IT TAKES TWO: MASKED APPEARANCE-MOTION MODELING FOR SELF-SUPERVISED VIDEO TRANSFORMER PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised video transformer pre-training has recently benefited from the mask-and-predict pipeline. They have demonstrated outstanding effectiveness on downstream video tasks and superior data efficiency on small datasets. However, temporal relation is not fully exploited by these methods. In this work, we explicitly investigate motion cues in videos as extra prediction target and propose our **M**asked **A**ppearance-**M**otion **M**odeling (MAM^2) framework. Specifically, we design an encoder-regressor-decoder pipeline for this task. The regressor separates feature encoding and pretext tasks completion, such that the feature extraction process is completed adequately by the encoder. In order to guide the encoder to fully excavate spatial-temporal features, two separate decoders are used for two pretext tasks of disentangled appearance and motion prediction. We explore various motion prediction targets and figure out RGB-difference is simple yet effective. As for appearance prediction, VQGAN codes are leveraged as prediction target. With our pre-training pipeline, convergence can be remarkably speed up, *e.g.,* we only require 2$\times$ fewer epochs than state-of-the-art VideoMAE (400 *v.s.* 800) to achieve the competitive performance. Extensive experimental results prove that our method learns generalized video representations. Notably, our MAM^2 with ViT-B achieves 82.3% on Kinects-400, 71.3% on Something-Something V2, 91.5% on UCF101, and 62.5% on HMDB51.

1 INTRODUCTION

With regard to a variety of video tasks, the Vision Transformer (Dosovitskiy et al., 2020) has achieved impressive results in supervised and unsupervised settings. For supervised learning, ViViT (Arnab et al., 2021), TimeSformer (Bertasius et al., 2021), Video Swin (Liu et al., 2021a), and MViT (Fan et al., 2021) are emerging and achieve favorable performance in video understanding related tasks by using labeled data. On the other hand, contrastive learning based self-supervised methods, e.g., MoCo (He et al., 2020), SimCLR (Chen et al., 2020b) and BYOL (Grill et al., 2020) etc., have also been widely extended to video representation learning. In (Feichtenhofer et al., 2021), a large-scale study on spatial-temporal representation learning using transformers and convolutional neural networks are conducted. More recently, a series of methods which explore masked video modeling (Wang et al., 2022b; Tong et al., 2022; Feichtenhofer et al., 2022b) have been proven to be promising for self-supervised representation learning. Specifically, the *mask-and-predict* pipeline of MAE (He et al., 2021) is extended to the video domain by ST-MAE (Feichtenhofer et al., 2022b) and VideoMAE (Tong et al., 2022), which reconstruct the raw pixels of masked video patches from the visible contexts in a spacetime-agnostic manner. Besides, BEVT (Wang et al., 2022b) jointly reconstructs the discrete visual tokens obtained by an offline trained tokenizer in the image and video domain.

However, the information density of video is much lower than image. Although ST-MAE (Feichtenhofer et al., 2022b) and VideoMAE (Tong et al., 2022) employ a high-ratio masking strategy to reduce the redundancy in the Spatio-temporal domain and holistically comprehend the video beyond low-level visual statistics, they are still constrained in their visual contents and suffer from the inconsistency between the masked pre-training with space-aware target and the video understanding task. To the best of our knowledge, no research has previously been done on how to leverage potential motion clues in video to guide pre-training in the *masked video modeling* manner.

Based on the aforementioned points, we creatively use two disentangled decoders to simultaneously reconstruct the visual appearance and motion targets. Specifically, we first study on the various motion cues that are concealed in the video data, such as optical flow, RGB difference, and temporal order (Xu et al., 2019). Then we employ visual tokens generated by the discrete variational autoencoder (dVAE) (Esser et al., 2021; Van Den Oord et al., 2017) as a target for visual reconstruction that offers supplementary semantic clue to motion information during the pre-training phase. Intuitively, we design a pair of disentangled decoders, one to reconstruct the visual tokens and the other to reconstruct the motion target. By forcing the encoder to learn inherent spatio-temporal relation in video data, two independent disentangled decoders enable the encoder to be more effectively transferred to downstream tasks. Additionally, decoupling the appearance and motion view on the target-level allows for the efficient handling of redundant spatio-temporal data, speeding up convergence in the pre-training phase as shown in Fig.2.

To eliminate the discrepancy between the view of appearance and motion, we adopt a regressor in (Chen et al., 2022) to further project the latent representation from the encoder in a spacetime-agnostic manner. In more detail, the regressor learns to query from embeddings of the visible tubes and predict the representations for masked video tubes. These predicted representations of masked tubes are constrained to be aligned with their groundtruth encoding output and are then utilized for completing pretext tasks. With the regressor, the feature encoding and pretext tasks completion are explicitly separated. Therefore, our MAM^2 prevents the encoder from being entangled in different objectives of reconstruction during pre-training, guiding the encoder to focus on spatial-temporal representation learning and fully excavate spatial-temporal features.

With our MAM^2 pre-training framework, we empirically show that convergence can be remarkably speed up and the learned video representations generalize very well on multiple standard video recognition benchmarks. In a nutshell, our contributions are as follows:

- We decouple the visual appearance and motion views to focus on different target in the vein of self-supervised video pre-training. As far as we are aware, we are the first to explore practical and efficient motion cues in the *Masked Video Modeling* pre-training paradigm.
- We propose a novel Masked Appearance-Motion Modeling (MAM^2) framework which reconstructs the appearance and motion targets separately. MAM^2 learns generalized spatio-temporal representation which better benefits the downstream video tasks. Additionally, we adopt a spacetime-agnostic regressor to avoid the encoder being entangled with pretext tasks, thus focusing on spatial-temporal feature excavation.
- Extensive investigation results on standard video benchmarks demonstrate that our MAM^2 achieves a significant performance improvement than previous state-of-the-art methods with even half of pre-training epochs. Specifically, our MAM^2 brings the gain of 1.0% on Kinetics-400, 0.7% on Something-Something V2 with a 2$\times$ pre-training epochs speedup, and obtains the improvement of 0.7% on UCF101 and 1.4% on HMDB51 with a 3$\times$ pre-training epochs speedup.

2 APPROACH

We propose the Masked Appearance-Motion Modeling (MAM^2) framework for self-supervised video representation learning. Our MAM^2 equips with the encoder-regressor-decoder architecture to prevent the encoder from becoming involved in the reconstruction task. To excavate the spatial-temporal features in the video stream, we establish two distinct pretext tasks to independently decode the visual and motion information of video data to guide the encoder learning process. In Sec. 2.1, we describe the architecture design of our MAM^2. We then go over several potential motion objectives for pre-training before introducing the optimization function in Sec. 2.2.

2.1 ARCHITECTURE

The architecture of our method is depicted in Fig.1. The encoder is pre-trained for video representation learning. A latent contextual regressor learns to query from embeddings of the visible tubes and predict the representations for masked video tubes. These predicted representations of masked tubes are constrained to be aligned with their groundtruth encoding output and are then utilized

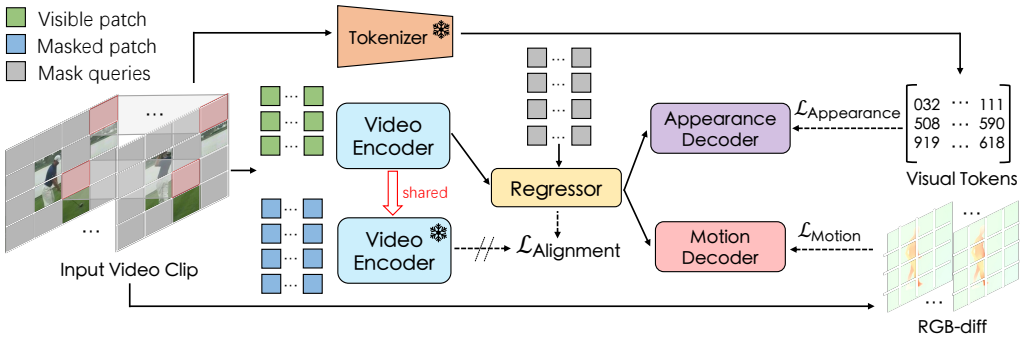

Figure 1: Overall architecture of our MAM^2 pretraining framework, which is an encoder-regressor-decoder pipeline.

for completing pretext tasks. Dual-stream disentangled decoders are leveraged for reconstructing appearance and motion pre-training targets.

Input. A video clip $\mathbf{X} \in \mathbb{R}^{T\times 3\times H\times W}$ as input is fed into the encoder, where T is temporal length and spatial resolution is $H \times W$. Following the protocol in ViT (Dosovitskiy et al., 2020) and TimeSformer (Bertasius et al., 2021), the input video clip is divided into $T \times N$ non-overlapping spatio-temporal patches with spatial size of $P \times P$, where $N = HW/P^2$. In particular, the video clip $\mathbf{X}$ is fed through a 2D convolution layer to embed patch tokens $\mathbf{X}^z \in \mathbb{R}^{TN\times D}$.

To maintain the spatio-temporal information of video data at the same time during the training procedure, we append learnable temporal positional embeddings $\mathbf{e}^t \in \mathbb{R}^{T\times D}$ and spatial positional embeddings $\mathbf{e}^s \in \mathbb{R}^{N\times D}$ to each input token. All input patches within the i^{th} ($i \in \{0, 1, ..., T-1\}$) frame (but different spatial locations) share the same temporal positional embedding $\mathbf{e}^t_i$, while all patches in the same j^{th} spatial location ($j \in \{0, 1, ..., N-1\}$), regardless of their frame indexes, are given the same spatial positional embedding $\mathbf{e}^s_j$. In our MAM^2, we adopt a "random tube masking" strategy which simply broadcasts a 2-D random mask to all time steps of frames. Our MAM^2 takes mask ratio ρ of 75% to perform random tube masking. It is illustrated in appendix.

Encoder. The encoder is a stack of factorized space-time attention blocks Bertasius et al. (2021) and it maps the embedded patches into their latent representations. The output of the space-time attention block is obtained by sequentially applying temporal and then spatial self-attention on its input. This architecture can well learn spatio-temporal representation of video frames, in addition, it significantly reduces computation and memory complexity.

Regressor and alignment. The process of regressor and latent representation alignment is analogous to context autoencoder (Chen et al., 2022). Latent contextual regressor is a stack of cross-attention blocks. In particular, each cross-attention block takes the representations of visible patches as key and value and the mask queries as query to regress representations for masked patches. We make the mask queries learnable in the training procedure. Aligning the regressed representations of the masked patches with the groundtruth representations produced by the encoder is achieved with the following constraint:

$$\mathcal{L}_{\text{Alignment}} = \frac{1}{|\mathcal{M}|}\sum_{p\in\mathcal{M}} \|\mathbf{r}_p - \hat{\mathbf{r}}_p\|_2^2, \tag{1}$$

where $\mathcal{M}$ is set of masked tokens, $|.|$ is the number of elements, p is the masked token index, $\mathbf{r}_p$ is the regressed representation and $\hat{\mathbf{r}}_p$ is the alignment target which is produced by feeding the groundtruth masked patch into the encoder. Eventually, it is the regressed representations of the masked patches that are used for pre-text tasks completion. The regressor accompanied by alignment constraint ensures the encoder focuses on representative space-time feature abstraction and explicitly avoids the decoders involving in feature encoding. In this way, the encoder is driven by pretext tasks to excavate powerful spatial-temporal features.

Decoders. Two pretext tasks are specifically implemented by dual-stream decoders in our MAM^2. One is an appearance decoder to decode visual tokens for the masked patches, and the other is a motion decoder to predict the RGB difference for masked regions. Similar to the encoder, both

decoders consist of a stack of factorized time and space attention blocks. Latent representations based on masked queries generated by the regressor serve as the input for decoders, avoiding direct utilization of the information contained in the visible tokens. As a result, the encoder is guided by the appearance and motion decoders to fully exploit spatio-temporal representation since it needs to simultaneously optimize the tasks of visual and motion prediction.

2.2 PRE-TRAINING OBJECTIVE

Appearance and Motion targets. For the appearance stream, we use a discrete variational autoencoder (dVAE) proposed in (Esser et al., 2021) to generate the discrete visual tokens of masked patches as the prediction targets of appearance decoder. On the other hand, we investigate three different types of motion targets on the motion stream:

- **Optical flow.** We extract on-the-fly optical flow by using a pre-trained deep network, Recurrent All-Pairs Field Transforms (RAFT) (Teed & Deng, 2020), on the consecutive sampled frames. By using optical flow as a target, encoders can be forced to extract short-term motion cues between video frames more effectively. However, using RAFT is computationally expensive, thus we investigate RGB difference as a more lightweight alternative.
- **RGB difference.** We simply compute the difference of the raw RGB values between adjacent video frames to obtain RGB difference target. This is an effective and efficient way to capture short-term motion pattern and also provide local motion statistic as a complement to the appearance target. In particular, our method simply predict the RGB difference between t th frame and $t-1$th frame on masked patches for $\forall t \in \{1, 2, ..., T-1\}$.
- **Clip order prediction.** In contrast to the motion modality discussed above, which focus on short-term motion cues, we explore long-range temporal prediction target. Specifically, we utilize the video clip order prediction task (Xu et al., 2019; Wang et al., 2021c) to predict the correct order of the randomly shuffled latent representations. The input of the motion decoder at each masked tube position is a tuple of temporally shuffled feature clips generated by the regressor for the masked tube, and we use a [CLS] token on motion decoder and its corresponding output representation is projected to a probability distribution over all possible feature clip orders. Actually, we divide and randomly shuffle the latent feature produced by the regressor to 2 clips (2 possible clip orders) at each masked tube.

Loss Function. For the appearance decoder, we use the off-the-shelf dVAE tokenizer to generate the discrete tokens as the targets. The discrete token label is assigned to each masked patch. Consequently, the target of appearance decoder is a K-class label where K is 16384. After a linear prediction head $\mathbf{W}_m \in \mathbb{R}^{D \times K}$, where D is 768 for ViT-Base, a softmax cross entropy loss is performed on the logits:

$$\mathcal{L}_{\text{Appearance}} = -\frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \sum_{j=1}^{K} y_{m(i,j)} \log(p_{i,j}), \tag{2}$$

where $y_{m(i,j)} \in \{0, 1\}$ indicates the ground-truth label of sample i is class j or not, $p_{i,j}$ indicates predicted probability of sample i in class j.

For motion decoder, we simply compute the RGB difference $\hat{\mathbf{D}} \in \mathbb{R}^{(T-1) \times \rho N \times C}$ of input T frames at the masked regions. The target is to reconstruct the RGB difference of masked tokens. MSE loss is also adopted here to optimize this decoder:

$$\mathcal{L}_{\text{Motion}} = \frac{1}{|\mathcal{M}'|} \sum_{p \in \mathcal{M}'} \|\mathbf{D}_p - \hat{\mathbf{D}}_p\|_2^2 \tag{3}$$

where $\mathcal{M}'$ is set of masked patches except for the ones of the T^{th} frame, $|.|$ is the number of elements, p is the index of masked patches and $\mathbf{D}_p$ is the decoder prediction. Finally, the hybrid loss is shown as follows:

$$\mathcal{L} = \mathcal{L}_{\text{Appearance}} + \mathcal{L}_{\text{Motion}} + \alpha \mathcal{L}_{\text{Alignment}} \tag{4}$$

where α is set to 2 in our experiments.

3 RELATED WORK

Video Recognition. Research on video recognition has gone through rapid development, thanks to the availability of large-scale video datasets, e.g., Kinetics (Carreira & Zisserman, 2017; Carreira et al., 2018; 2019) and Something-Something (Goyal et al., 2017). Convolutional networks have long dominated in video understanding tasks (Tran et al., 2015; 2018; Feichtenhofer et al., 2019; Lin et al., 2019; Feichtenhofer, 2020; Wu et al., 2021). With the seminal work of Vision Transformer(ViT) (Dosovitskiy et al., 2020; Touvron et al., 2021), researches on backbone architecture for video understanding have gradually shifited from CNNs to Transformers. TimeSformer (Bertasius et al., 2021) and ViViT (Arnab et al., 2021) suggest the factorized spacetime attention for a strong speed-accuracy tradeoff. Video SwinTransformer (Liu et al., 2021a) and MViT (Fan et al., 2021) introduce the hierarchical structure to reduce computational cost. MTV (Yan et al., 2022) separates encoders to represent each perspective of the input video. In this work, we follow the simple recipe of factorized spacetime attention and focus on exploring the pre-training of video transformer in a *masked video modeling* manner.

Self-supervised Video Representation Learning. For self-supervised video representation pretraining (SSVP), various pretext tasks are often designed based on the image contrastive self-supervised methods to learn spatiotemporal video representation including (Kuang et al., 2021; Yao et al., 2021; Bai et al., 2020; Wang et al., 2021a; Fang et al., 2022) and etc. Also, transformers remain the focus of video SSVP research, (Ranasinghe et al., 2022) and (Wang et al., 2022a) explore intriguing properties of vision transformers (Dosovitskiy et al., 2020) in the video domain while (Recasens et al., 2021; Xu et al., 2021a;b) jointly model multiple modalities with transformers. There are also some dense contrastive learning methods which have shown promissing results on video-related downstream tasks. In particular, DenseCL (Wang et al., 2021d), PixPro (Xie et al., 2021) perform pairwise contrastive learning at pixel-level where corresponding pixels are encouraged to be consistent.

On the other hand, masked visual autoencoders have also been proposed to learn effective video representations based on the mask-and-reconstruct pipeline due to its great success on image domain. Among the masked image modeling (MIM) pre-training schemes, MAE (He et al., 2021), BEiT (Bao et al., 2021) and CAE (Chen et al., 2022) are most popular ones. There are concurrent works mostly related to ours. ST-MAE (Feichtenhofer et al., 2022b) and VideoMAE (Tong et al., 2022) directly extend MAE framework to the video domain by reconstructing the raw pixels of masked video patches from the visible contexts in a spacetime-agnostic manner. BEVT (Wang et al., 2022b) jointly reconstructs the discrete visual tokens obtained by an offline tokenizer in the image and video domain. In contrast to previous methods, our study firstly reveals that explicitly introducing motion cues related pretext task brings more efficient and transferable spatiotemporal representations learning.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUPS

Architecture. We begin with the vanilla ViT architectures (Dosovitskiy et al., 2020), then we employ the factorzied space-time attention (Arnab et al., 2021) to build our video encoder. We adopt the ViT base and large architecture as our encoder. *Please refer to Appendix for more details of the architectures.* To minimize the parameter scale for positional embeddings, we add separable space and temporal positional embeddings to the patch tokens.

To jointly extract space-time features, we use a latent contextual regressor (Chen et al., 2022) that consists of four transformer layers with cross-attention. Two lightweight decoders are used to reconstruct visual tokens and the motion information of masked patches. Specifically, the appearance decoder is composed of four transformer layers based on factorised self-attention, just like the encoder, while the motion decoder is composed of two transformer layers. Additionally, the positional embeddings of the encoder are multiplexed onto the regressor and the decoders.

Implementation Details. In this paper, we evaluate our method on four widely used video datasets, *i.e.*, Kinetics-400, Something-Something V2, UCF-101 and HMDB-51. *The statistics of these datasets are provided in Appendix.* We conduct all experiments on Something-SomethingV2

appearance	motion	acc.
2	2	90.54
4	4	90.97
4	2	**91.09**

(a) **Depth of appearance decoder and motion decoder**. A shallower motion decoder compared with appearance decoder works better for a easier reconstruction target.

stride	acc.
2	90.86
4	**91.09**
8	90.27

(b) **Pretraining sampling stride**. The motion reconstruction target is more difficult for the larger sampling stride, and sampling stride of 4 works the best.

dim	acc.
384	90.08
512	90.35
768	**91.09**

(c) **Regressor and decoder width**. Matched regressor and decoder width with the encoder works the best in our framework.

case	ratio	acc.
cube	40%	89.87
tube	60%	90.26
tube	75%	**91.09**
tube	90%	87.76

(d) **Mask strategy**. 75% mask ratio for tube masking works the best.

target	acc.
VQGAN-8192	90.88
VQGAN-16384	**91.09**
DALL-E	90.61

(e) **Appearance reconstruction objective**. Results of using different tokenizers to form the appearance prediction targets.

target	acc.
optical flow	91.05
RGB-diff	**91.09**
clip-order	90.12

(f) **Motion reconstruction objective**. Results of using different motion target as in Sec. 2.2.

Table 1: **Ablation experiments** of model design on UCF101. We uniformly pretrain 800 epochs and finetune 100 epochs. The encoder is ViT-B and the input video with the size of $16\times224\times224$ is embeded to patch tokens with the size of $1\times16\times16$. The default settings are marked in gray.

and Kinetics-400 datasets with 32 NVIDIA A100-40GB GPUs. As for the smaller UCF101 and HMDB51 datasets, we use 8 NVIDIA A100-40GB GPUs. We follow BEVT (Wang et al., 2022b) to use the off-the-shelf visual tokenizer VQGAN (Esser et al., 2021) to generate semantic tokens, and employ the frame-level RGB difference as the ground-truth of motion prediction for masked video patches. *More implementation details are available in Appendix.*

4.2 Ablation Studies

We follow the paradigm of self-supervised pre-training and then fine-tuning in previous work (Feichtenhofer et al., 2022a; Tong et al., 2022), and conduct ablation studies to analyze the details of our design choices. Then, we investigate the effectiveness of different components of our framework.

Decoder Depth. Table 1a shows the performance of different transformer block number in decoders. Intuitively, the more difficult reconstruction target needs a deeper decoder with better modeling capability and vice versa. The training objective for reconstructing RGB difference is easier than reconstructing the semantic tokens. The result also verifies that appearance decoder with 4 blocks and motion decoder with 2 blocks works better than appearance and motion decoder both with 4 blocks (91.09% v.s. 90.97%) or 2 blocks (91.09% v.s. 90.54%).

Masking strategy. We start with the cube masking mechanism used by (Bao et al., 2021; Tan et al., 2021; Wang et al., 2022b) which a block-wise sampling mask is expanded along the temporal axis. Then, we study the temporal tube masking mechanism in VideoMAE (Tong et al., 2022) where all frames share the same random spatial mask as MAE does (He et al., 2022).

Interestingly, we observe that random tube masking mechanism works better than block-wise masking mechanism (89.87% *v.s.* 91.09%) in our framework. This is inconsistent with a series of token prediction methods in image and video domain, such as (Chen et al., 2022), (Bao et al., 2021) and (Wang et al., 2022b). We also increase the masking ration from 60% to 90% based on tube masking mechanism and report the result in Table 1d. The tube masking with 75% ratio works the best (91.09%). A higher marking ratio such as 90% performs poor (87.76%). A possible reason is that visual token prediction is more difficult than pixel reconstruction and more context is needed for predicting masked discrete tokens, such that the task can be completed with meaningful supervision signals for the encoder.

case	1 decoder 2 tasks	decoupled decoders		regressor	alignment	acc.
		semantic	motion			
MAM^2-c	✓	✗	✗	✓	✓	89.82
w/o appearance	✗	✗	✓	✓	✓	87.56
w/o motion	✗	✓	✗	✓	✓	90.03
w/o regressor	✗	✓	✓	✗	✓	90.84
w/o alignment	✗	✓	✓	✓	✗	90.63
MAM^2	✗	✓	✓	✓	✓	**91.09**

Table 2: **Empirical study on impact of each main components of our MAM^2 framework on UCF101 dataset.** The default settings are marked in gray.

Temporal sampling stride. For the branch of our motion decoder, the temporal sampling stride of video frames reflects the prediction difficulty when making predictions for the masked video patches. We provide the results of different temporal sampling stride in Table 1b. Stride of 2 and 8 both obtain slightly worse results (90.86% and 90.27%). Stride of 4 achieves the trade-off between the pretext task difficulty and the fine-tuning accuracy (91.09%).

Regresor and Decoder width. Table 1c report the result of different regresor and docoders depth. Note that, the depth of regressor, semantic decoder and the modiotn decoder are the same. In particular, we decrease the width from 512 to 384. We add a projection layer after the encoder in order to project the latent representation to the width of regressor for the case of 512 or 384. And the width of 768 achieve the best results of 91.09%.

Appearance prediction objective. We ablate different prediction targets for the appearance decoder and show results in Table 1e. We use the visual tokenizer of VQGAN (Esser et al., 2021) that pretrained on OpenImages with GumbelQuantization. The VQGAN tokenizer with a vocabulary size of 16384 achieves the best performance (91.09%). Using the VQGAN-8192 tokenizer results in a slight drop of 0.21%. A discrete variational autoencoder (dVAE) in (Ramesh et al., 2021) performs much worse than VQGAN (90.61%). Please refer to appendix for tokenizer details.

Motion prediction objective. We further conduct studies to evaluate different motion targets as introduced in Sec. 2.2 and results are summarized in Table 1f. The optical flow achieves the similar performance with RGB difference, however it introduces extra computational cost due to RAFT. Compared to optical flow, RGB-diff is a practical and efficient target for capturing the short-term motion cues. On the other hand, clip-order prediction (90.12%) providing long-term temporal evolution cues is unproductive compared to using only the appearance stream (90.03%, see "w/o motion" in Table 2). We attribute it to the position embedding added into the masked queries when querying the latent representations for masked patches by the regressor, which will collapse the clip-order prediction task, the position information implied in the latent representations can lead to leakage of original temporal information.

Contributions of the main components in MAM^2. In this subsection, we explore the contributions of the three main components in MAM^2 architecture, *i.e.*, the regressor, the disentangled decoders and the alignment constraint, and report the results on UCF101 dataset in Table 2. Firstly, we compare the performance of coupled decoder (*i.e.*, case MAM^2-c where only a single decoder with 2 prediction heads is used for the 2 pretext tasks) and disentangled decoders (case MAM^2). Results show that coupled decoder decreases the accuracy to 89.82% compared to default settings (91.09%).

Following the setting of disentangled decoders, we separately remove the the components of appearance decoder, motion decoder, regressor and the alignment constraint. It is clear that decoupling the appearance and motion decoders helps to improvement the overall performance significantly. In particular, only using the appearance decoder (w/o motion) result in a drop of 1.06% compared to default settings, and only using the motion decoder (w/o appearance) drops significantly (3.53%). Moreover, the w/o alignment achieves the result of 90.63% by removing the alignment constraint. The w/o regressor achieves the result of 90.84% by removing latent contextual regressor. These results imply that regressor together with the alignment loss is important to make the encoder focus on spatial-temporal feature excavation.

Method	Backbone	Pre-train data	Frames	Views	Param	Top-1	Top-5
Supervised pre-training							
NL I3D (Wang et al., 2018b)	ResNet101	IN-1K	128	10×3	62	77.3	93.3
TAM (Liu et al., 2021b)	ResNet152	IN-1K	16	4×3	59	79.3	94.1
TDN$_{En}$ (Wang et al., 2021b)	ResNet101$_{\times 2}$	IN-1K	8+16	10×3	88	79.4	94.4
Video Swin (Liu et al., 2021a)	Swin-B	IN-1K	32	4×3	88	80.6	94.6
TimeSformer (Bertasius et al., 2021)	ViT-B	IN-21K	8	1×3	121	78.3	93.7
TimeSformer (Bertasius et al., 2021)	ViT-L	IN-21K	96	1×3	430	80.7	94.7
ViViT FE (Arnab et al., 2021)	ViT-L	IN21K	128	1×3	N/A	81.7	93.8
Motionformer (Patrick et al., 2021)	ViT-B	IN-21K	16	10×3	109	79.7	94.2
Motionformer (Patrick et al., 2021)	ViT-L	IN-21K	32	10×3	382	80.2	94.8
Video Swin (Liu et al., 2021a)	Swin-L	IN-21K	32	4×3	197	83.1	95.9
ViViT FE (Arnab et al., 2021)	ViT-L	JFT-300M	128	1×3	N/A	83.5	94.3
ViViT (Arnab et al., 2021)	ViT-H	JFT-300M	32	4×3	N/A	84.9	95.8
ip-CSN (Tran et al., 2019)	ResNet152	K400	32	10×3	33	77.8	92.8
SlowFast (Feichtenhofer et al., 2019)	R101+NL	K400	16+64	10×3	60	79.8	93.9
MViTv1 (Fan et al., 2021)	MViTv1-B	K400	32	5×1	37	80.2	94.4
Self-supervised pre-training							
VIMPAC (Tan et al., 2021)	ViT-L	HowTo100M+DALLE	10	10×3	307	77.4	N/A
BEVT (Wang et al., 2022b)	Swin-B	IN-1K+K400+DALLE	32	4×3	88	80.6	N/A
MaskFeat (Wei et al., 2022)	MViT-L	K400	16	10×1	218	84.3	96.3
VideoMAE (Tong et al., 2022)	ViT-B	K400	16	5×3	87	80.9	94.7
VideoMAE (Tong et al., 2022)	ViT-L	K400	16	5×3	305	84.7	96.5
OmniMAE (Girdhar et al., 2022)	ViT-B	IN-1K+K400	16	5×3	87	80.6	N/A
OmniMAE (Girdhar et al., 2022)	ViT-L	IN-1K+K400	16	5×3	305	84.0	N/A
ST-MAE(Feichtenhofer et al., 2022a)	ViT-B	K400	16	7×3	87	81.3	94.9
ST-MAE(Feichtenhofer et al., 2022a)	ViT-L	K400	16	7×3	304	84.8	96.2
MAM^2 (Ours)	ViT-B	K400+VQGAN	16	7×3	114	**82.3**	**95.3**
MAM^2 (Ours, 600epoch)	ViT-L	K400+VQGAN	16	7×3	403	**84.8**	**96.4**
MAM^2 (Ours, 800epoch)	ViT-L	K400+VQGAN	16	7×3	403	**85.1**	**96.7**

Table 3: **Comparison with the state-of-the-art methods on Kinetics-400**. MAM^2 is pre-trained 800 epochs for ViT-B and ViT-L. "N/A" indicates the numbers are not available for us.

Method	Backbone	Pre-train data	Frames	Views	Param	Top-1	Top-5
Supervised pre-training							
TSM$_{En}$ (Lin et al., 2019)	ResNet50$_{\times 2}$	IN-1K	16+16	2×3	49	66.0	90.5
TAM (Liu et al., 2021b)	ResNet50$_{\times 2}$	IN-1K	8+16	2×3	51	66.0	90.1
TDN$_{En}$ (Wang et al., 2021b)	ResNet101$_{\times 2}$	IN-1K	8+16	1×3	88	69.6	92.2
SlowFast (Feichtenhofer et al., 2019)	ResNet101	Kinetics-400	8+32	1×3	53	63.1	87.6
MViTv1 (Fan et al., 2021)	MViTv1-B	Kinetics-400	64	1×3	37	67.7	90.9
TimeSformer (Bertasius et al., 2021)	ViT-B	IN-21K	8	1×3	121	59.5	N/A
TimeSformer (Bertasius et al., 2021)	ViT-L	IN-21K	64	1×3	430	62.4	N/A
ViViT FE (Arnab et al., 2021)	ViT-L	IN-21K+K400	32	4×3	N/A	65.9	89.9
Motionformer (Patrick et al., 2021)	ViT-B	IN-21K+K400	16	1×3	109	66.5	90.1
Motionformer (Patrick et al., 2021)	ViT-L	IN-21K+K400	32	1×3	382	68.1	91.2
Video Swin (Liu et al., 2021a)	Swin-B	IN-21K+K400	32	1×3	88	69.6	92.7
Self-supervised pre-training							
VIMPAC (Tan et al., 2021)	ViT-L	HowTo100M	10	10×3	307	68.1	N/A
BEVT (Wang et al., 2022b)	Swin-B	IN-1K+K400+DALL-E	32	1×3	88	70.6	N/A
OmniMAE(Girdhar et al., 2022)	ViT-B	IN-1K+SSv2	16	2×3	87	69.5	N/A
VideoMAE (Tong et al., 2022)	ViT-B	SSv2	16	2×3	87	70.6	92.7
MAM^2 (Ours)	ViT-B	SSv2+VQGAN	16	2×3	114	**71.3**	**93.1**

Table 4: **Comparison with the state-of-the-art methods on Something-Something V2.** MAM^2 is pre-trained 1200 epochs for ViT-B. "N/A" indicates the numbers are not available for us.

4.3 Main Results and Analysis

Comparisons with State-of-the-art methods. We report the result on Kinetics-400 and Something-Something V2 datasets in Table 3 and Table 4. Compare with the previous state-of-the-art methods, firstly, our MAM^2 achieves a gain of around +1.0% compared with the SOTA results of ST-MAE on

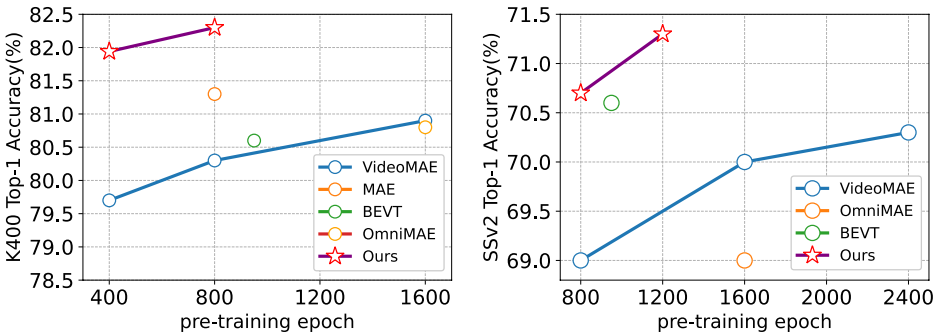

Figure 2: Top-1 accuracy on Kinetics-400 and Something-Something V2 with different masked video modeling methods. The default architecture is ViT-B. (*Best viewed in color.*)

Method	Backbone	Pre-train data	UCF101	HMDB51
CoCLR (Han et al., 2020)	S3D-G	UCF101	81.4	52.1
Vi^{2}CLR (Diba et al., 2021)	S3D	UCF101	82.8	52.9
MCN (Lin et al., 2021)	R3D	UCF101/HMDB51	85.4	54.8
VideoMAE (Tong et al., 2022)	ViT-B	UCF101/HMDB51	90.8	61.1
MAM^2 (Ours)	ViT-B	UCF101/HMDB51	**91.5**	**62.5**

Table 5: **Comparisons to Previous Methods on UCF101 and HMDB51.** MAM^2 is pre-trained 1200 epochs on UCF101 and 1600 epochs on HMDB51.

Kinetics-400 with the ViT-B architecture. On the temporal-heavy dataset of Something-Something V2, MAM^2 (ViT-B) can obtain a better performance of 71.3% compared to the VideoMAE (70.6%). Secondly, MAM^2 respectively achieves the pre-training with 800 and 1200 epochs on Kinetics-400 and Something-Something V2 that is 2$\times$ fewer than the VideoMAE, as shown in Fig.2. Thirdly, we report the results on two other small datasets (UCF101 and HMDB51), and compared with the previous methods. Table 5 shows that MAM^2 outperforms VideoMAE by 0.7% with 2.7$\times$ fewer pre-training epochs (1200 v.s. 3200), and obatain the gain of 1.4% with 3$\times$ fewer pre-training epochs (1600 v.s. 4800). At last, we observe that our MAM^2 with ViT-L also outperforms ST-MAE and VideoMAE when they are all pre-trained for 800 epochs, we can also achieve comparable results with ST-MAE or VideoMAE with only 600 epochs.

5 Limitation

Our Masked Appearance-Motion Modeling (MAM^2) framework excites the encoder efficiently exploring spatiotemporal information in video data and significantly improved performance on downstream video recognition tasks. In this work, subjected to GPU resources, we do not explore how our MAM^2 would behave under the circumstances of larger pre-training batch size or larger pre-training epochs. On the other hand, our pre-training framework introduces an extra tokenizer, which also impacts the overall performance a lot, besides, the on-the-fly computation of the tokenizer further limits our pre-training batch size. The recent work (Bai et al., 2022) inspires us to revisit the appearance stream in a masked representation modeling manner to deprecate tokenizer meanwhile providing reciprocal information with motion targets, we leave this as our future work.

6 Conclusion

In this paper, we make the first attempt to investigate motion cues explicitly in the Masked Video Modeling (MVM) pre-training scheme. With our MAM^2 framework, disentangled appearance and motion decoders are dedicated to reconstruct visual and motion knowledge respectively, meanwhile the encoder is forced to learn robust and generalized spatiotemporal video representations. The disentangled decoders help to reduce the convergence of pre-training. Experimental results show that our method achieves competitive results on all video action recognition datasets with even a half number of pre-training epochs of state-of-the-art MVM methods.

REFERENCES

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

Yutong Bai, Haoqi Fan, Ishan Misra, Ganesh Venkatesh, Yongyi Lu, Yuyin Zhou, Qihang Yu, Vikas Chandra, and Alan Yuille. Can temporal information help with contrastive self-supervised learning? *arXiv preprint arXiv:2011.13046*, 2020.

Yutong Bai, Zeyu Wang, Junfei Xiao, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. Masked autoencoders enable efficient knowledge distillers. *arXiv preprint arXiv:2208.12256*, 2022.

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

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021.

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

Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pp. 1691–1703. PMLR, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020b.

Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Ali Diba, Vivek Sharma, Reza Safdari, Dariush Lotfi, Saquib Sarfraz, Rainer Stiefelhagen, and Luc Van Gool. Vi2clr: Video and image for visual contrastive learning of representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1502–1512, 2021.

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

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

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

B Fang, W Wu, C Liu, Y Zhou, D He, and W Wang. Mamico: Macro-to-micro semantic correspondence for self-supervised video representation learning. *Proc. ACMMM*, 2022.

Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 203–213, 2020.

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

Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3299–3309, 2021.

Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022a.

Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022b.

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

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

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *arXiv preprint arXiv:2010.09709*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

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

Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8129–8138, 2020.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Haofei Kuang, Yi Zhu, Zhi Zhang, Xinyu Li, Joseph Tighe, Sören Schwertfeger, Cyrill Stachniss, and Mu Li. Video contrastive learning with global context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3195–3204, 2021.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pp. 2556–2563. IEEE, 2011.

Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7083–7093, 2019.

Yuanze Lin, Xun Guo, and Yan Lu. Self-supervised video representation learning with meta-contrastive network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8239–8249, 2021.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021a.

Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13708–13718, 2021b.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

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

Kanchana Ranasinghe, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Michael S Ryoo. Self-supervised video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2874–2884, 2022.

Adria Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Pătrăucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1255–1265, 2021.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

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

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

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

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Jinpeng Wang, Yuting Gao, Ke Li, Jianguo Hu, Xinyang Jiang, Xiaowei Guo, Rongrong Ji, and Xing Sun. Enhancing unsupervised video representation learning by decoupling the scene and the motion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10129–10137, 2021a.

Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14010–14020, 2022a.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018a.

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

Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14733–14743, 2022b.

Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang. Self-supervised learning for semi-supervised temporal action proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1905–1914, 2021c.

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

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3024–3033, 2021d.

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

Wenhao Wu, Dongliang He, Tianwei Lin, Fu Li, Chuang Gan, and Errui Ding. Mvfnet: Multi-view fusion network for efficient video recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2943–2951, 2021.

Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16684–16693, 2021.

Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10334–10343, 2019.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. VLM: Task-agnostic video-language model pre-training for video understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4227–4239, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.370. URL https://aclanthology.org/2021.findings-acl.370.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, November 2021b. Association for Computational Linguistics.

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

Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10656–10664, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

	Kinetics-400	Something-Something V2	UCF101	HMDB51
Training Split	240k	169k	9.5k	3.5k
Validation Split	20k	20k	3.5k	1.5k
Number of Classes	400	174	101	51
Average Video Duration	10s	4s	7s	4s

Table 6: **The statistics for our used video recognition datasets.**

A DATASETS

We evaluate our MAM^2 on four common video recognition datasets: Kinetics-400 (Carreira & Zisserman, 2017), Something-Something V2 (Goyal et al., 2017), UCF101 (Soomro et al., 2012) and HMDB51 (Kuehne et al., 2011). In Table Table 6, we report the key statistics of these four datasets. We demonstrate the effectiveness of our MAM^2 on both Spatially-heavy datasets such as Kinetics-400, UCF101 and HMDB51, and Temporally-heavy dataset, e.g. Something-Something V2.

B IMPLEMENTATION DETAILS

In the pre-training phase, we consistently use the dense sampling strategy in (Feichtenhofer et al., 2019) with stride 4. Moreover, we perform linearly scaling strategy for the base learning rate w.r.t $lr = base\ learning\ rate \times batch\ size/256$. During the pre-training phase, we adopt the repeated augmentation trick (Hoffer et al., 2020) for a fair comparison with (Tong et al., 2022; Feichtenhofer et al., 2022b). We uniformly sample 16 frames during the pre-training and finetuning phases for all datasets. Moreover, we use frames extracted offline from the raw videos during pre-traing phase to reduce the data loading time, and we load the raw videos on the fly during fine-tuning phase.

Kinetics-400. Our settings follow (Feichtenhofer et al., 2022b; Tong et al., 2022). We trained MAM^2 for 800 epochs in the pre-training phase. During the fine-tuning phase, we use the dense sampling with stride 4. For a fair comparison, we perform the inference protocol with 7 clips x 3crops following (Feichtenhofer et al., 2022b). Table 7 shows the pre-training settings for the architecture ViT/B and ViT/L and Table 8 shows the fine-tuning details.

Something-Something V2. We trained MAM^2 based on ViT-B for 1200 epochs in the pre-training by default. During the fine-tuning phase, we perform the uniform sampling strategy with stride 2 following TSN (Wang et al., 2018a) and we adopt the inference protocol with 2 clips x 3crops for comparison with (Tong et al., 2022). Default settings for ViT-B in pre-training and fine-tuning are correspondingly shown in Table 7 and Table 8. Notably, $flip\ augmentation$ is deprecated during both the pre-training and fine-tuning phase.

UCF101. Our MAM^2 is pre-trained for 800 epochs by default in ablation study and we report the final result with 1200 epochs in Table 5. The $base\ learning\ rate$ is set to $1.5e$-4 during the pre-traing and is set to $1e$-4 during the fine-tuning. The batch size in single node is set to 32 for pre-training and fine-tuning. We adopt dense sampling with temporal stride of 4 for pre-traing and fine-tuning. Default settings in pre-training and fine-tuning are correspondingly shown in Table 7 and Table 8.

HMDB51. Our MAM^2 is pre-trained for 1600 epochs by default and we report the final result in Table Table 5. The $base\ learning\ rate$ is set to $1.5e$-4 during the pre-traing and is set to $1e$-4 during the fine-tuning. The batch size in single node is set to 32 for pre-training and fine-tuning. We adopt dense sampling with temporal stride of 4 for pre-training and fine-tuning. Default settings for ViT-B in pre-training and fine-tuning are correspondingly shown in Table 7 and Table 8.

C ARCHITECTURE DETAILS

We complement more architecture details of MAM^2 following Sec. 2.1. We design an encoder-regressor-decoder architecture with dual stream decoders for self-supervised video transformer pre-

config	Kinetics-400	Sth-Sth V2	UCF101	HMDB51
optimizer	AdamW (Loshchilov & Hutter, 2017)			
base learning rate	1.5e-4			
weight decay	0.05			
optimizer momentum	β_1, β_2=0.9, 0.95 (Chen et al., 2020a)			
batch size	128	128	32	32
learning rate schedule	cosine decay (Loshchilov & Hutter, 2016)			
pre-training epochs	800	1200	1200	1600
warmup epochs	20	30	30	40
flip augmentation	*yes*	*no*	*yes*	*yes*
augmentation	MultiScaleCrop			

Table 7: **Pre-training setting.**

config	Kinetics-400	Sth-Sth V2	UCF101	HMDB51
optimizer	AdamW (Loshchilov & Hutter, 2017)			
base learning rate	5e-4(B) 1e-3(L)	5e-4	1e-4	1e-4
weight decay	0.05			
optimizer momentum	β_1, β_2=0.9, 0.999 (Chen et al., 2020a)			
layer-wise lr decay	0.75 (Chen et al., 2020a)			
batch size	256(B) 64(L)	256	24	24
learning rate schedule	cosine decay (Loshchilov & Hutter, 2016)			
warmup epochs	5			
training epochs	100(B) 35(L)	35	100	60
flip augmentation	*yes*	*no*	*yes*	*yes*
RandAug	(9, 0.5) (Cubuk et al., 2020)			
label smoothing	0.1 (Szegedy et al., 2016)			
mixup	0.8 (Zhang et al., 2017)			
cutmix	1.0 (Yun et al., 2019)			
drop path	0.1 (Huang et al., 2016)			
repeated sampling	0.1 (Hoffer et al., 2020)			

Table 8: **Fine-tuning setting.**

training. Table 9 takes the 16-frame ViT-Base for example and displays the pipeline details during the pre-training phase.

D TOKENIZER

For the appearance prediction, we use the discrete variational autoencoder (dVAE) in VQGAN (Esser et al., 2021) to generate the visual tokens as the pre-training target. The dVAE acts as a visual tokenizer which transforms each video frame with the size of 224×224 into 14×14 visual tokens according to a pre-trained codebook where the vocabulary size of each visual token is 16384. Note that we deprecate the decoder of the dVAE following (Bao et al., 2021; Wang et al., 2022b).

E MASK SAMPLING

Fig.3 shows the *cube* and *tube* masking strategies. In particular, *cube* mask is repeated from a 2-D block-wise mask on temporal dimension. And *tube* mask is repeated from a 2-D random-wise mask on temporal dimension.

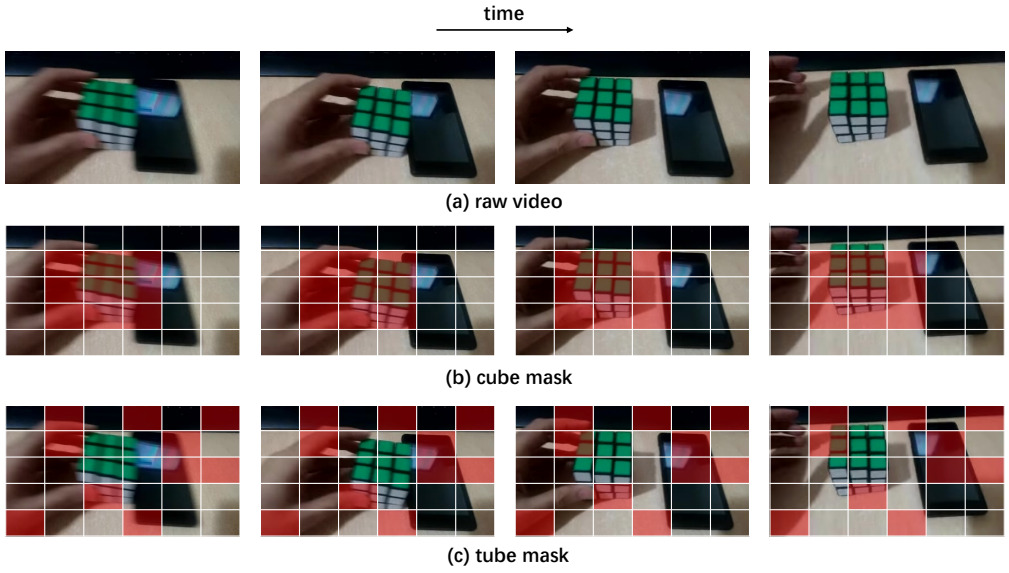

Figure 3: Demonstration of different masking strategies. (a) is the raw input video. (b) is *cube* masking strategy and (c) is *tube* masking strategy. The masked patches are marked in red.

stage	**component**	**output sizes**
data	*temporal stride* 4	3×16×224×224
patch embedding	*patch size* 1×16×16	768×16×196
masking	tube mask *mask ratio* = 0.75	768×16×[196×(1-0.75)]
encoder	[MHA-T(768) MHA-S(768) MLP(3072)] ×12	768×16×[196×(1-0.75)]
reshape	*from* 768×16×(196×0.25) *to* 768×(16×196×0.25)	768×(16×196×0.25)
regressor	[Cross-attn(768) MLP(3072)] ×4	768×16×[196×(0.75)]
reshape	*from* 768×(16×196×0.25) *to* 768×16×(196×0.25)	768×16×(196×0.25)
appearance-decoder	[MHA-T(768) MHA-S(768) MLP(3072)] ×4	768×16×[196×(0.75)]
appearance-projector	Linear(16384)	16384×16×[196×(0.75)]
motion-decoder	[MHA-T(768) MHA-S(768) MLP(3072)] ×2	768×16×[196×(0.75)]
motion-projector	Linear(768)	768×16×[196×(0.75)]

Table 9: **Architectures details during the pre-training of our MAM^2.** "MHA-T" and "MHA-S" correspondingly denote the temporal and spatial multi-head self-attention. "Cross-attn" denotes cross attention layer in a space-time agnostic manner. In particular, "Cross-attn" takes the latent representation of visible patches with a shape of $[768 \times (16 \times 196 \times 0.25)]$ as the key and value representations and takes the mask queries witch a shape of $[768 \times (16 \times 196 \times 0.75)]$ as the query representations. We use masking ratio