks of spatio-temporal information and destroy natural spatio-temporal correlations in videos, significantly impairing performance. Despite this, improvements are still observed from “block standard masking” to “block running masking,” demonstrating the effectiveness of our running masking strategy;

iii) Downsampling is a straightforward way to reduce the computational costs. We downsample the width and height of the input videos to half of the standard training settings, i.e., $16 \times 112^2$, and the computation is comparable to that when we set the mask ratio to 75%. Downsampling notably degenerates the performance by 2.23% from cell running masking, indicating that low resolution drops more detailed information than masks;

iv) Compared with “random standard masking,” “uniform running masking” without running cells has limited improvement, while our ‘cell running masking’ performs much stronger. The results validate that our cell running masking can benefit from the diversity brought by various cell states in training.

3) The Spatial Size of Running Cell: Next, we show that our defined running cell plays a critical role in running masking. As shown in Table IV, running masking with a large cell size, such as $14 \times 14$ or $7 \times 7$, exhibits weaker performance than a small cell size, e.g., $2 \times 2$. This is caused by the unitary form of masks. Specifically, to prevent the masks in the large cell from degenerating into a block-wise mask, the first frame should be uniformly initialized. For instance, the cell size of uniform running masking in Fig. 3(c) is $14 \times 14$. Therefore, as discussed in Section III-B, large cell sizes with uniform masks suffer from limited diversities and result in overfitting during training. Our small cell size ($2 \times 2$) is more flexible and achieves the highest performance.

4) Reconstruction Loss and Accuracy: Here, we present the correlation between reconstruction loss and accuracy in Fig. 5. We have observed two intriguing findings. Firstly, there exists an inverse relationship between accuracy and reconstruction loss. For instance, cell running masking results in the highest accuracy with the lowest reconstruction loss, while other masking strategies with larger reconstruction loss may lead to weaker performances. This implies that a small reconstruction loss is capable of recovering rich details for invisible patches and thus enhances the classification accuracy. Secondly, our running strategy with block masking and random masking can reduce the reconstruction error and improve accuracy by approximately 0.3%, indicating that the encoders can leverage more spatio-temporal correlations from the proposed running strategy.

5) Training Augmentations of Cell Running Masking: One of the advantages of a running cell with a small spatial size is its flexibility, meaning that a simple combination of multiple cells can achieve a complex mask. To further investigate this, we ablated different spatial and temporal augmentations during the training in Table V(a). The term “Spatially Random” refers to each cell randomly choosing its starting state, as shown in Fig. 4(a). We observe that the random combinations of cells led to random masking, resulting in poor performance. On the other hand, “Spatially Repeated” indicates that cells share the same randomly selected starting state, as shown in Figs. 3(c) and 4(b), which demonstrates better performance. Additionally, the temporal shuffling slightly increases the randomness, resulting in better performance. This implies that the training masks with reasonable randomness can introduce improvements of about 0.4%, and that simple control of running cells in the spatial and temporal space achieves this regulation.

6) Different Starting States in Testing: There are four different running states in our proposed running cell with a spatial size of $2 \times 2$. Different starting states will produce different masking maps when evaluating the trained models. It is also worth evaluating the impact of various starting states on performance. As shown in Table V(b), the different starting states exhibit comparable performance, and their differences are negligible. This demonstrates that the models trained by MAR are resilient to different states of cell running masking.

7) Cross Validations of Different Mask Ratios: Table V(c) presents the accuracies with different mask ratios for training and testing. We can make the following observations: (i) When testing the models with a large mask ratio, e.g., 75%, the models should also be trained with similar mask ratios. Otherwise, the model cannot learn to compensate for the lost contexts; (ii) appropriately increasing the mask ratio in training can also improve accuracy. For example, training with 25% masks and testing with no mask has an advantage over the training with no mask by 0.34%, i.e., 71.36% vs. 71.02%; (iii) large mask ratios can remarkably save computational costs. Our default setting with only 50% masks can save more than 50% of computation while preserving comparable performance to training and testing with no mask. Thus, our proposed MAR saves both training and testing overhead.

8) The Effect of Bridging Classifier: The comparisons of the proposed bridging classifier and linear classifier under different mask ratios are presented in Table VI. It can be observed that larger mask ratios can lead to more performance degradation. Simply replacing the linear classifier with a bridging classifier brings notable improvements, especially for large mask ratios. The bridging classifier has an advantage over the linear baseline of around 1.0%. This suggests that the mask exacerbates the nonlinearity of the features and confirms our statement that the abstraction level of encoded features still needs to be further bridged for classification.

9) The Abstract Level of Features: In fact, features with lower abstract levels carry more pixel-level information, such as color, and texture, while higher abstract levels contain more semantics for the classification task, such as actors, objects, and...
their interactions. To verify the influence of the bridging classifier on the abstract level of features, as shown in Fig. 6, we plot the CKA similarity [69] between the intermediate features output by the VideoMAE pre-trained blocks and the final features output by the trained bridging classifier. This metric can measure the semantic level of shallow features since the final features used for classification are usually closer to the real semantics. It is clear that the features evolve slowly at the VideoMAE pre-trained blocks. Conversely, the features evolve faster when the features are encoded by the bridging classifier. This demonstrates that there is indeed a semantic gap between the VideoMAE pre-trained features and the classification task, but the bridging classifier can significantly reduce this gap.

10) The Width and Depth Design of Bridging Classifier: The lightweight bridging classifier consists of multiple vanilla Transformer [59] blocks. Table VII illustrates the impact of different decoder designs on performance. Firstly, the Multi-Layer Perceptron (MLP) is widely known for its simple design with non-linear modelling capability. However, its performance is slightly weaker than that of linear classifiers. This indicates that the simple structure of MLP cannot extract abstract information from the encoded features. In contrast, an extremely lightweight decoder, with width = 256 and depth = 1, exhibits a notable improvement of 0.56% in performance, indicating the superior non-linear expressiveness of the Transformer block. Enlarging the width and depth can further boost the performance, with width = 512 and depth = 2 exhibiting the strongest performance. Wider or deeper decoders not only lead to overfitting but also introduce more unnecessary computational burden. Finally, since the decoder operates only visible tokens, the highest performance setting requires only an additional 6.51 GFLOPs (around 8%) compared to the linear classifier.

11) The Input of Bridging Classifier: MAE-based approaches [9, 18, 20] skip the masked tokens in the encoder and apply them with positional embeddings in the lightweight reconstruction decoder. We also evaluate their designs in our proposed bridging classifier in Table VIII. It can be summarized that both the additional positional embeddings and the masked tokens damage the accuracy. This is probably because both factors are low-level priors and thus more specialized for the reconstruction task. In contrast, the high-level semantics required by the classification task is not strongly dependent on these two factors.
12) Reconstruction Target: MAE in both the image domain [20] and video domain [18] demonstrate that reconstructing per-patch normalized pixels works well for self-supervised pre-training. As shown in Table IX, we are interested in whether this finding still holds in MAR. Compared with reconstructing the original video pixels, using normalized pixels [18], [20] always performs better, which is in line with the pre-training settings.

13) The Initialization of Reconstruction Decoder: We utilize VideoMAE [9] pre-trained parameters to initialize the ViT encoder and the reconstruction decoder by default. The effect of the initialization state of the reconstruction decoder is explored in Table IX. It can be observed that the randomly initialized decoder performs decently, while the pre-trained decoder still improves by 0.15%. We speculate that the pre-trained decoder has already converged, which can regulate the encoder directly.

14) The Effect of Reconstruction Loss Weight, i.e., $\lambda$: The reconstruction branch is only involved in the computation during training, and its main purpose is to further enhance the encoder’s ability to perceive the missing context by reconstructing the invisible patches. In Table X(a), we analyze the parameter sensitivity of $\lambda$. When $\lambda$ is set to 0.0, and no reconstruction branch is used, the weakest performance is observed. We note that a slight increase in $\lambda$, i.e., 0.1, leads to improvement. Larger $\lambda$, i.e., 1.0 with a stronger reconstruction constraint, may cause the encoder to preserve more low-level cues in features, resulting in a 0.19% performance degradation.

15) Pre-Training Dataset: In Table X(b), we compare a randomly initialized encoder and various models pre-trained on three different datasets: ImageNet-1K [82], Kinetics-400 [7] and Something-Something v2 [8]. When using the model pre-trained on ImageNet-1 K, the 2D patch embedding layer is inflated to the 3D embedding layer following [9], [23]. By comparing the MAE pre-trained model and the supervised pre-trained model on ImageNet-1 k, we observe that the MAE pre-trained model outperforms the supervised one by 10.74%, which indicates that MAE pre-training guarantees the performance of MAR. This is probably because the completion ability for invisible patches must be pre-trained by the MAE task, which is not available in the supervised pre-trained models. Next, we see that the encoders pre-trained on the video datasets outperform the training from scratch as well as the image-based pre-trained ones. Furthermore, the model pre-trained on Something-Something v2 shows better accuracy than the Kinetics-400 pre-trained one, suggesting that the domain gap between target and pre-training datasets found by VideoMAE [9] still exists in our MAR.

16) Data Augmentations: Although the self-supervised MAE pre-trainings [9], [18], [20] require only multi-scale cropping for training, data augmentation is still one of the crucial factors for Transformer models. If not specified, our used data augmentations follow the settings in VideoMAE [9]. In Table VII, we disable four different data augmentations, i.e., RandAugment [72], Random Erasing [83], MixUp [73] and CutMix [74] to observe the sensitivity of MAR to data augmentation. We observe that each data augmentation can bring a performance gain of 0.2-0.3%. In fact, using masks to delete a proportion of the spatio-temporal patches can also be considered one of the data augmentations. However, since the masked patches can be easily reconstructed, other data augmentation strategies are still needed.

17) Wall-Clock Training Time: Fig. 8 compares the training time consumed by the different training methods on Something-Something v2 dataset. It can be observed that MAR can reduce half of the training cost. Specifically, standard training takes 11.8 hours, while our MAR only needs 5.9 hours and achieves better accuracy than the standard scheme. Although the introduction of the bridging classifier increases the training time slightly, it is still acceptable compared to the standard training scheme.

18) Inference Throughput and Latency: MAR not only accelerates training but also reduces computation costs in inference. As shown in Table XIII, the inference throughput increases in a similar proportion as the FLOPs decreases. Since the codes are not targeted optimization, the latency cannot be further reduced when the mask ratio is increased from 50% to 75%. However, compared with the baseline, the latency still significantly declines. More importantly, the proportionally increased throughput demonstrates that MAR effectively improves the utilization of hardware resources.

### Table IX

| Target             | Initialization of Re.Decoder | Top-1 | Top-5 |
|--------------------|-----------------------------|-------|-------|
| Pixels w/o norm    | Random                      | 70.69 | 92.61 |
| Pixels w/o norm    | VideoMAE pre-trained [9]    | 70.71 | 93.04 |
| Pixels w/ norm     | Random                      | 70.82 | 92.83 |
| Pixels w/ norm     | VideoMAE pre-trained [9]    | 70.97 | 92.75 |

### Table X

(a) **Reconstruction Loss Weight**: $\lambda$ is the balance parameter in (5). Note that the reconstruction branch does not work when $\lambda = 0$. (b) **Pre-Training Datasets and Methods**: “SUP.” is the abbreviation of “SUPERVISED.”

| $\lambda$ | Top-1 | Top-5 |
|-----------|-------|-------|
| 0.0       | 70.72 | 92.79 |
| 0.1       | 70.97 | 92.75 |
| 0.2       | 70.82 | 92.80 |
| 0.4       | 70.82 | 92.93 |
| 1.0       | 70.78 | 92.72 |

### Table XI

| RandAug. | RandEra. | MixUp | CutMix | Top-1 | Top-5 |
|----------|----------|-------|--------|-------|-------|
| ✔️        | ✔️        | ✔️    | ✔️     | 70.61 | 92.60 |
| ✔️        | ✔️        | ✔️    | ✔️     | 70.78 | 92.92 |
| ✔️        | ✔️        | ✔️    | ✔️     | 70.63 | 92.80 |
| ✔️        | ✔️        | ✔️    | ✔️     | 70.70 | 92.77 |
| ✔️        | ✔️        | ✔️    | ✔️     | 70.97 | 92.75 |
TABLE XII
SYSTEM-LEVEL COMPARISONS ON KINETICS-400 ACTION CLASSIFICATION. "ρ" IS THE MASK RATIO. "FLOPS × CR × CL." REFERS TO "FLOPS × CROPS × CLIPS". TO AVOID CONFUSION, WE MARK THE MAE IN THE VIDEO DOMAIN AS "MAE-V".

| Method               | Pre-training Dataset | Supervised Pre-training | Architecture | Input Size | FLOPS × CR × CL. (G) | Param (M) | Top-1 (%) | Top-5 (%) |
|----------------------|----------------------|--------------------------|--------------|------------|----------------------|------------|-----------|-----------|
| NonLocal I3D [84]    | ImageNet-1K ✓        | ResNet101                | ✓            | 128 × 224^2 | 234 × 3 × 10         | 62         | 73.9      | 93.3      |
| TAdaConvNexT-T [32]  | ImageNet-1K ✓        | ConvNexT-T               | ✓            | 32 × 224^2  | 94 × 3 × 4           | 38         | 79.1      | 93.7      |
| Motionformer [85]    | ImageNet-21K ✓       | ViT-L                    | ✓            | 32 × 224^2  | 1185 × 3 × 10        | 382        | 80.2      | 94.8      |
| Video Swin [4]       | ImageNet-1K ✓        | Swin-B                   | ✓            | 32 × 224^2  | 282 × 3 × 4          | 88         | 80.6      | 94.6      |
| TimeFormer [46]      | ImageNet-21K ✓       | ViT-L                    | ✓            | 96 × 224^2  | 8353 × 3 × 1         | 430        | 80.7      | 94.7      |
| ViTFe [6]            | ImageNet-21K ✓       | ViT-L                    | ✓            | 128 × 224^2 | 3980 × 3 × 1         | 81.7       | 93.8      |           |
| Video Swin [4]       | ImageNet-21K ✓       | Swin-L                   | ✓            | 32 × 224^2  | 604 × 3 × 4          | 197        | 83.1      | 95.9      |
| SIFAR [86]           | ImageNet-1K ✓        | Swin-B                   | ✓            | 64 × 224^2  | 270 × 3 × 4          | 381        | 83.5      | 95.7      |
| DTP [87]             | ImageNet-1K ✓        | Swin-B                   | ✓            | 64 × 224^2  | 266 × 3 × 4          | 381        | 83.5      | 95.7      |
| SIFAR [88]           | ImageNet-21K ✓       | SIFAR-L-12               | ✓            | 8 × 224^2   | 944 × 3 × 1          | 196        | 84.2      | 96.0      |
| Uniformer [48]       | ImageNet-1K ✓        | Uniformer-B              | ✓            | 32 × 224^2  | 259 × 3 × 4          | 84         | 96.9      |           |
| MTD^P [40]           | -                    | Inception-v1             | ✓            | 64 × 224 × 224 | 115 × 3 × 10        | 27         | 77.4      | 92.9      |
| LgNet [89]           | -                    | ResNet5                  | ✓            | 64 × 128 × 128 | 63 × 3 × 10         | 31         | 77.6      | 93.0      |
| TCLC [36]            | -                    | R101                     | ✓            | 128 × 224^2 | N/A                  | N/A        | 79.2      | 93.7      |
| SlowFast [1]         | -                    | R101+NL                  | ✓            | (16 + 64) × 224^2 | 359 × 3 × 10        | 60         | 78.9      | 93.9      |
| BEVT [67]            | IN-1K+DALLE           | Swin-B                   | ✓            | 32 × 224^2  | 282 × 3 × 5          | 88         | 80.6      | N/A       |
| MViT [5]             | -                    | MViT-B                   | ✓            | 64 × 11 × 11 | 455 × 1 × 1          | 37         | 81.2      | 95.1      |
| MaskFeat [66]        | Kinetics-400          | MViT-L                   | ✓            | 16 × 224^2  | 377 × 1 × 1          | 218        | 84.3      | 96.3      |

TABLE XIII
INFERENCE THROUGHPUT AND LATENCY MEASURED ON ACTUAL HARDWARE. RESULTS ARE OBTAINED USING ONE V100-32 GB GPU. THE BATCH SIZES FOR THROUGHPUT AND LATENCY ARE 32 AND 1, RESPECTIVELY.

| Method | Accuracy (%) | GFLOPs | Throughput (Vd/s) | Latency (ms) |
|--------|--------------|--------|-------------------|--------------|
| VideoMAE [9] | 70.3 | 180 | 51.15 | 28.0 |
| MAR^p=75% | 69.5 | 41 (0.23×) | 226.15 (4.42×) | 16.4 (0.58×) |
| MAR^p=50% | 71.0 | 86 (0.47×) | 115.61 (2.26×) | 16.8 (0.60×) |

C. Comparisons With the Previous Methods

Tables XII and XIV compare our training scheme with other state-of-the-art methods on Kinetics-400 [7] and Something-Something v2 [8]. The relevant settings are listed in detail for comparison, including network architectures and calculation costs. We can draw the following observations from the table. First, with the same encoders, i.e., pre-trained by VideoMAE, MAR can consistently outperform the standard training scheme by at least 0.3% with only around 47% of GFLOPs. Second, MAR can easily scale up to large models and achieve more improvements. The ViT-Large trained by MAR on Kinetics-400 dataset surpasses standard training by 1.4%. With a large mask ratio (i.e., 75%), our ViT-Large has an advantage over the standard trained ViT-Base of 2.6% (81.3% vs. 83.9%) on Kinetics-400, while our ViT-Large saves 27% of computation overhead (180 GFLOPs vs. 131 GFLOPs). Moreover, it is also worth noting that our trained ViT-Large models even exceed ViT-Huge by 0.2% on Kinetics-400 and Something-Something v2. Especially, our ViT-Large only costs 14% GFLOPs compared to ViT-Huge with standard training. Third, the models trained by MAR achieve superior performance on both datasets compared to previous approaches under similar GFLOPs, even though they use the supervised pre-training on larger datasets.

19) Visualizations: We qualitatively visualize the reconstructed videos by the reconstruction branch in Fig. 7. We observe that even when the cell running masking discards 50% of the spatio-temporal patches, the reconstructed videos can still fully express the high-level semantics of the videos. This suggests that it is not necessary for the encoder to operate on all spatio-temporal patches.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fourth, as analyzed in Fig. 6, since the bridging classifier can effectively improve the semantic level of VideoMAE pre-trained features, replacing the linear classifier with a bridging classifier can significantly boost performance (e.g., from 80.7% to 81.3% on Kinetics-400 with ViT-B). However, compared with MAR, the gains are only around 0.3% on average, while the computation costs are more than 220% of MAR.

In Fig. 9, we provide an intuitive comparison with the state-of-the-art efficient action recognition approach, i.e., AdaFocusV2, which also removes spatio-temporal redundancy in videos for efficiency. It can be observed that MAR surpasses AdaFocusV2 by notable margins with similar computation costs. For example, MAR outperforms AdaFocusV2 by 2.7% (63.2% vs. 60.5%) with about 23 GFLOPs. Moreover, MAR does not introduce any
We speculate that the small datasets with limited videos can be easily reconstructed by models, thus leading to overfitting, especially for UCF101 datasets with simple backgrounds. Therefore, our MAR is more superior on large-scale datasets.

V. CONCLUSION

In this work, we propose Masked Action Recognition (MAR), a simple and computationally friendly training scheme for vanilla Vision Transformers (ViTs) in videos. MAR is investigated from two perspectives of ViTs, i.e., reducing the number of input patches and bridging the semantic gaps of output features. For the former, the cell running masking strategy is designed to generate spatio-temporal interleaved masks, which preserves the spatio-temporal correlations in videos. For the latter, the lightweight bridging classifier is proposed to bridge the semantic gaps between encoded features and specialized classification features. Empirical results show that MAR costs only 47% of the computation and exceeds the performance of the standard training scheme. In addition, strong generalizations of MAR have also been demonstrated on several video datasets with different scales. Overall, this work exploits the powerful context modelling capability of ViTs, and significantly improves the training and testing efficiency with better performance. In future works, to further save computational costs with a less performance penalty, semantic-based mask ratios and masking maps are worthwhile prospects to be explored.

REFERENCES

[1] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “Slowfast networks for video recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6202–6211.
[2] C. Feichtenhofer, “3D: Expanding architectures for efficient video recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 203–213.
[3] L. Wang, Z. Tong, B. Ji, and G. Wu, “TDN: Temporal difference networks for efficient action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 1895–1904.
[4] Z. Liu et al., “Video swin transformer,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 3202–3211.
[5] H. Fan et al., “Multiscale vision transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 6824–6835.
[6] A. Azab et al., “VViT: A video vision transformer,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 6836–6846.
[7] W. Kay et al., “The kinetics human action video dataset,” 2017, arXiv:1705.06950.
[8] R. Goyal et al., “The “something something” video database for learning and evaluating visual common sense,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2017, pp. 5842–5850.
[9] Z. Tong, Y. Song, J. Wang, and L. Wang, “VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2022, pp. 7780–7789.
[10] Y. Wang et al., “Adaptive focus for efficient video recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 16249–16258.
[11] Y. Wang et al., “Adafocus V2: End-to-end training of spatial dynamic networks for video recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2022, pp. 20030–20040.
[12] Z. Wu, C. Xiong, C.-Y. Ma, R. Socher, and L. S. Davis, “AdaFrame: Adaptive frame selection for fast video recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1278–1287.
[13] B. Kochar, D. Tran, and L. Torresani, “SSSampler: Sampling salient clips from video for efficient action recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6232–6242.
[14] W. Wu, D. He, X. Tan, S. Chen, and S. Wen, “Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6222–6231.

Table XV

Comparisons to Other State-of-the-art Methods on UCF101 and HMDB51

| Method         | Backbone | Extra Data | UCF101 | HMDB51 |
|----------------|----------|------------|--------|--------|
| VideoMAE [9]   | ViT-B    |            | 90.8   | 61.4   |
| MAR [9] [14%]  | ViT-B    |            | 110.7  | 64.1   |
| MemDPC [9]     | R-2D3D   | K400       | 86.1   | 54.5   |
| CoCLR [90]     | S3D-G    | UCF101     | 87.9   | 54.6   |
| SeCo [94]      | TSN-R50  | K400       | 88.3   | 55.6   |
| ViCLR [91]     | S3D-G    | K400       | 91.3   | 63.4   |
| ParamCrop [95] | S3D-G    | K400       | 93.5   | 68.0   |
| CORP [96]      | I3D      | K400       | 93.6   | 68.0   |
| RSPNet [97]    | S3D-G    | K400       | 93.9   | 64.7   |
| HiCo [98]      | S3D-G    | UK400      | 94.6   | 70.6   |
| CVRL [99]      | Slow-R152| K600       | 94.7   | 72.1   |
| ρBYOL [100]    | Slow-R50 | R500       | 95.5   | 74.1   |

In addition, comparisons on two smaller video datasets, UCF101 [81] and HMDB51 [80] are also reported in Table XV. It can be observed that our MAR still leads to on-par or better performance under multiple settings on these two datasets.

Fig. 8. The training time (32 V100 GPUs) vs. validation error rate on Something-Something V2. “C.R.M.(50%)” means the cell running masking with a mask ratio of 50%. “Linear” and “Bridging” refer to the linear classifier and bridging classifier, respectively.

Fig. 9. AdaFocusV2 [11] v.s. MAR on Something-Something V2 in terms of inference efficiency. The two algorithms adopt one center-crop for evaluation. We implement MAR on top of ViT-B, and reduce the FLOPs by increasing the mask ratio.
[37] S. Liu and X. Ma, “Attention-driven appearance-motion fusion network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog., 2021, pp. 6155–6164.

[40] J. Wang, Y. Lin, M. Zhang, Y. Gao, and A. J. Ma, “Multi-level temporal dilated dense prediction for action recognition,” IEEE Trans. Multimedia, vol. 24, pp. 2553–2566, 2021.

[41] J. Lin, C. Gan, and S. Han, “TSM: Temporal shift module for efficient video understanding,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 7083–7093.

[42] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 10012–10022.

[43] H. Touvron et al., “Training data-efficient image transformers & distillation through attention,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 10347–10357.

[44] D. Neiman, O. Bar, M. Zohar, and D. Asselmann, “Video transformer,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 3163–3172.

[45] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?,” in Proc. Int. Conf. Mach. Learn., 2021, Art. no. 4.

[46] Y. Li et al., “MvitV2: Improved multiscale vision transformers for classification and detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog., 2022, pp. 4804–4814.

[47] K. Li et al., “Universal Transformer for efficient spatiotemporal representation learning,” in Proc. Int. Conf. Mach. Learn., 2022.

[48] Y.-D. Zheng, Z. Liu, T. Lu, and L. Wang, “Dynamic sampling networks for efficient action recognition in videos,” IEEE Trans. Image Process., vol. 29, pp. 7970–7983, 2020.

[49] H. Fan et al., “Watching a small portion could be as good as watching all: Towards efficient video classification,” in Proc. Int. Joint Conf. Artif. Intell., 2018, pp. 705–711.

[50] W. Xu et al., “Dynamic inference: A new approach toward efficient video action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. Workshops, 2020, pp. 676–677.

[51] Y. Zhi, Z. Tong, L. Wang, and G. Wu, “MGSampler: An explainable sampling strategy for video action recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 1513–1522.

[52] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 1096–1103.

[53] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc., 2019, pp. 4171–4186.

[54] H. Bao et al., “UnilmV2: Pseudo-masked language models for unified language model pre-training,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 642–652.

[55] Y. Liu et al., “Roberta: A robustly optimized BERT pretraining approach,” 2019, arXiv:1907.11692.

[56] Z. Yang et al., “XLNet: Generalized autoregressive pretraining for language understanding,” in Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 5755–5763.

[57] L. Dong et al., “Unified language model pre-training for natural language understanding and generation,” in Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 13063–13075.

[58] A. Vaswani et al., “Attention is all you need,” in Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 6000–6010.

[59] M. Chen et al., “Generative pretraining from pixels,” in Proc. Proc. Int. Conf. Mach. Learn., 2020, pp. 1691–1703.

[60] Z. Xie et al., “Simmim: A simple framework for masked image modeling,” in Proc. Proc. Int. Conf. Mach. Learn., 2020, pp. 9653–9663.

[61] D. Neiman et al., “Universal Transformer for efficient spatiotemporal representation learning,” Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 5755–5763.

[62] L. Dong et al., “Unified language model pre-training for natural language understanding and generation,” Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 13063–13075.

[63] A. Vaswani et al., “Attention is all you need,” Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 6000–6010.

[64] M. Chen et al., “Generative pretraining from pixels,” Proc. Proc. Int. Conf. Mach. Learn., 2020, pp. 1691–1703.

[65] Z. Xie et al., “Simmim: A simple framework for masked image modeling,” Proc. Proc. Int. Conf. Mach. Learn., 2020, pp. 9653–9663.

[66] D. Neiman et al., “Universal Transformer for efficient spatiotemporal representation learning,” Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 5755–5763.

[67] L. Dong et al., “Unified language model pre-training for natural language understanding and generation,” Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 13063–13075.

[68] A. Vaswani et al., “Attention is all you need,” Proc. Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 6000–6010.

[69] M. Chen et al., “Generative pretraining from pixels,” Proc. Proc. Int. Conf. Mach. Learn., 2020, pp. 1691–1703.

[70] Z. Xie et al., “Simmim: A simple framework for masked image modeling,” Proc. Proc. Int. Conf. Mach. Learn., 2020, pp. 9653–9663.
S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, “Similarity of neural network representations revisited,” in Proc. Int. Conf. Mach. Learn., 2019, vol. 97, pp. 3519–3529.

I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in Proc. Int. Conf. Learn. Representations, 2019.

I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” in Proc. Int. Conf. Learn. Representations, 2017.

E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 702–703.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization,” in Proc. Int. Conf. Learn. Representations, 2018.

S. Yun et al., “Cutmix: Regularization strategy to train strong classifiers with localizable features,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2019, pp. 6023–6032.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2826.

G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep networks with stochastic depth,” in Proc. 14th Eur. Conf. Comput. Vis., 2016, pp. 646–661.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

P. Goyal et al., “Accurate, large minibatch SGD: Training imagenet in 1 hour,” 2017, arXiv:1706.02677.

E. Hoffer et al., “Augment your batch: Improving generalization through instance repetition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8129–8138.

H. Kuehne, H. Jhuang, E. Garrate, T. Poggio, and T. Serre, “HMDB: A large video database for human motion recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2011, pp. 2556–2563.

K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” 2012, arXiv:1211.0402.

J. Deng et al., “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.

Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, pp. 13001–13008.

X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7794–7803.

M. Patrick et al., “Keeping your eye on the ball: Trajectory attention in video transformers,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2021, pp. 12493–12506.

F. Long et al., “Stand-alone inter-frame attention in video models,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 3192–3201.

F. Long et al., “Dynamic temporal filtering in video models,” in Proc. 17th Eur. Conf. Comput. Vis., 2022, pp. 475–492.

Q. Fan, C.-F. Chen, and P. Panda, “Can an image classifier suffice for action recognition?” in Proc. Int. Conf. Learn. Representations, 2022.

J. Zhou, Z. Fu, Q. Huang, Q. Liu, and Y. Wang, “LGNet: A local-global network for action recognition and beyond,” IEEE Trans. Multimedia, early access, Jul. 07, 2022, doi: 10.1109/TMM.2022.3189253.

F. Han, W. Xie, and A. Zisserman, “Self-supervised co-training for video representation learning,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2020, vol. 33, pp. 5679–5690.

A. Diba et al., “Vi2CLR: Video and image for contrastive learning of representation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 1502–1512.

Z. Huang et al., “Self-supervised motion learning from static images,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 1276–1285.

T. Han, W. Xie, and A. Zisserman, “Memory-augmented dense predictive coding for video representation learning,” in Proc. 16th Eur. Conf. Comput. Vis., 2020, pp. 312–329.

T. Yao, Z. Zhang, Y. Qu, Y. Fan, and T. Mei, “SECO: Exploring sequence supervision for unsupervised representation learning,” in Proc. AAAI Conf. Artif. Intell., 2021, vol. 35, pp. 10656–10664.

Z. Qing et al., “ParamCrop: Parametric cubic cropping for video contrastive learning,” IEEE Trans. Multimedia, early access, 2023, doi: 10.1109/TMM.2023.3244126.

K. Hu et al., “Contrast and order representations for video self-supervised learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 7939–7949.

P. Chen et al., “RSPNet: Relative speed perception for unsupervised video representation learning,” in Proc. AAAI Conf. Artif. Intell., 2021, vol. 35, pp. 1045–1053.

Z. Qing et al., “Learning from untrimmed videos: Self-supervised video representation learning with hierarchical consistency,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 13821–13831.

R. Qian et al., “Spatiotemporal contrastive video representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 6964–6974.

C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, and K. He, “A large-scale study on unsupervised spatiotemporal representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 3299–3309.

Zhiwen Qing received the B.S. degree in 2019 from the Huzhou University of Science and Technology, Wuhan, China, where he is currently working toward the Ph.D. degree in pattern recognition and intelligent systems. His research interests include action recognition, self-supervised video representation learning.

Shiwei Zhang received the Ph.D. degree from the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently a Researcher of computer vision with Alibaba Group, DAMO Academy, Hangzhou, China. His research interests include video understanding, video generation, multimodal representation learning, and machine learning.

Ziyuan Huang received the B.Eng. degree in vehicle engineering from Tongji University, Shanghai, China, in 2019. He is currently working toward the Ph.D. degree with Advanced Robotics Centre, National University of Singapore, supervised by Professor Marcelo Ang. His research interests include video understanding, including action recognition and localization, video representation learning, multi-modal learning, and video-based scene understanding.

Xiang Wang received the B.S. degree in 2020 from the Huazhou University of Science and Technology, Wuhan, China, where he is currently working toward the Ph.D. degree in pattern recognition and intelligent systems. His research interests include few-shot action recognition, semi-supervised learning, and action detection.
Yuehuan Wang received the graduation degree from the University of Electronic Science and Technology of China, Chengdu, China, in 1993, and the M.S. degree in computer system architecture and the Ph.D. degree in pattern recognition and artificial intelligence from the Huazhong University of Science and Technology, Wuhan, China, in 1996 and 2001, respectively. He is currently a Visiting Scholar with the University of Bordeaux III, Bordeaux, France, and Washington University, St Louis, MO, USA. He is currently a Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests include computer vision, image understanding, target tracking, and automatic target recognition.

Yiliang Lv received the M.S. degree from the school of Automation, University of Science and Technology, Beijing, China, in 2014. He is currently a Researcher of computer vision with Alibaba Group, DAMO Academy, Beijing. His research interests include video understanding, multi-modal information retrieval and machine learning.

Changxin Gao (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, China, in 2010. He is currently an Associate Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests include pattern recognition and surveillance video analysis.

Nong Sang (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, China, in 2000. He is currently a Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests include low-quality image enhancement, object detection and recognition, object tracking, image/video semantic segmentation, action detection and recognition.