MOFO: MOtion FOcused Self-Supervision for Video Understanding

Mona Ahmadian, Frank Guerin, Andrew Gilbert
University of Surrey, UK
{m.ahmadian, f.guerin, a.gilbert}@surrey.ac.uk

Appendix

We also conducted various ablation studies to examine the design choices made in our proposed strategy.

A Motion-focused Self-supervised Learning

Experimental setting. We use the technique of **?** to extract optical flow from a video to create a motion map, which is 40% faster by parallelizing IO and computation.
MOFO uses ViT-Base as a decoder/encoder backbone, trained for 800 epochs on Something-Something V2 and Epic-Kitchens for the SSL independently. We follow the training and experiential parameters from recent work **?** to ensure a fair comparison and finetune for 100 epochs with early stopping. The model takes 16 frames from the video with 224×224 size and divides the input video into a 3D $16 \times 16 \times 8$ patch embeddings, resulting in $H = 224$, $W = 224$, $T = 16$, $H_t = 16$, $W_t = 16$, $T_t = 8$, and $N = 392$. While we have a fixed number of input patches for our model, we do not have a fixed number of inner N_{inner} and outer N_{outer} embeddings due to varying size of the motion area in each video clip. We report Top-1 accuracy on Epic-Kitchens and Top-1 and Top-5 accuracy on Something-Something V2 on downstream tasks and use Pytorch and DeepSpeed **?** on 4xNVIDIA Quadro RTX-5000 GPU for our experiments.

Masking ratio. VideoMAE **?** recommended tube masking with an extremely high ratio which helps reduce information leakage during masked modelling. They demonstrated the best efficiency and efficacy with a masking ratio of 90%. Therefore, we explore the effect of the inside masking ratio for verb classification on Epic-Kitchens in Fig. 1. It shows that the model pretrained with a masking ratio of 90% as the general masking ratio for a video and a high ratio for inside masking ratio (75%) achieves the highest efficiency level. Thus, we continue experimenting with the rest by fixing the inside mask ratio to 75%.

Reconstructed frames This section shows several reconstructed image frames from a video in Fig. 2 and Fig. 3. We use an asymmetric encoder-decoder architecture to accomplish video self-supervised pretraining tube masking with a high ratio for MAE pretraining. We can reconstruct the masked patches using random tube masking by finding the spatially and temporally corresponding unmasked patches in the adjacent frames. The loss function is the mean squared error (MSE) loss between normalised masked tokens and reconstructed tokens in pixel space. Videos are all randomly chosen from the validation sets of both datasets. Our proposed MOFO model ensures that a fixed number of masks exist within the motion area compared to the VideoMAE model. These examples suggest that, compared to VideoMAE, our MOFO model reconstructs the samples in the motion area significantly more accurately, demonstrating that the model has focused on the motion area. We can produce satisfying reconstruction results, mainly when motion occurs with our MOFO, by applying

37th Conference on Neural Information Processing Systems (NeurIPS 2023). Workshop: Self-Supervised Learning-TheoryandPractice.

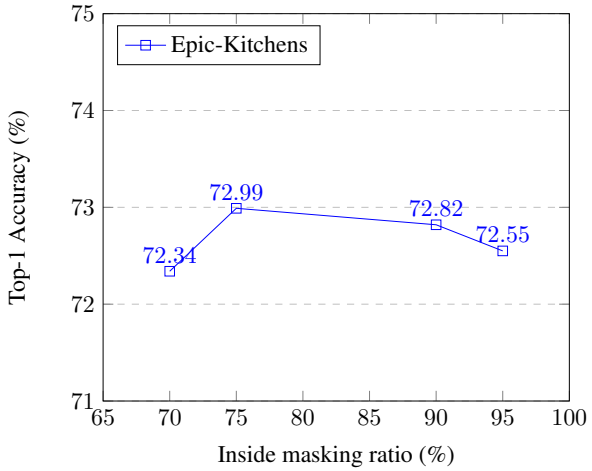


Figure 1: The effect of inside masking ratio on Epic-Kitchens-100 dataset for verb classification demonstrates that a high inside masking ratio (75%) delivers the best efficiency and effectiveness trade-off.

extremely high ratio masking at random (90%) while always masking a fixed percentage of the tubes (75%) inside the motion area.

B Motion-focused Finetuning

Setup details Given a set of patches $\{\mathbf{p}_i\}_1^N$, the transformer yields two sets of embeddings: $\{\mathbf{e}^{\text{inner}}\}_{j=1}^{N_{\text{inner}}}$ for the inner motion boxes and $\{\mathbf{e}^{\text{outer}}\}_{k=1}^{N_{\text{outer}}}$ for the outer ones, as described by:

$$\{\mathbf{e}^{\text{inner}}\}_{j=1}^{N_{\text{inner}}}, \{\mathbf{e}^{\text{outer}}\}_{k=1}^{N_{\text{outer}}} = \text{ViT}\big(\{\mathbf{p}_i\}_1^N\big) \tag{1}$$

These embeddings are then processed by a cross-attention mechanism, where Q, K, and V represent query, key, and value, respectively. The CrossAttention function is formalised as follows:

$$\text{CrossAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where $Q = \mathbf{e}^{\text{inner}}$, $K = V = \mathbf{e}^{\text{outer}}$. In the context of multi-head attention, each attention head i is computed by applying the CrossAttention function to the query, key, and value matrices, each weighted by a different learned weight matrix $W_i^Q \in \mathbb{R}^{d_{model} \times d_q}, W_i^k \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ respectively:

$$\text{head}_i = \text{CrossAttention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

Finally, the fused embedding $\mathbf{e}^{\text{fused}}$ is computed by concatenating the results from all attention heads and then applying another learned weight matrix $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. This multi-head cross-attention (MCA) operation can be represented as:

$$\mathbf{e}^{\text{fused}} = \text{MCA}(Q, K, V) = \text{Concat}\left(\text{head}_1, \cdots, \text{head}_h\right) W^O \tag{4}$$

We employ $h = 3$ parallel attention layers, or heads, in this work. We also use $d_q = d_k = d_v = d_{model}$ for each. The model is ultimately finetuned with a cross-entropy loss $\mathcal{L}$:

$$\begin{aligned} \mathcal{L} &= -\sum_n \mathbf{y}_n \log \hat{\mathbf{y}}_n \\ \hat{\mathbf{y}} &= \text{FC}(\mathbf{e}^{\text{fused}}) \end{aligned} \tag{5}$$

where, $\mathbf{y}_n$ is the true label for nth video clip, $\hat{\mathbf{y}}_n$ is its predicted label, and FC is the fully connected layers typically used for classification.

MCA hyper-parameters ablation. We list the MCA hyperparameters used in our MOFO finetuning experiments here. We experiment with various head and depth settings when Epic-Kitchens

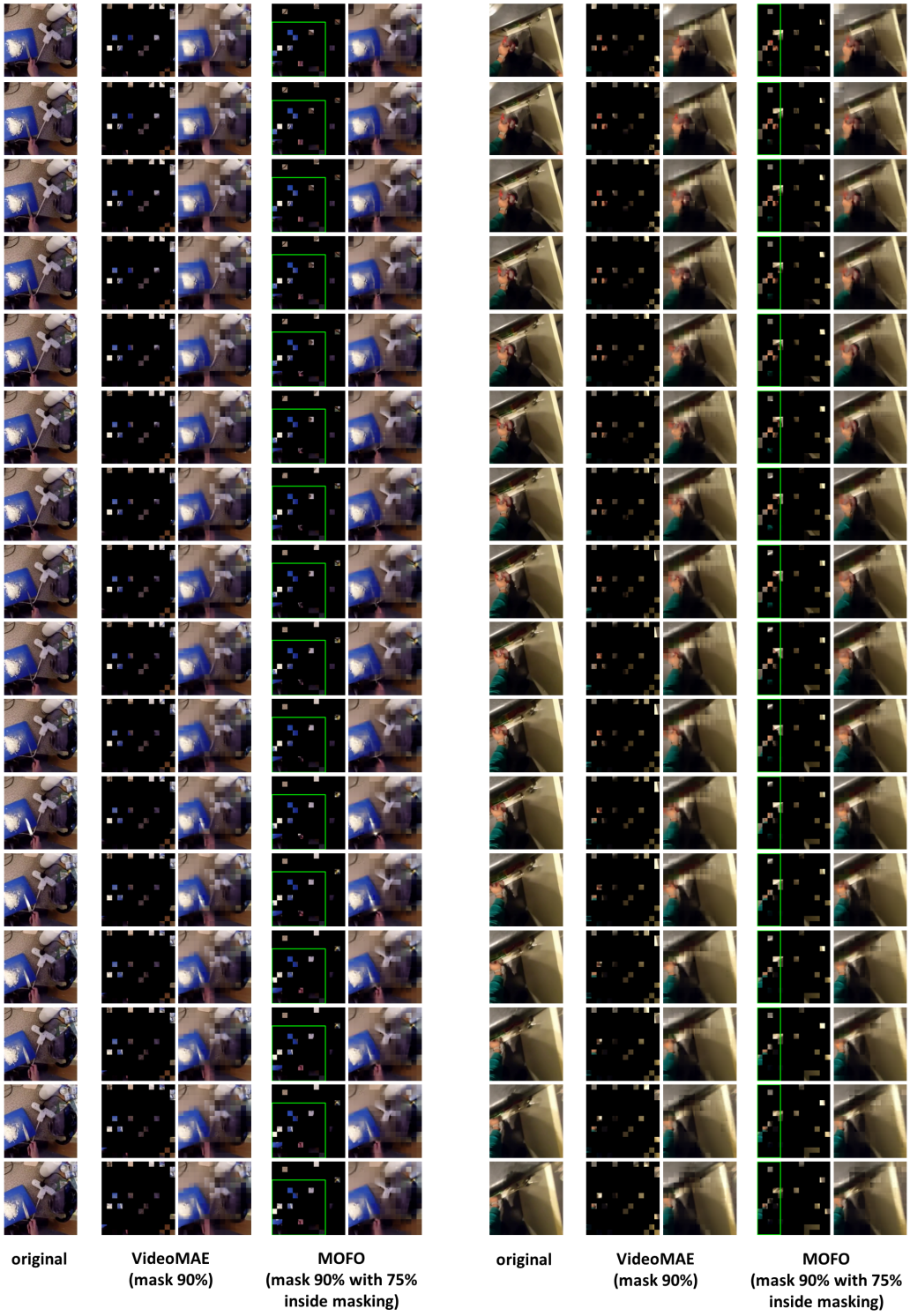


Figure 2: Qualitative Comparison on reconstructions using VideoMAE and MOFO on **Epic-Kitchens** dataset. MOFO Reconstructions of videos are predicted by MOFO pre-trained with a masking ratio of 90% and an inside masking ratio of 75% .

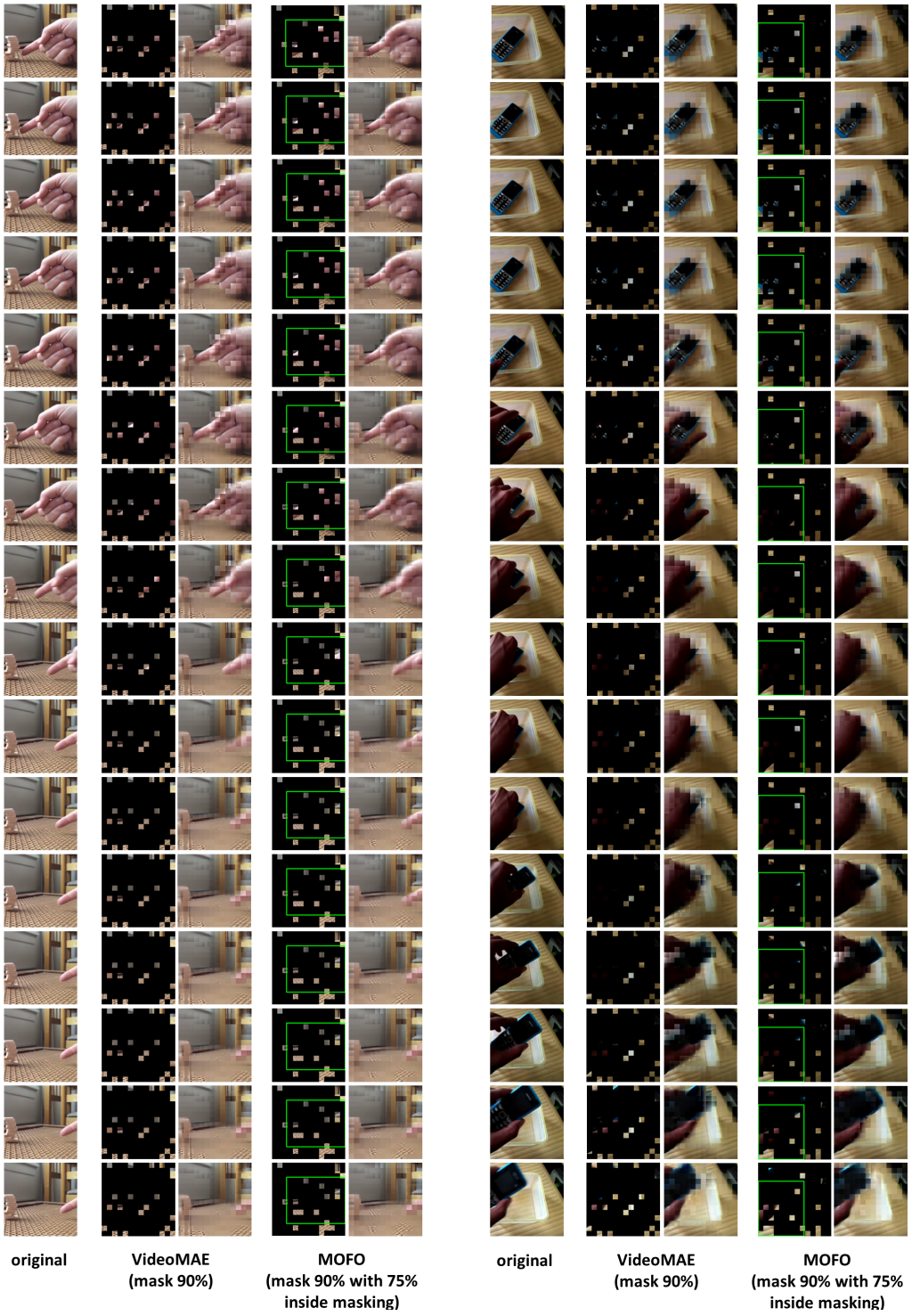


Figure 3: Qualitative Comparison on reconstructions using VideoMAE and MOFO on **Something-Something V2** dataset. MOFO Reconstructions of videos are predicted by MOFO pre-trained with a masking ratio of 90% and an inside masking ratio of 75%.

is the target dataset shown in Table 1. We experiment with these parameters for the verb task on Epic-Kitchens to find the best choice for the cross-attention layer we suggested for MOFO finetuning. The final head and depth are 3 and 1, respectively.

Table 1: Ablation experiment for number of head and depth in MOFO finetuning

Finetuning method	Backbone training	CA heads	CA depths	**Epic-Kitchens** Verb Top-1
VideoMAE	VideoMAE	-	-	71.6
MOFO	VideoMAE	1	1	73.5
MOFO	VideoMAE	1	2	73.8
MOFO	VideoMAE	1	3	73.6
MOFO	VideoMAE	2	1	73.7
MOFO	VideoMAE	2	2	73.3
MOFO	VideoMAE	**3**	**1**	**74.0**
MOFO	VideoMAE	3	2	73.5
MOFO	VideoMAE	4	1	73.8
MOFO	VideoMAE	4	2	73.3

Visualisation of GradCAM using MOFO self-supervision We visualise the GradCAM and motion map in Fig. 4 for the samples in which VideoMAE can't identify the class, but our MOFO can. The attention maps show how effective our approach is in capturing the motion area. Visualisation of important areas. The heatmap indicates how much the pretrained model attends to the region.

C Ablation Study

We finetune the learned model for action classification to evaluate the learned model as a pretrained model and train on a new downstream task with the learned representation. We perform such an evaluation on our self-supervised model to gain some insights into the generality of the learned features. For finetuning, we follow the same protocol in **?** to provide a fair comparison and call it regular finetuning. The entire feature encoder and a linear layer are finetuned end-to-end with cross-entropy loss, The recognition accuracy for our MOFO SSL using regular finetuning is reported in Table 2 shown as MOFO*. We demonstrate significant performance improvement over the other self-supervised approaches, comparable to the best-supervised approach. All variants of our model are presented in section A outperformed the existing result using ViT-MAE, but we found that the 75% inside masking ratio worked the best. Compared to VideoMAE **?**, our approach achieves significantly better results while the number of backbone parameters remains the same. While MOFO** indicates our result with pretraining on non-motion SSL and MOFO finetuning, which further increases accuracy, MOFO† denotes the MOFO SSL and MOFO finetuning, which we mentioned in Table **??** as MOFO(Proposed), and this provides the greatest performance over the best-performing methods on Epic-Kitchens verb, noun and action classification and on Something Something V2 action classification.

D Domain Generalization

Domain generalisation aims to build a predictor that can perform well in an unseen test domain, known as out-of-distribution generalisation. The main objective of this experiment is to learning video representations that transfer well to a novel previously unseen dataset. We take the MOFO and non-MOFO pretrained models that have already learned features from one dataset and finetune them to adapt them to a new dataset. Results in Table 3 show that our proposed MOFO model and non-MOFO pretrained model got on-par results; our MOFO pretrained model's accuracy on SSV2 is marginally higher when pretraining is done on Epic-Kitchens, and marginally worse on Epic-Kitchens when pretraining is done on SSV2. These results have inspired me to design a self-supervision task to enhance generalisation.

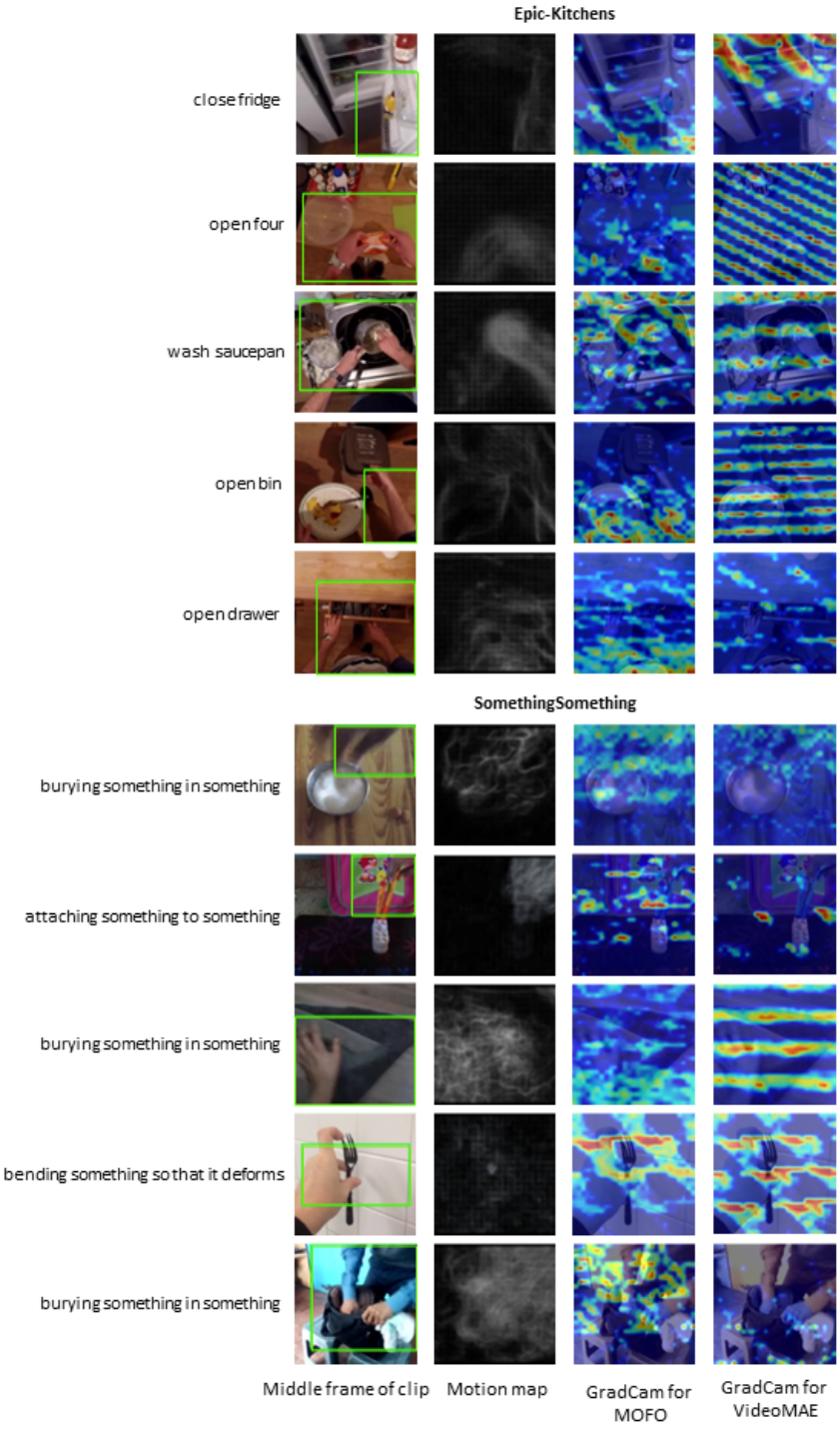


Figure 4: We visualise the attention maps generated by GradCAM based on VideoMAE and MOFO for Epic-Kitchens and the Something-Something V2 dataset. The attention maps show that our proposed approach can better capture the motion area.

Table 2: Human activity recognition on **Epic-Kitchens** and **Something-Something V2 (SSV2)** in terms of Top-1 and Top-5 accuracy. blue: This is the result computed by us using the public code MOFO* is pretrained by our MOFO SSL and uses non-MOFO finetuning. MOFO** This is our result with pretraining on non-MOFO SSL and has MOFO finetuning. MOFO† denotes the MOFO SSL and MOFO finetuning.

Method	Backbone	Param	SSV2 Action Top-1	SSV2 Action Top-5	Epic-Kitchens Verb Top-1	Epic-Kitchens Noun Top-1	Epic-Kitchens Action Top-1
Supervised							
TDN_{EN} **?**	ResNet101×2	88	69.6	92.2	-	-	-
SlowFast **?**	ResNet101	53	63.1	87.6	65.6	50.0	38.5
TSM **?**	ResNet-50	-	63.4	88.5	67.9	49.0	38.3
MViTv1 **?**	MViTv1-B	37	67.7	90.9		-	-
TimeSformer **?**	ViT-B	121	59.9	-	-	-	-
TimeSformer **?**	ViT-L	430	62.4	-	-	-	-
ViViT FE **?**	ViT-L	-	65.9	89.9	66.4	56.8	44.0
Mformer **?**	ViT-B	109	66.5	90.1	66.7	56.5	43.1
Mformer **?**	ViT-L	382	68.1	91.2	67.1	57.6	44.1
Video SWin **?**	Swin-B	88	69.6	92.7	67.8	57.0	46.1
Self-supervised							
VIMPAC **?**	ViT-L	307	68.1	-	-	-	-
BEVT **?**	Swin-B	88	70.6	-	-	-	-
VideoMAE **?**	ViT-B	87	70.8	92.4	71.6	66.0	53.2
ST-MAE **?**	ViT-L	304	72.1	-	-	-	-
OmniMAE **?**	ViT-B	87	69.5	-	-	-	39.3
Omnivore(Swin-B) **?**	ViT-B	-	71.4	93.5	69.5	61.7	49.9
Ours(MOFO*)	ViT-B	87	**72.7**	**94.2**	**73.0**	**67.1**	**54.1**
Ours(MOFO)**	ViT-B	102	**74.7**	**95.0**	**74.0**	**68.0**	**54.5**
Ours(MOFO†)	ViT-B	102	**75.5**	**95.3**	**74.2**	**68.1**	**54.5**

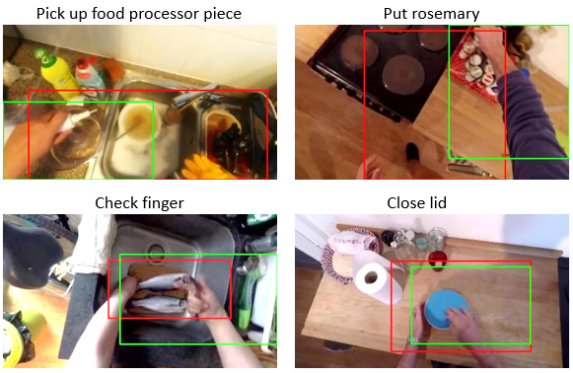
Table 3: Human activity recognition on **Epic-Kitchens** and **Something-Something V2** in terms of Top-1 accuracy. blue: This is the result computed by us using the public code MOFO* is pretrained by our MOFO SSL and uses non-MOFO (regular) finetuning.

Method	Backbone	Pretrain Dataset	Something-Something V2 Action Top-1	Epic-Kitchens Verb Top-1	Epic-Kitchens Noun Top-1	Epic-Kitchens Action Top-1
VideoMAE **?**	ViT-B	$Something - SomethingV2$	70.8	70.2	62.9	50.7
VideoMAE **?**	ViT-B	$Epic - Kitchens$	67.3	71.6	66.0	53.2
Ours(MOFO*)	ViT-B	$Something - SomethingV2$	72.7	70.0	62.7	50.6
Ours(MOFO*)	ViT-B	$Epic - Kitchens$	67.4	73.0	67.1	54.1

E Automatic Motion Area Detection

Automatic vs. supervised motion area detection. We compare the results using our automatically detected motion areas and the ground truth bounding box annotation provided by **?** on the Epic-Kitchens dataset in Table 5(b). Our automatic motion detection results are close compared to supervised annotations, as seen in Table 5(b), despite the challenging camera motion from the egocentric videos.

We compute the Intersection over the Union (IoU) metric to compare our automatic detector with the supervised annotated bounding boxes on both datasets **??**. For the Epic-Kitchens dataset, the IoU is 40%, and for Something-Something V2, the IoU is 31%. Although these numbers are lower, our automatic motion detection only detects motion and ignores unnecessary static objects near the



(a)

Method	Annotation	**Epic-Kitchens** Verb Top-1
MOFO supervision	Supervised	73.26
	Automatic(ours)	72.99

(b)

Figure 5: (a) Comparison between the unsupervised and supervised motion area detection, green rectangles indicate the unsupervised while red ones show supervised detected motion area. (b) Effect of supervised vs. automatic motion area utilisation in MOFO.

motion. As you can see in Fig. 5(a), our automatic motion box still focuses on the area and object of interest, which is the key requirement.

In Fig. 6, we present additional qualitative examples of our automatic motion area detection compared with the provided supervised annotation for Epic-Kitchens and Something-Something V2 datasets. These samples show that our proposed automatic motion area detection minimises the impact of the static object in the motion box while highlighting the motion areas. Our automatic motion box concentrates on the area and item of interest, which is necessary for our proposed approach, even for self-supervision or finetuning.

F Related Work

Self-supervised learning (SSL) is a developing machine learning technique that has the potential to address the issues brought about by over-dependence on labelled data. High-quality labelled data have been essential for many years to develop intelligent systems using machine learning techniques. Consequently, high-quality annotated data costs are a significant bottleneck in the training process. Grow the research and development of generic AI systems at an inexpensive cost. Self-learning mechanisms with unstructured data are one of the top focuses of AI researchers. Collecting and labelling a wide range of diverse data is almost impossible. Researchers are developing self-supervised learning (SSL) methods that can pick up on fine details in data to address this issue. The introduction to self-supervised learning in video understanding is followed by a review of the literature on video action recognition, the downstream task we have recently focused on.

F .1 Self-supervised video representation learning

The effectiveness of deep learning-based computer vision relies on the availability of a considerable amount of annotated data, which is time-consuming and expensive to obtain. Supervised learning is trained over a given task with a large, manually labelled dataset. In addition to the costly manual labelling, generalisation mistakes and erroneous correlations are other problems with supervised learning.

Large labelled datasets are difficult to create in particular situations, making it challenging to construct computer vision algorithms. Most computer vision applications in the real world use visual categories not included in a common benchmark dataset. In specific applications, visual categories or their appearance are dynamic and vary over time. Therefore, self-supervised learning could be created that uses a limited number of labelled examples to learn to recognise new concepts effectively. A substantial research effort focuses on learning from unlabeled data, which is much easier to acquire in real-world applications. The ultimate goal is to make it possible for machines to comprehend new concepts quickly after only viewing a few labelled instances, similar to how quickly humans can learn.

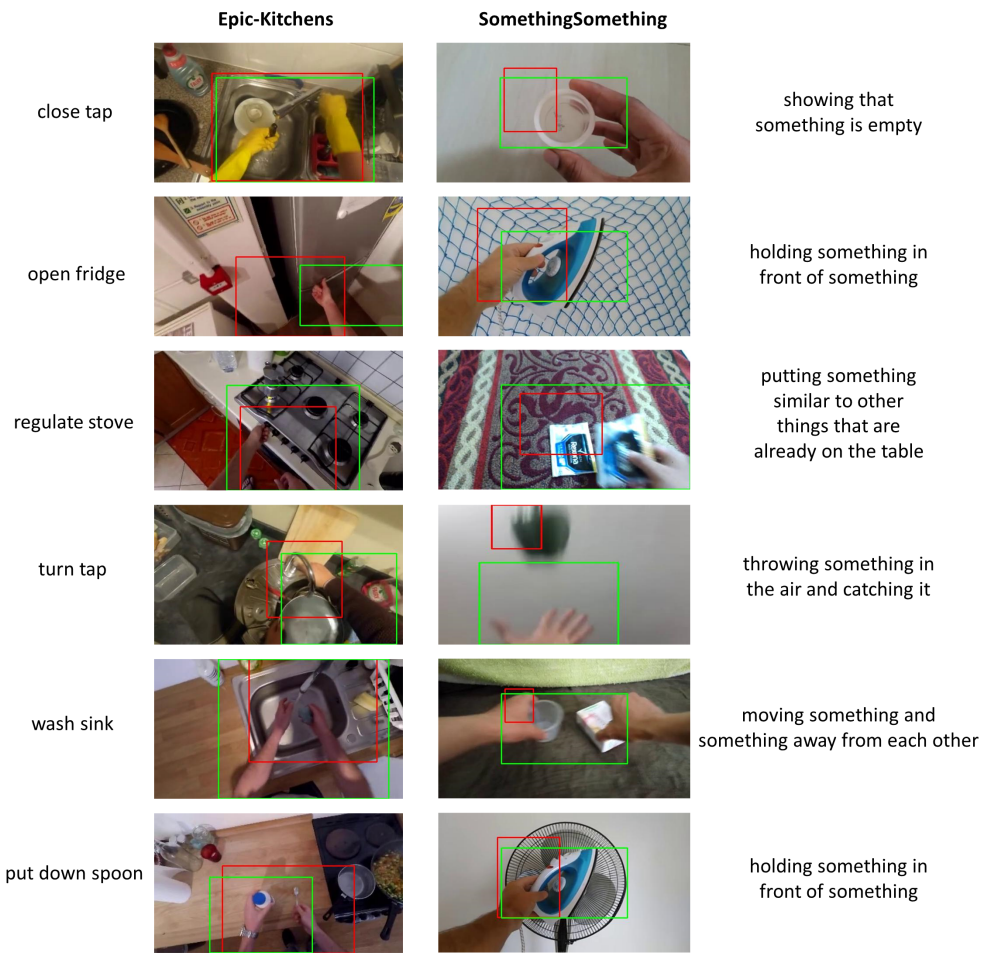


Figure 6: Comparison between the unsupervised and supervised motion area detection, green rectangles indicate the unsupervised while red ones show supervised detected motion area.

SSL has gained considerable popularity since its introduction in natural language processing **?** and computer vision **???** owing to its ability to learn effective data representations without requiring manual labels. Acquiring detailed manual labels is arguably more difficult (and often expensive) in many image and video-related tasks, which makes SSL an increasingly popular paradigm in video analysis.

The goal of video self-supervised learning for computer vision is to learn meaningful video representations without explicit supervision, and the model trains itself to learn one part of the input from another part of the input. Self-supervised learning algorithms can learn representations by solving pretext tasks that can be formulated using only unlabeled data. These auxiliary tasks can guide the model to learn intermediate representations of data. By solving these tasks, the model learns to extract relevant features from the input data and understand the underlying structural meaning beneficial for practical downstream tasks. Based on the surrogate task employed, the training objective for self-supervised learning is defined, and model parameters are updated through gradient descent to minimise prediction error. Therefore, models are trained to solve these pretext tasks. As a result, they learn to capture meaningful and useful representations that can be used for various downstream video understanding tasks, such as video action recognition F .2.

Video-based self-supervised learning techniques start from image tasks. Several specifically designed tasks, including image inpainting **?**, solving jigsaw puzzles **?**, and image colour channel prediction **?** are proposed to learn image features. SSL has recently yielded successful results in learning visual representations from unlabeled videos with various pretext tasks **???**. These methods use a backbone that has been pretrained with images or videos in a self-supervised manner to perform tasks on videos,

including contrastive learning **???**, self-distillation **?**, or Masked Modeling which selects a random section of the input sequence to mask out, and then predicts the features of those sections **????**. Many existing works **???** have been proposed to focus on temporal information, such as making models sensitive to the temporal differences of input data.

As mentioned before, earlier works build on a concept of self-supervision by taking RGB frames as input to learning to predict action concepts **?**, using Convolutional Neural Networks (CNNs) models to use frame-wise features and average pooling **?** discarding the temporal order. Thus, frame-wise CNN scores were fed to LSTMs **?** while in two-stream networks **?**, representations are computed for each RGB frame and every ten stacked optical flow frames. Spatio-temporal 3D CNN filters **????** model spatio-temporal patterns.Persistence of Appearance, a motion cue proposed by PAN **?**, allows the network to extract the motion information from adjacent RGB frames directly. Vision Transformers (ViTs) **??** have emerged as an effective alternative to traditional CNNs. The architecture of Vision Transformer is inspired by the prominent Transformer encoder **??** used in natural language processing (NLP) tasks, which process data in the form of a sequence of vectors or tokens. Like the word tokens in NLP Transformer, ViT generally divides the image into a grid of non-overlapping patches before sending them to a linear projection layer to adjust the token dimensionality. Feed-forward and multi-headed self-attention layers are then used to process these tokens. ViTs have a wide range of applications in numerous tasks due to their capacity to capture global structure through self-attention, such as classification **???**, object detection **??**, segmentation **???** and retrieval **?**.

Inspired by ViT **?**, ViViT **?** and Timesformer **?** were the first two works that successfully implemented a pure transformer architecture for video classification, improving upon the state of the art previously set by 3D CNNs. In these models, the video clip of RGB frames is embedded into 3D patches to produce downsampled feature maps. Then, these encoded 3D patches are encoded by a Video Transformer **??**. In the following work, **?** defines the tubelet embedding tokenisation method and inspired some other works to represent a video input by extracting non-overlapping, spatiotemporal tubes to propose their method **?**.

In another line of research, Masked Autoencoders (MAEs) have recently been demonstrated to be powerful yet conceptually simple and efficient and have proven an effective pretraining paradigm for Transformer models of text **?**, images **?**, and, more recently, videos **?**. The learned self-supervised model from the pretext task can be applied to any downstream computer vision tasks, including classification, segmentation, detection, etc.

Nowadays, encoder-decoder Transformer-based architectures are commonly used in self-supervised learning for video representation learning. These architectures take advantage of the Transformer models' strengths, initially created for natural language processing challenges, and adapt them to process and comprehend video data. In the context of video representation learning, the encoder-decoder Transformer architecture typically consists of the following components:

1. **Encoder** The encoder processes the input video data and generates a condensed representation of the video. Each video frame or 3D tublets is typically treated as a sequence of features to be input into the Transformer encoder. Multiple layers of self-attentional and feed-forward neural networks can be used in the encoder to capture the video's temporal dependencies, spatial relationships, and long-range dependencies.

2. **Decoder:** Based on the self-supervised task, the decoder generates a prediction using the encoder's learned representation. The decoder must solve the surrogate task used for self-supervised learning. For instance, if the self-supervised objective is to anticipate the temporal order of shuffled frames, the decoder may correctly predict that order.

In transformer-based architecture, the self-attention mechanism powers both the encoder and decoder. Self-attention architectures typically are made up of a series of transformer blocks. Each transformer block consists of two sublayers: a feed-forward layer and a multi-head self-attention layer. An input is divided into patches, and attention evaluates each 3D input patch's usefulness before drawing on it to produce the output. The Transformer's self-attention mechanism lets the model focus on different parts of the video frames while considering their dependencies. Therefore, considering their relative importance, it draws from each input component to produce the output. The query(Q), key(K), and value(V) vectors are the three sets of calculated vectors in the transformer architecture. These are determined by multiplying the input by a linear transformation.

F .2 Video action recognition

Although it is simple for humans to recognise and categorise actions in video, automating this process is challenging. Human action recognition in video is of interest for applications such as automated surveillance **?** detecting anomalies in a camera's field of view that has attracted attention from vision researchers **?**, elderly behaviour monitoring **?**, human-computer interaction, content-based video retrieval **?**, and video summarization **?**. Activity analysis must be able to identify atomic movements like "walking," "bending," and "falling" on their own while monitoring the daily activities of elderly people, for instance **?**. Therefore, action recognition is a challenging problem with many potential applications.

Action Recognition Datasets Human action recognition aims to understand human activities occurring in a video as humans can understand. While some simple actions, like standing, can be recognised from a single frame (image), most human actions are much more complex and occur over a more extended period. Therefore, they must be observed through consecutive frames (video). To assist organisations in understanding real-time action and dynamic, organic movement, AI/ML models use human action datasets.

Something-Something V2 **?** This publically available dataset is an extensive collection of human-object interaction of densely labelled 174 video sequences. The dataset was created by many crowd workers performing pre-trained daily human–object interaction physical activities; 220,847 videos and JPG images have variable spatial resolutions and lengths.

Egocentric vision, sometimes known as first-person vision, is a sub-field of computer vision that deals with analysing images and videos captured by a wearable camera, often worn on the head or the chest and thus naturally approximates the wearer's visual field. The idea of using egocentric videos has recently been utilised thanks to novel, lightweight and affordable devices such as GoPro and similars **?**. As a fundamental problem in egocentric vision, one of the tasks of egocentric action recognition aims to recognise the actions of the camera wearers from egocentric videos. This community did not have an extensive dataset to be used for pertaining or to have a standard dataset for benchmarking until the appearance of the Epic-Kitchens **???**, the largest and most complete egocentric dataset contains 97 verb classes, 300 noun classes and 3806 action classes. Understanding egocentric videos requires detecting the actor's movement and the object with which the actor interacts.

Several existing methods leveraged object detection to improve egocentric video recognition **????**, among which **?** also incorporate temporal contexts to help understand the ongoing action. These approaches may have limited uses in real-world systems since they demand time-consuming, labour-intensive item detection annotations and are computationally expensive. In contrast, our framework does not depend on costly object detectors. Recently, Shanetal.**?** developed a hand-object detector to locate the active object. When the detector is well-trained, it can be deployed on the target dataset; however, running it on high-resolution frames still costs far more than using our method.

Motion in action recognition: Motion cues**???** have been recognised as necessary for video understanding in the past few years. Most works use optical flow, a motion representation component in many video recognition techniques, to obtain the statistical motion labels required for their work **?**, separating the background from the main objects in optical flow frames. Optical flow is the pattern of visible motion of objects and edges and helps calculate the motion vector of every pixel in a video frame. Optical flow is widely used in many video processing applications as a motion representation feature that can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement. Optical flow-based techniques are sensitive to camera motion since they capture absolute movement. Optical flow computation is one of the fundamental tasks in computer vision. In practice, the flow has been helpful for a wide range of problems, for example, pose estimation **?**, representation learning **?**, segmentation **?**, and even utilised as a tracking substitute for visual signals (RGB images) **?**. Since optical flow can capture continuous or smoothly varying motion, such as motion caused by a change in camera view, it is not a good idea to use it to detect a change in salient objects. To build pixel-level representations from raw high-resolution videos with complex scenes, **?** proposes a self-supervised representation learning framework based on a flow equivariance objective. This representation is beneficial for object detection. In another work **?**, a multi-task motion-guided video salient object detection network is proposed consisting of two sub-networks. One sub-network is used to detect salient objects in still images, and the other is used to detect motion saliency in optical flow images. Most motion descriptors use absolute motions and thus only work well when the camera and background are relatively static, such as Fleet & Jepson's phase-based features **?** and Viola et al.'s generalised wavelet features **?**. Therefore, the critical problem is

identifying characteristics that accurately capture the motion of hands or objects while impervious to the camera and backdrop motion.

Relying only on optical flow to capture the motion is not a robust solution as it is heavily affected by camera motion. To mitigate this problem, **?** presented a self-supervised spatiotemporal video representation by predicting a set of statistical labels derived from motion and appearance statistics using extracting optical flow across each frame and two motion boundaries **?** which are obtained by computing gradients separately on the horizontal and vertical components of the optical flow.

In another line of work, masked autoencoder models have been proposed to learn underlying data distribution in a self-supervised manner without explicitly focusing on motion **?**. Even though this model can perform spatiotemporal reasoning over content, the encoder backbone could be more effective in capturing motion representations. The critical contribution of our work is explicitly imposing motion information in both SSL phases in the self-supervised pretext training without human annotations and then in the finetuning stage, besides introducing an automatic motion detection to detect salient objects and motion in the video without the overhead and limitation of a pretrained and annotated object detector.