# SMILE: Infusing Spatial and Motion Semantics in Masked Video Learning

Fida Mohammad Thoker, Letian Jiang, Chen Zhao†, Bernard Ghanem
King Abdullah University of Science and Technology (KAUST)
{fida.thoker,letian.jiang,chen.zhao,Bernard.Ghanem}@kaust.edu.sa

## Abstract

*Masked video modeling, such as VideoMAE, is an effective paradigm for video self-supervised learning (SSL). However, they are primarily based on reconstructing pixel-level details on natural videos which have substantial temporal redundancy, limiting their capability for semantic representation and sufficient encoding of motion dynamics. To address these issues, this paper introduces a novel SSL approach for video representation learning, dubbed as SMILE, by infusing both spatial and motion semantics. In SMILE, we leverage image-language pretrained models, such as CLIP, to guide the learning process with their high-level spatial semantics. We enhance the representation of motion by introducing synthetic motion patterns in the training data, allowing the model to capture more complex and dynamic content. Furthermore, using SMILE, we establish a new self-supervised video learning paradigm capable of learning strong video representations without requiring any natural video data. We have carried out extensive experiments on 7 datasets with various downstream scenarios. SMILE surpasses current state-of-the-art SSL methods, showcasing its effectiveness in learning more discriminative and generalizable video representations. Code is available:* https://github.com/fmthoker/SMILE

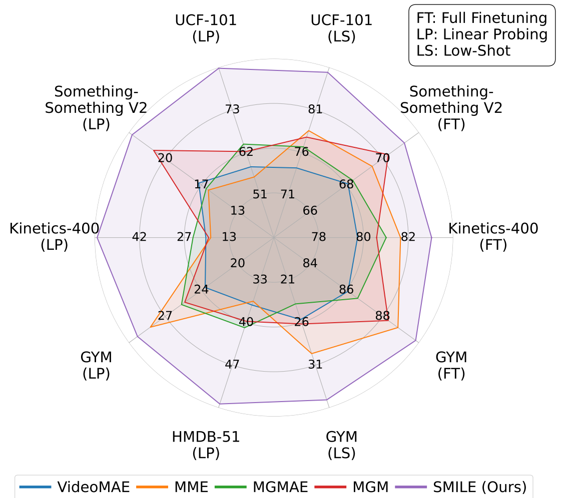


Figure 1. **Comparison with SOTA masked video modeling methods**. Our method significantly outperforms prior masked video modeling methods across diverse downstream settings.

## 1. Introduction

Self-supervised learning (SSL) [43, 44, 57, 59] plays a crucial role for many video understanding tasks, such as action recognition [18, 26, 51, 52, 55] and detection [28, 32], significantly reducing the reliance on extensive human annotation. The goal of video SSL is to learn video representations that capture both spatial and temporal semantics effectively. Masked video modeling has emerged as a promising approach with state-of-the-art performance [12, 21, 53, 57]. It involves masking a large portion of video frames and reconstructing the masked regions using the unmasked parts.

However, this approach has some limitations. *First*, it relies heavily on pixel reconstruction, which can lead to overfitting low-level details. This limits the model's ability to learn meaningful high-level semantics essential for discriminative representations, as shown by the suboptimal linear probing performance of current masked video modeling methods [12, 21, 53, 57] in Figure 1. The *second* key issue in masked video modeling is the substantial temporal redundancy of video data with most videos having little foreground motion, human or object. Consequently, the reconstruction objective can be solved using the temporal correlations between co-located patches or by spatial correlations in tube masking [57], resulting in inadequate motion awareness in the learned representations. This inturn leads to poor generalization on downstream tasks, requiring a finer motion understanding as shown in [55].

In this work, we address these limitations by proposing a novel framework for masked video modeling, dubbed as SMILE (**S**patial and **M**otion semant**I**cs in masked video **LE**arning), to capture high-level spatial and motion semantics leading to generalizable video representations. To improve the spatial semantics in masked video modeling, we propose to explicitly infuse high-level spatial semantics from pretrained image foundation models such as

CLIP [45]. Such models encapsulate high-level semantics learned through large-scale image-text alignment and have been adapted for video understanding tasks. For example, ActionCLIP [61], ViFiCLIP [47], and X-CLIP [36]) adapt CLIP for action recognition using videos with action labels, whereas ViCLIP [64] and VideoPrism [72]) adapt it for video-text alignment with video-text pairs. We take inspiration from them but aim to learn video representations with unlabeled videos. In particular, we propose replacing the reconstruction of pixels in masked video modeling with that of CLIP embedding, thus providing high-level semantics as the supervision signal.

To improve motion awareness, we propose to reduce the temporal redundancy in input training videos with synthetic motion patterns. Specifically, we randomly paste objects onto the video frames and move them smoothly along the temporal dimension to simulate variable temporal transitions of motion-intensive videos. This reduces the temporal redundancy in the original videos and results in videos with an additional prominent foreground motion. Furthermore, adding objects with known locations allows us to add a motion focus to the mask-and-predict task by explicitly masking the regions of object motion. This in turn encourages the network to avoid an over-reliance on static spatial information and encode the motion dynamics of the added objects, when solving the reconstruction task. We summarize our contributions as follows.

- We introduce SMILE, a novel SSL approach for masked video modeling. It encodes spatial and motion semantics using guidance from image-text pretrained models and synthetic-motion augmented videos, leading to more discriminative and generalizable video representation.
- We present a new strategy to augment video data with diverse synthetic motions, paired with a masking strategy that focuses on reconstructing these added motions to improve masked video modeling pretraining.
- Through extensive evaluation on 7 datasets, including Kinetics-400 [26], UCF-101 [52], HMDB-51 [29], FineGYM [50], EPIC-Kitchens-100 [8], SomethingSomething V2 [18], and Charades [51], we demonstrate that SMILE significantly outperforms existing state-of-the-art video SSL methods and exhibits better generalization.

## 2. Related Works

**Video Self-Supervised Learning.** Video SSL aims to learn the spatio-temporal dynamics of video data without the need for any supervision. Three main learning paradigms have emerged: transformation prediction, contrastive learning, and masked video modeling. In transformation prediction, models learn video patterns by solving tasks such as space-time puzzles [1, 25, 27], predicting order in video clips [15, 24, 30, 33, 34, 66], or estimating speed [3, 7, 69]. Contrastive learning is based on instance discrimination [38] and learning invariances to spatio-temporal augmentations [2, 6, 14, 35, 40, 42, 44, 46, 54, 56]. Masked video modeling methods [12, 21, 53, 57, 67], like VideoMAE [57], use a mask-and-predict task to reconstruct videos from partial data with an encoder-decoder setup similar to MAE [20]. By masking a high proportion of pixels (*e.g.*, 90%) and reconstructing them such methods learn useful video representations. Our approach aims to improve the spatial and motion semantics of the video representations learned by masked video modeling.

**CLIP for Video Understanding.** CLIP [45] captures detailed spatial semantics from extensive image-text training, making it a strong foundation for directly modeling video data with added temporal dynamics, *e.g.*, [36, 41, 58, 61, 64, 68]. ViFiCLIP [47] finetunes CLIP on video by adding a temporal pooling layer, ActionCLIP [61] includes a temporal positional embedding and aggregation layer, and X-CLIP [36] integrates cross-frame attention to share information between frames. Most relevant to us are works that also use CLIP for video representation learning [13, 31, 64]. ViCLIP [64] initializes a model with CLIP weights and employs further contrastive pretraining with video-text pairs. UMT [31] uses CLIP as a teacher and aligns its predictions with the CLIP features for unmasked tokens, followed by a 2nd stage of learning from video-text contrastive pairs. TGM [13] relies on CLIP text-encoder features to guide masked video modeling and video-text contrastive learning in a unified framework. In contrast, our SMILE only learns from video data in a self-supervised manner, reconstructing CLIP features of masked tokens and added objects.

**Motion-aware Video Self-Supervised Learning.** Encoding motion dynamics in video representations is essential for generalizing across various video tasks [55]. As a result, many video SSL methods focus on enhancing the motion sensitivity of learned representations. For example, [16, 19, 21, 37, 65] capture motion using optical flow, [9, 60] do so by removing background or static elements and [11, 22, 56, 59] incorporate synthetic motions. [22, 59] learn by predicting the speed and trajectory of the pseudo motions, and [56] contrasts video with different types of motions. These approaches are specific to transformation prediction and contrastive learning paradigms. We take inspiration from these works and inject synthetic motions into the masked video learning.

Most relevant to our work are methods like [12, 21, 53, 67], which enhance motion understanding in VideoMAE [57] by modifying certain design choices. MGMAE [21] and MGM [12] improve masking strategies to mask the regions of motion using optical flow and motion vectors. MotionMAE [67] adds frame difference as an additional target for reconstruction, and MME [53] uses optical flow to extract dense object trajectories and then predicts these trajectories and their HOG features. However, these methods are limited by the temporal redundancies in the original data, restricting the amount of motion or motion-focused regions available for learning in the reconstruction task. In contrast, we add synthetic motions to the original video data, reducing temporal redundancy and increasing motion-rich regions to learn from. As shown in our experiments, reconstructing these modified videos encourages the model to capture motion dynamics, enhancing the motion awareness of the learned video representations.

# 3. Methodology

## 3.1. Revisiting Masked Video Modeling

The goal of masked video modeling [12, 21, 57] is to learn video representations by employing a mask-and-predict task with an asymmetric encoder-decoder architecture. An input video $V \in \mathbb{R}^{T\times H\times W\times 3}$ is first divided into non-overlapped 3D space-time tokens (*e.g.*, $2 \times 16 \times 16$) as $\mathcal{T} = \{v_1, \ldots, v_N\}$. From $\mathcal{T}$, a subset of tokens $\mathcal{T}^{mask}$ are randomly masked with a high masking ratio (*e.g.*, 90%) and the unmasked tokens $\mathcal{T}^{unmask} = \mathcal{T} \setminus \mathcal{T}^{mask}$ are fed into a transformer encoder $\Phi_{enc}$. Then, the output of the encoder and the learnable tokens containing the positional information of the masked tokens are passed to a shallow decoder $\Phi_{dec}$ to reconstruct the input pixel with the following loss:

$$\mathcal{L} = \frac{1}{|\mathcal{T}^{mask}|} \sum_{i\in\mathcal{T}^{mask}} \|v_i - Y_i\|_2^2, \tag{1}$$

where $v_i$ refers to the *pixel values* of the $i$-th space-time token in $\mathcal{T}^{mask}$ and $Y_i = \Phi_{dec}(\Phi_{enc}(\mathcal{T}^{unmask}))$ denotes the corresponding reconstructed pixels for $i$-th token given the set of all unmasked tokens $\mathcal{T}^{unmask}$.

Such pixel reconstruction drives the model to focus on low-level features, and with high temporal redundancy in video data, it primarily reconstructs static appearance details of video data. However, effective video understanding also requires capturing temporal progression and high-level semantics, such as scene context and object interactions.

## 3.2. General Scheme of SMILE

Our goal is to infuse spatial and motion semantics in masked video modeling. To achieve this, we integrate two key elements into the masked video modeling paradigm. To improve motion semantics, we overlay local motion patterns in input videos in the form of moving objects. To improve spatial semantics, we replace the low-level pixel targets with high-level features extracted from a pretrained CLIP encoder. By combining both, we learn video representations that can encode both spatial and motion dynamics necessary for modeling downstream video tasks. The overview of our method is shown in Figure 2.

**Motion Infusion.** Given a input video $V \in \mathbb{R}^{T\times H\times W\times 3}$. We first sample a random object $o$ from a set of predefined segmented objects $O$. Next, we generate a motion trajectory (Section 3.3), along which we overlay this object onto the video $V$ to obtain the motion-augmented video $V'$. By adding objects with synthetic motions, we reduce temporal redundancy in the original videos and encourage the model to learn from the motion dynamics of the overlaid objects.

**Masking.** We employ two separate masking strategies on $V'$ as follows. For the original regions, we employ random tube masking as in VideoMAE [57]. For added object regions, we propose *trajectory-based masking*. Since we know the location of the object in each frame of $V'$, we explicitly drop space-time tokens at random along its trajectory. Employing a high masking ratio along the object trajectory ensures that solving the reconstruction task requires encoding the object's motion dynamics, thereby enhancing the motion awareness of learned representations.

**Architecture.** We employ a teacher-student reconstruction architecture with a pretrained image-encoder as the teacher and an asymmetric encoder-decoder network $\Phi$ as the student. The augmented video is patchified into 3D space-time tokens $\mathcal{T} = \{v_1^{'}, \ldots, v_N^{'}\}$ as in general video masked autoencoders (Section 3.1). The tokens are then masked using tube and trajectory-based masking to generate sets of masked $\mathcal{T}^{mask}$ and unmasked $\mathcal{T}^{unmask}$ tokens. The unmasked tokens $\mathcal{T}^{unmask}$ and the learnable tokens containing the positional information of the masked patches are used as the input to $\Phi$ as in Section 3.1.

**Reconstruction.** Instead of directly reconstructing pixels, we project the input video $V'$ onto the teacher's feature space to generate high-level targets. In particular, we choose a CLIP model as the projection network due to its high-level semantics learned from language guidance. Specifically, the input $V'$ is passed frame-by-frame through the CLIP encoder to extract features. The extracted features are stacked into a sequence of feature tokens as $\mathcal{F} = \{f_1^{'}, \ldots, f_N^{'}\}$. To match the number of tokens with $\mathcal{T}$, we only use the CLIP features for the first time slice of the 3D space-time tokens, thus generating a total of $N$ feature tokens, each corresponding to a space-time token in the input video. The encoder-decoder is then trained with a masked feature reconstruction loss as:

$$\mathcal{L} = \frac{1}{|\mathcal{T}^{mask}|} \sum_{i\in\mathcal{T}^{mask}} \|f_i^{'} - Y_i\|_2^2, \tag{2}$$

where $f_i^{'}$ is the *CLIP feature projection* of $i$-th space-time token $v_i^{'}$ in $\mathcal{T}^{mask}$ and $Y_i = \Phi_{dec}(\Phi_{enc}(\mathcal{T}^{unmask}))$ denotes the corresponding reconstructed feature for $i$-th token given the set of all unmasked tokens $\mathcal{T}^{unmask}$. Such formulation first increases the temporal dynamics in otherwise static data, avoiding learning shortcuts for reconstruction. Second, projecting targets onto the CLIP space encourages the network to capture more high-level video semantics.

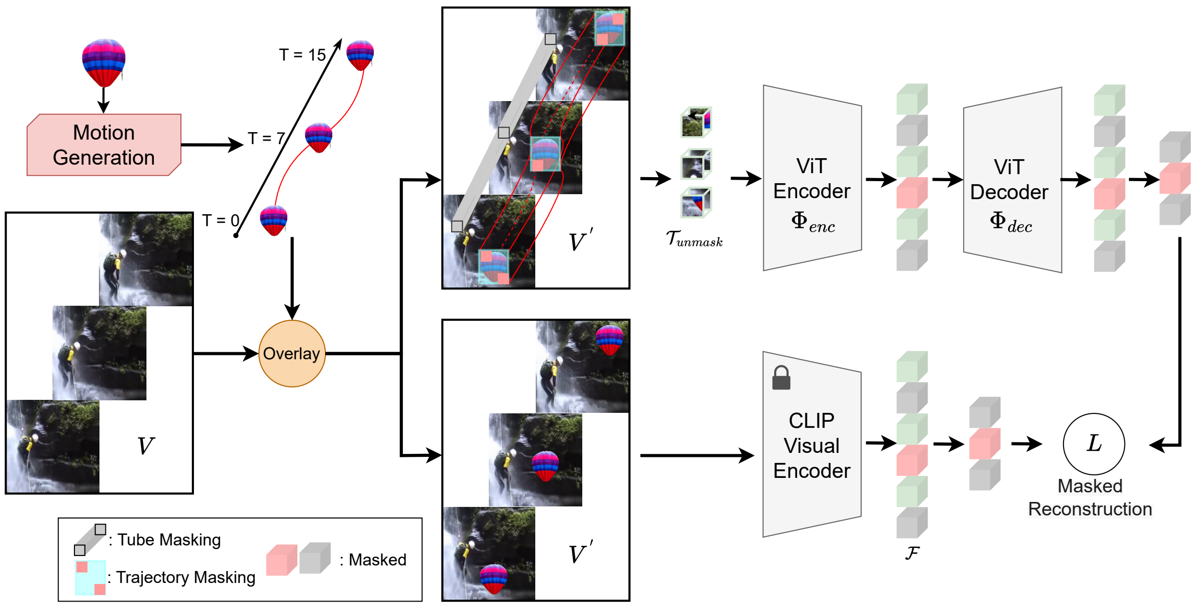


Figure 2. **Overall architecture of our SMILE.** An input video clip $V$ is overlaid with a segmented object along a randomly generated trajectory to generate $V^{'}$ infusing synthetic object motion in $V$. $V^{'}$ is passed frame-by-frame through the CLIP encoder to extract feature tokens $\mathcal{F}$. $V^{'}$ is patchified into a set of space-time tokens $\mathcal{T}$. We apply two types of masking upon $\mathcal{T}$, tube masking on the space-time tokens of the original video and trajectory-based masking on the tokens of added objects. The unmasked tokens $\mathcal{T}_{unmask}$ are fed into the encoder-decoder network $\Phi_{enc}\Phi_{dec}$ which is trained to reconstruct the masked feature tokens in $\mathcal{F}$.

Overall, the key intuition is that CLIP features capture high-level information, such as scene context and object interactions, which guides the network to attend to both the overlaid object motions and the original video semantics.

### 3.3. Motion Generation

To infuse motion, we overlay a segmented object along a randomly generated trajectory onto the original video frames. To introduce more complex dynamics, the object is also transformed *e.g.* rotation, along the trajectory.

**Appearance and Trajectory.** To infuse a new synthetic motion onto a video sequence $V{=}[v_1, v_2, ..., v_T]$ comprising $T$ frames, we begin by selecting an object $o$ from $O$ at random, where $O$ denotes the set of off-the-shelf segmented objects generated using Stable Diffusion [48] and X-Paste[71]. We then sample a scale $P{\times}Q$ from a predefined set of sizes for $o$ to model different object sizes, where $P \ll H$ and $Q \ll W$. Next, we sample a sequence of locations for the object $\mathrm{Traj}_o = [(x^1, y^1), .., (x^T, y^T)]$ representing the center coordinate $(x^i, y^i)$ of the object $o$ in the corresponding frame. Specifically, we generate a random non-linear path for the trajectory by sampling $M$ 2D points ($M \gg T$) uniformly from $x \in [0, H]$ and $y \in [0, W]$. To ensure smoothness in the motion, a $1D$ Gaussian filter is applied along $x$ and $y$ dimensions as $[(h(x^1), h(y^1)), \ldots, (h(x^M), h(y^M))]$, where $h(z) = \frac{1}{\sqrt{2\pi}\kappa} e^{-z^2/2\kappa^2}$ and $\kappa$ is the smoothing factor. The resulting trajectory is then downsampled to $T$ target locations for the object $o$. This formulation introduces diverse motions with varying spatio-temporal dynamics in the form of objects with different shapes, sizes, and temporal progression.


Figure 3. **Motion overlaid videos** showing the position change and transformation of the added objects along the time dimension.

**Transformation.** To increase the complexity of the object's motion, we also apply scale and rotation transformations along the temporal dimension. Specifically, we select 3 frame indices, the first, the last, and a random middle one, $v_1$, $v_m$, and $v_T$, respectively. For these three frame indices, a rotation angle $\alpha_i$, and a scaling factor $S_i$ are selected randomly. The angles and scales for the rest of the frames are linearly interpolated between these keyframes. Finally, the object $o$ is rotated and scaled with the corresponding transformations $\begin{pmatrix} \cos\alpha_i & -\sin\alpha_i \\ \sin\alpha_i & \cos\alpha_i \end{pmatrix}$ and $\begin{pmatrix} S_i & 0 \\ 0 & S_i \end{pmatrix}$, before overlaying it onto the video frame $v_i$.

Figure 3 shows two sample videos with our added motions. While these motions may not appear entirely realistic, they represent complex motion dynamics of moving structured pixels, and introducing them into a mask-and-predict framework guides the model to learn from such dynamics.

This results in video representations with improved downstream performance on motion-aware tasks, as we also validate empirically. Interestingly, we also show in the experiments (Section 5) that overlaying such object motions on black or noise frames can learn strong video representations without requiring any original videos.

### 3.4. Training Procedure

By adding synthetic motions, we have introduced a domain shift in the input for reconstruction, which may affect the performance of downstream tasks. To explore effective learning from both original videos $V$ and motion-overlaid videos $V'$, we explore two training strategies, *i.e.* **mixed training** and **progressive training.** In the *mixed training* strategy, we use both $V$ and $V'$ as the input to the network and directly optimize for the reconstruction of both inputs. The idea is to simultaneously encode the spatio-temporal dynamics of the original video data and the motion dynamics of the augmented data, with a limited domain shift. In the *progressive training* strategy, we employ a two-stage learning approach that first learns from augmented data $V'$ alone, then continues to train the model with the original data $V$ alone. This way we can first learn to encode the spatio-temporal dynamics of the augmented data and then align the network to further encode the dynamics of the original data and its domain.

## 4. Experiments

We evaluate SMILE against prior video SSL methods on action recognition benchmarks: Kinetics-400 (K400) [26], SomethingSomething V2 [18] (SSv2), UCF-101 [52] (UCF), HMDB-51 [29] (HMDB), FineGYM [50] (GYM), and EPIC-Kitchens-100 [8] (EPIC), achieving the best performance in linear probing and full finetuning (Section 4.2). SMILE also shows better generalization on the SEVERE benchmark[55] (Section 4.4). Ablations and further discussion are provided in Section 4.3 and Section 5, respectively.

### 4.1. Implementation Details

**Model.** Following VideoMAE [57], we use ViT-S or ViT-B with joint space-time attention for our student network. We choose CLIP ViT-B [45] as our teacher network.

**Motions.** We follow PIN [10] to create the set of segmented objects $O$ with Stable Diffusion [48] and X-Paste [71]. We sample the object size $P \times Q$ uniformly from $[32 \times 32, 128 \times 128]$ and sample the rotation angles from $[-90, 90]$ and scaling factors from $[0.5, 1.5]$.

**Pretraining.** We follow previous masked video modeling works [12, 21, 53, 57] and use Kinetics-400 [26] (K400) or SomethingSomething V2 [18] (SSv2) for pretraining unless specified otherwise. We use the same hyperparameters as in VideoMAE [57]. Unless stated otherwise, we train with our proposed *progressive training* for a total of 600 epochs, 300 epochs in each stage. More training details are in **supplementary** material.

Table 1. **Linear probing comparison of masked video modeling methods.** Our SMILE learns highly discriminative video representations useful for both spatial and motion-focused domains. All results are obtained with the ViT-B backbone pretrained on K400 and Top-1 accuracies (%) are reported.

| Method | K400 | UCF | HMDB | SSv2 | GYM | EPIC |
|---|---|---|---|---|---|---|
| VideoMAE [57] | 20.7 | 58.6 | 37.7 | 17.5 | 23.9 | 33.2 |
| MVD [63] | 18.7 | 49.1 | 28.6 | 12.2 | 22.7 | 29.7 |
| MME [53] | 19.1 | 56.0 | 37.1 | 16.6 | 29.0 | 32.2 |
| MGMAE [21] | 24.9 | 64.4 | 41.3 | 16.8 | 26.1 | 33.2 |
| EVEREST [23] | 14.1 | 51.8 | 30.3 | 14.5 | 23.3 | 30.5 |
| MGM [12] | 19.8 | 62.5 | 40.3 | <u>21.7</u> | 25.8 | 32.4 |
| SIGMA [49] | 47.5 | 80.7 | <u>52.3</u> | 20.8 | <u>30.1</u> | <u>34.2</u> |
| SMILE w/o motion | <u>54.6</u> | <u>82.6</u> | 51.7 | 21.6 | 27.9 | 32.3 |
| SMILE (ours) | **56.2** | **83.8** | **53.4** | **23.7** | **30.2** | **34.4** |

### 4.2. Comparison with State-of-the-Art

**Linear Probing.** We freeze the pretrained backbone and only train a linear classifier on the target dataset. When evaluating the prior works for comparison, we utilize their publicly available checkpoints for the ViT-B backbone pretrained on the K400 dataset for 800 epochs except for MVD (2000 epochs) and MME (1600 epochs). We evaluate the downstream task using a common setup for a fair comparison. Details about datasets and experimental settings are in the **supplementary** material.

Table 1 shows that our method, even without synthetic motion, significantly outperforms all state-of-the-art methods on K400 and UCF, and is among the top performers on HMDB, SSv2, GYM, and EPIC. This highlights that CLIP feature reconstruction offers a stronger supervisory signal than pixel reconstruction, especially in tasks requiring spatial semantics like K400 and UCF. Adding synthetic motions further boosts performance by an average of $1.8\%$ across datasets, enhancing motion cues through reconstruction. On motion-sensitive datasets like SSv2, GYM, and EPIC, our approach consistently surpasses MGM [12], MGMAE [21], and MME [53], confirming that synthetic motions help overcome the lack of motion diversity in current datasets that may limit these methods.

**Full Finetuning.** We finetune both the pretrained backbone and the classification head end-to-end following [21, 53, 57] on the downstream datasets K400 and SSv2. Following prior masked video modeling works [12, 21, 57], we show results with self-supervised pretraining on K400 and SSv2 datasets for the VIT-B backbone. We follow VideoMAE [57] for finetuning and evaluation protocols with more details in the **supplementary**.

Tables 2 and 3 show the results for SSv2 and K400, respectively. On SSv2, we achieve the best performance with both K400 and SSv2 pretraining, highlighting our method's effectiveness for both in-domain and cross-domain transfer.

Table 2. **Full finetuning comparison of various self-supervised methods on Something-Something V2**. All results are obtained with a ViT-B backbone. $*$ denotes results obtained by our evaluation. Our SMILE achieves state-of-the-art performance. Top-1 accuracies (%) are reported.

| Method | Backbone | Epochs | Pretrain | Top-1 |
|---|---|---|---|---|
| VideoMAE [57] | ViT-B | 800 | K400 | 68.5 |
| OmniMAE [17] | ViT-B | 800 | K400 | 69.0 |
| MGMAE* [21] | ViT-B | 800 | K400 | 68.9 |
| MME [53] | ViT-B | 800 | K400 | 70.5 |
| SIGMA [49] | ViT-B | 800 | K400 | 71.1 |
| MGM* [12] | ViT-B | 800 | K400 | 71.1 |
| SMILE (ours) | ViT-B | 600 | K400 | **72.1** |
| OmniMAE [17] | ViT-B | 800 | SSv2 | 69.5 |
| VideoMAE [57] | ViT-B | 800 | SSv2 | 69.6 |
| MGM [12] | ViT-B | 800 | SSv2 | 70.6 |
| MME [53] | ViT-B | 800 | SSv2 | 70.0 |
| SIGMA [49] | ViT-B | 800 | SSv2 | 71.2 |
| MGMAE [21] | ViT-B | 800 | SSv2 | 71.0 |
| SMILE (ours) | ViT-B | 800 | SSv2 | **72.5** |

For K400, we exceed the prior top SSL approach by 1.6%. Additionally, our gains over motion-focused methods, such as MGMAE (1.5% on SSv2, 1.9% on K400), MME (2.5% on SSv2, 1.6% on K400), and MGM (1.9% on SSv2, 2.3% on K400), confirm our improved motion sensitivity, even in full-finetuning settings. Notably, our model surpasses VideoMAE, MotionMAE, MME, and MGMAE trained for more than twice the number of epochs (see **supplementary** for the extensive comparison). We attribute this efficiency to CLIP feature reconstruction, which directly captures high-level semantics compared to pixel-level learning.

### 4.3. Ablations

We ablate various components of our SMILE to show their impact on the downstream performance in Table 4. For computational efficiency, we use ViT-S as the backbone and pretrain with a smaller subset of K400, denoted as $\text{K400}_\text{m}$, containing around 80K samples. For evaluation, we use $\text{K400}_\text{m}$ and $\text{SSv2}_\text{m}$ that denotes a subset of SSv2 with 50% training data. Unless stated otherwise, we use CLIP features as the reconstruction target, add two synthetic objects, mask 90%, use trajectory masking, and train for 400 epochs.

**Synthetic motion, *yes* or *no*?** In Table 4a, we show the impact of adding synthetic motions to the masked video modeling framework for both pixel and feature reconstruction. We can see that adding our synthetic motion patterns significantly improves the downstream performance of both pixel and feature reconstruction.

**Reconstructing *pixels* or *features*?** Table 4h compares pixel-based and feature-based reconstruction targets in masked video modeling. Reconstructing in feature space significantly boosts performance over pixel reconstruction, with a +12.1% gain on spatially-focused tasks like $\text{K400}_\text{m}$ and +5.1% on motion-focused tasks like $\text{SSv2}_\text{m}$. This suggests that using a high-level feature space as the reconstruction target provides more meaningful supervision for learning transferable spatio-temporal representations.

Table 3. **Full finetuning comparison of various self-supervised methods on Kinetics-400**. All results are obtained with a ViT-B backbone. $*$ denotes results obtained by our evaluation. Our SMILE achieves state-of-the-art performance. Top-1 accuracies (%) are reported.

| Method | Backbone | Epochs | Pretrain | Top-1 |
|---|---|---|---|---|
| SVT [46] | ViT-B | - | K400 | 78.4 |
| VideoMAE [57] | ViT-B | 800 | K400 | 80.0 |
| BEVT [62] | ViT-B | - | K400 | 80.6 |
| OmniMAE [17] | ViT-B | 800 | K400 | 80.8 |
| MGM [12] | ViT-B | 800 | K400 | 80.8 |
| MGMAE [21] | ViT-B | 800 | K400 | 81.2 |
| SIGMA [49] | ViT-B | 800 | K400 | 81.5 |
| MME* [53] | ViT-B | 800 | K400 | 81.5 |
| SMILE (ours) | ViT-B | 600 | K400 | **83.1** |

**Reconstructing which features: *CLIP*, *MAE* or *DINO*?** Table 4b compares several image foundation models—MAE [20], DINO [4], DINO-v2 [39], and CLIP [45]—as reconstruction targets. CLIP features outperform the others, likely due to CLIP's image-text alignment training on a large dataset, enhancing its transferability and adaptability to video tasks as shown in [36, 47, 61]. Compared to pixel reconstruction in Table 4h, all feature-based approaches show substantial performance gains.

**Trajectory-based masking: *yes* or *no?*** Table 4g shows the impact of our trajectory-based masking. Without trajectory masking, all tokens are masked using tube masking, which masks the whole video without distinguishing between object and non-object space-time tokens. In contrast, trajectory masking first masks space-time tokens associated with objects, followed by tube masking on the remaining tokens, maintaining the overall masking ratio. Our trajectory masking improves performance by approximately 1% on both $\text{K400}_\text{m}$ and $\text{SSv2}_\text{m}$, validating that masking along object trajectories enhances motion-aware representation learning.

**How much to mask?** Table 4e shows the impact of masking ratio $m$ on downstream performance. We mask $m$ of space-time tubes within the objects and $m$ of space-time tubes in the whole video with $m$ being $95\%, 90\%$, or $80\%$. We observe the best performance with $m$=80%.

**How many synthetic objects?** In Table 4f, we analyze the impact of adding more object motions to the input video on downstream performance. We evaluate adding $1, 2$ and $3$ object motions and compare the results. We observe that adding 2 objects is better than using a single object and the performance saturates when adding more objects.

**Which object transformations?** In Table 4c, we show the impact of object transformation on downstream performance. Both rotation and scaling improve the performance individually and combining them achieves the best results.

| Synth. | Target | K400$_m$ | SSv2$_m$ |
|---|---|---|---|
| w/o | Pixels | 66.0 | 55.1 |
| w/ | Pixels | **68.1** | **56.8** |
| w/o | Features | 78.1 | 60.2 |
| w/ | Features | **78.9** | **61.6** |

(a) **Synthetic motion, *yes* or *no*?** Yes. It improves performance for both pixel and feature targets.

| Target | K400$_m$ | SSv2$_m$ |
|---|---|---|
| MAE [20] | 77.6 | 59.5 |
| DINO [4] | 77.6 | 59.7 |
| DINOv2 [39] | 77.8 | 59.7 |
| CLIP [45] | **78.1** | **60.2** |

(b) **Reconstructing which features?** CLIP features provide the best target. No motion is added.

| Transform | K400$_m$ | SSv2$_m$ |
|---|---|---|
| None | 77.0 | 60.8 |
| Scaling | 78.4 | 61.3 |
| Rotation | 78.2 | 61.4 |
| Both | **78.9** | **61.6** |

(c) **Which object *transformations*?** Using both rotation and scaling works the best.

| Training | K400$_m$ | SSv2$_m$ |
|---|---|---|
| Single | 78.9 | 61.6 |
| Mix | 79.5 | 62.2 |
| Progressive | **79.8** | **62.5** |

(d) **Which training strategy?** Progressive training outperforms both for the same number of iterations.

| Ratio | Target | K400$_m$ | SSv2$_m$ |
|---|---|---|---|
| 95% | features | 77.7 | 60.4 |
| 90% | features | 78.9 | 61.6 |
| 80% | features | **80.3** | **62.0** |

(e) **How much to mask?** Masking 80% trajectory and total tokens.

| # Objects | K400$_m$ | SSv2$_m$ |
|---|---|---|
| 1 | 78.6 | 61.0 |
| 2 | **78.9** | **61.6** |
| 3 | 78.6 | 61.3 |

(f) **How many *objects*?** Adding two object motions is the best.

| Masking | K400$_m$ | SSv2$_m$ |
|---|---|---|
| w/o Trajectory | 77.8 | 60.7 |
| w/ Trajectory | **78.9** | **61.6** |

(g) **Trajectory masking, *yes* or *no*?** Yes. Both targets are features.

| Target | K400$_m$ | SSv2$_m$ |
|---|---|---|
| Pixels | 66.0 | 55.1 |
| Features | **78.1** | **60.2** |

(h) **Reconstructing *pixels* or *features*?** Reconstructing features is much better. No motion is added.

Table 4. **Ablation experiments of our proposed SMILE.** The default setting uses clip reconstruction targets with two synthetic objects and a 90% masking ratio with trajectory-based masking. We also show full-scale ablations with ViT-B and K400 pretraining in **supplementary**.

**Which training strategy?** Table 4d compares our proposed training strategies. We also compare learning with the single learning strategy, where only augmented videos are used for training. We observe that both mixed and progressive learning are better than learning from augmented videos only. Progressive is better than mixed, most likely due to a reduced domain shift in the second stage where only original videos are used.

**Qualitative analysis.** To compare the temporal dynamics of video representations learned by different SSL methods, we compute their feature similarity across different frames of input videos (Figure 4). For our model, the features of different frames have larger differences, indicating better temporal awareness. More plots are in the **supplementary**.

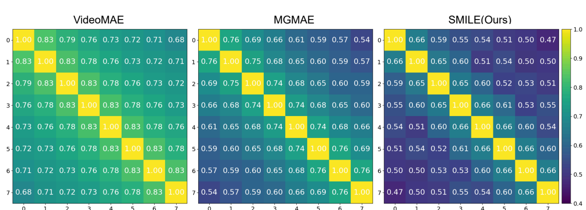


Figure 4. **Feature similarity across different frames for different SSL methods.** We compute this on K400 validation videos.

### 4.4. Generalization Assessment

To demonstrate the generalization of our learned video representations to diverse downstream video setups, we evaluate SMILE on the SEVERE benchmark [55], which includes 8 experiments targeting 4 key generalization factors: *domain shift*, *sample efficiency*, *action granularity*, and *task shift*. Detailed configurations are provided in the **supplementary**. Beyond SEVERE, we also show generalization to more temporal tasks in **supplementary**.

Table 6 shows the results. For the **Domain Shift**, we test cross-domain adaptability on SSv2 and Gym99, which differ from the pretraining dataset (K400). Our method significantly outperforms all others, demonstrating superior adaptability to domain shifts. For **Sample Efficiency**, we evaluate low-shot action recognition on UCF and GYM with only 1,000 samples for finetuning. Our approach surpasses previous methods, achieving a 5% improvement on GYM over the best prior model, MME, showcasing its strong low-shot recognition capability even in motion-sensitive tasks. For the **Action Granularity**, tested on FX-S1 and UB-S1 from FineGym, our model achieves the best results, excelling at distinguishing fine-grained motions within gymnastics routines (e.g., jump variations in FX-S1). It outperforms motion-aware models like MME, MGMAE, and MGM, highlighting superior motion generalizability. For **Task Shift**, evaluated with temporal repetition counting on UCF-RC [70] and multi-label recognition on Charades [51], our method matches prior performance on repetition counting and achieves a 9% improvement over MME on Charades, demonstrating broad task adaptability. Overall, SMILE (without synthetic motion) is on par with the prior best with the mean performance of 64.9%. With synthetic motion SMILE, we observe a further 3.0% gain, reinforcing our model's generalization strength through synthetic motion.

## 5. Discussions

**Comparison with Other CLIP Adaptations.** Table 5 presents a comparison between our SMILE and the CLIP adaptation methods ViCLIP [64] and UMT [31], along with the CLIP baseline. Our method achieves superior performance on motion-sensitive datasets, particularly in linear probing, indicating that our CLIP adaptation learns better video representations than the video-text alignment of ViCLIP and the unmasked feature alignment of UMT, despite using less data. More results are in the **supplementary**.

| Pretraining | | Finetuning | | | | Linear Probing | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | Video Dataset | UCF | GYM | SSv2 | EPIC | UCF | GYM | SSv2 | EPIC |
| CLIP [45] | - | 93.6 | 88.0 | 66.7 | 50.3 | 77.5 | 20.7 | 11.3 | 25.1 |
| ViCLIP [64] | Intervid-10M [64] | 95.2 | 89.7 | 67.9 | 55.0 | 86.7 | 27.3 | 18.9 | 27.3 |
| UMT [31] | K700 [5] | 96.0 | 89.9 | 70.1 | 50.1 | **88.0** | 26.4 | 18.8 | 28.2 |
| SMILE (ours) | K400 [26] | **96.4** | **90.8** | **71.9** | **63.3** | 83.8 | **30.2** | **23.7** | **34.6** |

Table 5. **Comparision with CLIP adaptations for video representation learning.** Despite using less data, we learn better video representations than other CLIP adaptations, especially for motion-focused downstream domains.

Table 6. **Generalization assessment on SEVERE benchmark [55]** for recent video SSL methods in terms of domain shift, sample efficiency, action granularity, and task shift. We evaluate prior works with the official ViT-B backbone checkpoints pretrained on the K400 dataset. SMILE shows the best generalization performance across setups. Adding synthetic motions further improves the performance.

| **Method** | **Domain shift** | | **Sample efficiency** ($10^3$) | | **Action granularity** | | **Task shift** | | **Mean** |
|---|---|---|---|---|---|---|---|---|---|
| | SSv2 | Gym99 | UCF | GYM | FX-S1 | UB-S1 | UCF-RC↓ | Charades | |
| VideoMAE [57] | 68.6 | 86.6 | 74.6 | 25.9 | 42.8 | 65.3 | 0.172 | 14.4 | 57.6 |
| MVD [63] | 70.0 | 82.5 | 67.1 | 17.5 | 31.3 | 50.5 | 0.184 | 16.1 | 52.1 |
| MGMAE [21] | 68.9 | 87.2 | 77.2 | 24.1 | 33.7 | 79.5 | 0.181 | 17.9 | 58.8 |
| MGM [12] | 71.1 | 89.1 | 78.4 | 26.4 | 38.6 | 86.9 | **0.152** | 22.5 | 62.2 |
| SIGMA [49] | 70.9 | 89.7 | 84.1 | 28.0 | 55.1 | 79.9 | 0.169 | 23.1 | 64.2 |
| MME [53] | 70.1 | 89.7 | 79.2 | 29.8 | **55.5** | 87.2 | 0.155 | 23.6 | 65.0 |
| SMILE w/o motion | 71.6 | 90.0 | 85.2 | 32.0 | 53.4 | 74.3 | 0.175 | 30.5 | 64.9 |
| SMILE (ours) | **72.1** | **90.8** | **86.4** | **35.1** | 55.1 | **88.3** | 0.170 | **32.5** | **67.9** |

Table 7. **Learning without natural videos.** We train a ViT-S for pixel reconstruction. Our synthetic object motions can learn video representations without using any natural videos.

| **Unnatural Video Scheme** | Pretrain Data | K400$_m$ | SSv2$_m$ |
|---|---|---|---|
| No Pretraining | - | 39.7 | 28.2 |
| Video-frame | K400$_m$-1frm | 54.0 | 43.1 |
| Video-frame + Motion | K400$_m$-1frm | 63.6 | 53.2 |
| Image | Places-80K | 52.1 | 40.5 |
| Image + Motion | Places-80K | 60.1 | 52.1 |
| Black image + Motion | - | 59.2 | 50.0 |
| Noise image + Motion | - | 61.9 | 51.2 |

**Learning without Natural Videos.** In prior experiments, we overlay our object motions onto natural videos from the Kinetics-400 dataset, raising the question of whether downstream improvements stem from original video priors (motion, action, scene) or our added object motions. To investigate this, we construct videos by overlaying object motions onto four different backgrounds: (1) a single frame from natural videos, (2) a single scene image from the image dataset Places [73], (3) a plain black image, and (4) a noise image. Each background image is duplicated across $T$ frames, creating static clips onto which moving objects are placed. For (1), we select one frame per video from K400$_m$ (denoted K400$_m$-1frm), preserving action and scene cues without motion priors. For (2), we sample 80K images from Places (Places-80K), matching the dataset size of K400$_m$-1frm, to remove action and motion priors. Approaches (3) and (4) exclude all natural video or image priors entirely. Evaluation involves full finetuning on K400$_m$ and SSv2$_m$, consistent with our previous ablation studies.

Table 7 reports results for different augmented video types. Adding object motions consistently enhances downstream performance, whether combined with action and scene cues from natural videos (row 2 vs. row 3) or only scene cues from images (row 4 vs. row 5). Even with black or noisy backgrounds (rows 6 and 7), performance remains robust. Remarkably, using object motions alone yields substantial improvements—22.2% on K400$_m$ and 23.0% on SSv2$_m$ compared to no pretraining. These findings demonstrate our method's effectiveness in learning video representations from synthetic motions alone. More results are in the **supplementary**. In summary, our method enables a new paradigm of learning strong video representations with masked video modeling without relying on natural videos.

## 6. Conclusions

In this work, we proposed SMILE, a novel self-supervised approach for video representation learning that addresses key limitations in existing masked video modeling methods by integrating both spatial and motion semantics. Our SMILE, leverages the high-level spatial semantics from image-language pretrained models like CLIP and enhances motion representation through synthetic motion patterns. By capturing complex spatial features and motion dynamics, SMILE achieves superior performance on diverse downstream tasks, demonstrating improved discrimination and generalization over state-of-the-art video SSL methods. Our extensive evaluations demonstrated the advantages of our SMILE, setting a new paradigm for robust and effective video self-supervised learning.

**Acknowledgements.** This work is supported by the KAUST Center of Excellence for Generative AI under award number 5940. For computing time, this research used Ibex managed by the Supercomputing Core Laboratory at King Abdullah University of Science & Technology (KAUST) in Thuwal, Saudi Arabia. We also thank Michael Dorkenwald for providing the synthetic objects used in this work.

## References

[1] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2

[2] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[3] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 6, 7

[5] 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. 8

[6] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. Rspnet: Relative speed perception for unsupervised video representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 2

[7] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised visual learning by variable playback speeds prediction of a video. *IEEE Access*, 9:79562–79571, 2021. 2

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision (IJCV)*, 2021. 2, 5

[9] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. Motion-aware contrastive video representation learning via foreground-background merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[10] Michael Dorkenwald, Nimrod Barazani, Cees GM Snoek, and Yuki M Asano. Pin: Positional insert unlocks object localisation abilities in vlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 5

[11] Hazel Doughty, Fida Mohammad Thoker, and Cees GM Snoek. Locomotion: Learning motion-focused video-language representations. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 50–70, 2024. 2

[12] David Fan, Jue Wang, Shuai Liao, Yi Zhu, Vimal Bhat, Hector Santos-Villalobos, Rohith MV, and Xinyu Li. Motion-guided masking for spatiotemporal representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3, 5, 6, 8

[13] David Fan, Jue Wang, Shuai Liao, Zhikang Zhang, Vimal Bhat, and Xinyu Li. Text-guided video masked autoencoder. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[14] 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 (CVPR)*, 2021. 2

[15] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[16] Kirill Gavrilyuk, Mihir Jain, Ilia Karmanov, and Cees G M Snoek. Motion-augmented self-training for video recognition at smaller scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[17] Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Omnimae: Single model masked pretraining on images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

[18] 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 (ICCV)*, 2017. 1, 2, 5

[19] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[20] 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 (CVPR)*, 2022. 2, 6, 7

[21] Bingkun Huang, Zhiyu Zhao, Guozhen Zhang, Yu Qiao, and Limin Wang. Mgmae: Motion guided masking for video masked autoencoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3, 5, 6, 8

[22] Ziyuan Huang, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Rong Jin, and Marcelo H Ang. Self-supervised motion

learning from static images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[23] Sunil Hwang, Jaehong Yoon, Youngwan Lee, and Sung Ju Hwang. Everest: Efficient masked video autoencoder by removing redundant spatiotemporal tokens. In *International Conference on Machine Learning (ICML)*, 2024. 5

[24] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[25] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*, 2018. 2

[26] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2, 5, 8

[27] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 2

[28] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified CNN architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644*, 2019. 1

[29] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 2, 5

[30] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[31] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 7, 8

[32] Shuming Liu, Chen-Lin Zhang, Chen Zhao, and Bernard Ghanem. End-to-end temporal action detection with 1b parameters across 1000 frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[33] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2

[34] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[35] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[36] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 6

[37] Jingcheng Ni, Nan Zhou, Jie Qin, Qian Wu, Junqi Liu, Boxun Li, and Di Huang. Motion sensitive contrastive learning for self-supervised video representation. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[38] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[39] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6, 7

[40] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[41] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Dual-path adaptation from image to video transformers. *arXiv preprint arXiv:2303.09857*, 2023. 2

[42] Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

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

[44] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 2, 5, 6, 7, 8

[46] 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 (CVPR)*, 2022. 2, 6

[47] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned

clip models are efficient video learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 6

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4, 5

[49] Mohammadreza Salehi, Michael Dorkenwald, Fida Mohammad Thoker, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. Sigma: Sinkhorn-guided masked video modeling. In *European Conference on Computer Vision (ECCV)*, 2025. 5, 6, 8

[50] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5

[51] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 7

[52] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1, 2, 5

[53] Xinyu Sun, Peihao Chen, Liangwei Chen, Changhao Li, Thomas H. Li, Mingkui Tan, and Chuang Gan. Masked motion encoding for self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 5, 6, 8

[54] Fida Mohammad Thoker, Hazel Doughty, and Cees GM Snoek. Skeleton-contrastive 3d action representation learning. In *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, 2021. 2

[55] Fida Mohammad Thoker, Hazel Doughty, Piyush Bagad, and Cees GM Snoek. How severe is benchmark-sensitivity in video self-supervised learning? In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 5, 7, 8

[56] Fida Mohammad Thoker, Hazel Doughty, and Cees G. M. Snoek. Tubelet-contrastive self-supervision for video-efficient generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[57] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2, 3, 5, 6, 8

[58] Shuyuan Tu, Qi Dai, Zuxuan Wu, Zhi-Qi Cheng, Han Hu, and Yu-Gang Jiang. Implicit temporal modeling with learnable alignment for video recognition. *arXiv preprint arXiv:2304.10465*, 2023. 2

[59] Guangting Wang, Yizhou Zhou, Chong Luo, Wenxuan Xie, Wenjun Zeng, and Zhiwei Xiong. Unsupervised visual representation learning by tracking patches in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[60] Jinpeng Wang, Yuting Gao, Ke Li, Yiqi Lin, Andy J Ma, Hao Cheng, Pai Peng, Feiyue Huang, Rongrong Ji, and Xing Sun. Removing the background by adding the background: Towards background robust self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[61] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 2, 6

[62] 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 (CVPR)*, 2022. 6

[63] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Lu Yuan, and Yu-Gang Jiang. Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5, 8

[64] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 2, 7, 8

[65] Fanyi Xiao, Joseph Tighe, and Davide Modolo. Maclr: Motion-aware contrastive learning of representations for videos. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[66] 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 (CVPR)*, 2019. 2

[67] Haosen Yang, Deng Huang, Bin Wen, Jiannan Wu, Hongxun Yao, Yi Jiang, Xiatian Zhu, and Zehuan Yuan. Self-supervised video representation learning with motion-aware masked autoencoders. *arXiv preprint arXiv:2210.04154*, 2022. 2

[68] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition. *arXiv preprint arXiv:2302.03024*, 2023. 2

[69] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[70] Huaidong Zhang, Xuemiao Xu, Guoqiang Han, and Shengfeng He. Context-aware and scale-insensitive temporal repetition counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7

[71] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo

Zhou, Qi Chu, et al. X-paste: Revisiting scalable copy-paste for instance segmentation using clip and stablediffusion. In *International Conference on Learning Representations (ICLR)*, 2023. 4, 5

[72] Long Zhao, Nitesh B Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. Videoprism: A foundational visual encoder for video understanding. *arXiv preprint arXiv:2402.13217*, 2024. 2

[73] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 8