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

## Supplementary Material

In this appendix, we present supplementary analysis and detailed experimental validations. Section A provides an extensive state-of-the-art comparison for finetuning performance on K400 and SSv2 datasets. Section B further evaluates the generalization capability of SMILE on additional downstream video tasks. A detailed comparison against prior CLIP adaptation methods is provided in Section C. Section D shows more results for learning without natural videos. In Section E, we include additional experiments, such as performance under fixed training budgets, extended ablation studies with larger datasets, and qualitative visualizations. Finally, Section F specifies dataset details and clearly outlines the training and evaluation procedures used throughout our experiments.

## A. Extensive Comparison on K400 and SSv2

In the main paper, due to the space limit, we compare only with prior self-supervised methods using the same pretraining setup—specifically, a ViT-B backbone trained for 600 epochs on K400 and 800 on SSv2 datasets. Here, we provide a broader comparison, including self-supervised methods with varying pretraining setups and numerous supervised methods. Results for K400 and SSv2 finetuning are presented in Table 1 and Table 2, respectively.

Our method not only reaches state of the art among all self-supervised methods but also matches or even surpasses many supervised methods that use specialized backbones, such as the hierarchical 3D transformer in MViTv2 [23] (more than +0.2% on K400 and +1.6% on SSv2) and the multi-head relation aggregation in Uniformer [21] (more than +0.1% on K400, +0.9% on SSv2), highlighting its ability to learn superior representations without heavily customized architectures. Additionally, we significantly outperform supervised methods using the same backbone (ViT-B), e.g. TimeSformer [2] by more than +2.4% on K400 and +12.6% on SSv2 and Mformer [30] by more than +3.4% on K400 and +5.4% on SSv2. This demonstrates our method's strength in capturing better spatio-temporal dynamics than these supervised approaches.

Our approach significantly improves on VideoMAE [43] baseline, achieving gains of +3.6% on K400 and +4.9% on SSv2 for ViT-S, when trained for 800 epochs. Similarly, for ViT-B, it achieves an improvement of +3.1% on K400 and +3.6% on SSv2, when trained for 600 epochs on K400. Furthermore, our method outperforms all prior self-supervised approaches under similar pretraining settings, including the same backbone, dataset, and training epochs (e.g., K400 on ViT-B for 600/800 epochs). This includes surpassing methods such as CV-MAE-V [27], which employs contrastive video masked autoencoding, SIGMA [36], which uses Sinkhorn-Guided feature clustering, and OmniMAE [11], which reconstructs from both images and videos. These results underline the effectiveness of our method across diverse self-supervised learning strategies.

The tables also compare our method with approaches that use significantly longer training schedules, such as MVD [46], ST-MAE [10], and MotionMAE [49]. MVD achieves strong performance but relies on a resource-intensive pipeline, involving 1200 epochs of pretraining for both VideoMAE [43] and MAE [14], followed by 400 epochs of distillation, making direct comparisons challenging. Despite training for only 600 epochs, our method surpasses MVD [46] by 0.4% on K400 finetuning and consistently outperforms other methods trained for 1600 epochs on K400 or 2400 epochs on SSv2. This includes motion-aware methods like MotionMAE [49], MME [40], MG-MAE [16], and MGM [8], demonstrating the superior training efficiency of our approach. When trained for 1200 epochs SMILE achieves a boost of 0.3% on both K400 and SSv2 finetuning showing the scalability of our method for longer training schedules.

To summarize, our method surpasses many supervised methods and achieves state-of-the-art performance among video SSL methods while maintaining training efficiency.

## B. Generalization to More Temporal Tasks

In the main paper, we show the generalization capability of our method for diverse downstream settings in SEVERE-benchmark. Here we show the generalization of our method to more video understanding tasks. In particular, we evaluate temporally aware tasks: Unsupervised Video Object Segmentation (**Un-VOS**) and Temporal Action Localization (**TAL**). The goal is to evaluate the motion modeling capability of video representations with **TAL** requiring motion boundary awareness and **Un-VOS** requiring object motion propagation modeling.

### B.1. Unsupervised Video Object Segmentation

**Setup.** We adopt the evaluation approach from [36] to assess the learned temporal and spatial features via unsupervised video object segmentation using the benchmark introduced by [35]. Unlike conventional action recognition benchmarks that aggregate features into a single global representation, this task examines the encoder's capability to generate consistent temporal object segmentation maps. Specifically, extracted space-time features are grouped us-

Table 1. **Detailed comparison with supervised and self-supervised pretraining methods for full finetuning on Kinetics-400 (K400)**. $*$ denotes results obtained by our evaluation. Params denote the number of parameters in millions. Our SMILE outperforms many supervised methods, achieves the best performance among self-supervised methods, and demonstrates a faster convergence.

| Method | Backbone | Epochs | Pretrain | Top-1 | Params |
|---|---|---|---|---|---|
| *supervised* | | | | | |
| Mformer [30] | Mformer-B | - | K400 | 79.7 | 109 |
| VideoSwin [25] | Swin-B | - | K400 | 80.6 | 88 |
| TimeSformer [2] | ViT-B | - | K400 | 80.7 | 430 |
| MViTv1 [9] | MViTv1-B | - | K400 | 80.2 | 37 |
| MViTv2 [23] | MViTv2-B | - | K400 | 82.9 | 52 |
| Uniformer-B [21] | Uformer-B | - | K400 | 83.0 | 50 |
| *self-supervised* | | | | | |
| VideoMAE* [43] | ViT-S | 800 | K400 | 75.9 | 22 |
| VideoMAE [43] | ViT-S | 1600 | K400 | 79.0 | 22 |
| SMILE (ours) | ViT-S | 800 | K400 | **79.5** | 22 |
| VideoMAE [43] | ViT-B | 800 | K400 | 80.0 | 87 |
| VideoMAE [43] | ViT-B | 1600 | K400 | 81.5 | 87 |
| ST-MAE [10] | ViT-B | 1600 | K400 | 81.3 | 87 |
| MVD [46] | ViT-B | 1600+400 | K400 | 82.7 | 87 |
| MotionMAE [49] | ViT-B | 1600 | K400 | 81.7 | 87 |
| CMAE-V [27] | ViT-B | 800 | K400 | 80.2 | 87 |
| CMAE-V [27] | ViT-B | 1600 | K400 | 80.9 | 87 |
| BEVT [45] | ViT-B | 800+150 | K400 | 80.6 | 87 |
| OmniMAE [11] | ViT-B | 800 | K400 | 80.8 | 87 |
| SIGMA [36] | ViT-B | 800 | K400 | 81.5 | 87 |
| MGM [8] | ViT-B | 800 | K400 | 80.8 | 87 |
| MGM [8] | ViT-B | 1600 | K400 | 81.7 | 87 |
| MME* [40] | ViT-B | 800 | K400 | 81.5 | 87 |
| MME [40] | ViT-B | 1600 | K400 | 81.8 | 87 |
| MGMAE [16] | ViT-B | 800 | K400 | 81.2 | 87 |
| MGMAE [16] | ViT-B | 1600 | K400 | 81.8 | 87 |
| SMILE (ours) | ViT-B | 600 | K400 | **83.1** | 87 |
| SMILE (ours) | ViT-B | 1200 | K400 | **83.4** | 87 |

Table 2. **Detailed comparison with supervised and self-supervised pretraining methods for full finetuning on Something-Something V2 (SSv2)**. $*$ denotes results obtained by our evaluation. Params denote the number of parameters in millions. Our SMILE outperforms many supervised methods, achieves the best performance among self-supervised methods, and demonstrates a faster convergence.

| Method | Backbone | Epochs | Pretrain | Top-1 | Params |
|---|---|---|---|---|---|
| *supervised* | | | | | |
| Mformer [30] | Mformer-B | - | K400 | 66.7 | 109 |
| VideoSwin [25] | Swin-B | - | K400 | 69.6 | 88 |
| TimeSformer [2] | ViT-B | - | K400 | 59.5 | 121 |
| MViTv1 [9] | MViTv1-B | - | K400 | 67.7 | 37 |
| MViTv2 [23] | MViTv2-B | - | K400 | 70.5 | 52 |
| Uniformer-B [21] | Uformer-B | - | K400 | 71.2 | 50 |
| *self-supervised* | | | | | |
| OmniMAE [11] | ViT-B | 800 | SSv2 | 69.5 | 87 |
| VideoMAE [43] | ViT-B | 800 | SSv2 | 69.6 | 87 |
| VideoMAE [43] | ViT-B | 2400 | SSv2 | 70.8 | 87 |
| CMAE-V [27] | ViT-B | 800 | SSv2 | 69.7 | 87 |
| CMAE-V [27] | ViT-B | 1600 | SSv2 | 70.5 | 87 |
| MME [40] | ViT-B | 800 | SSv2 | 70.0 | 87 |
| MGM [8] | ViT-B | 800 | SSv2 | 70.6 | 87 |
| MGM [8] | ViT-B | 2400 | SSv2 | 72.1 | 87 |
| SIGMA [36] | ViT-B | 800 | SSv2 | 71.2 | 87 |
| MGMAE [16] | ViT-B | 800 | SSv2 | 71.0 | 87 |
| MGMAE [16] | ViT-B | 2400 | SSv2 | 72.3 | 87 |
| SMILE (ours) | ViT-B | 800 | SSv2 | **72.5** | 87 |
| VideoMAE* [43] | ViT-S | 800 | K400 | 64.2 | 22 |
| SIGMA [36] | ViT-S | 800 | K400 | 68.7 | 22 |
| SMILE (ours) | ViT-S | 800 | K400 | **69.1** | 22 |
| OmniMAE [11] | ViT-B | 800 | K400 | 69.0 | 87 |
| VideoMAE [43] | ViT-B | 800 | K400 | 68.5 | 87 |
| MVD [46] | ViT-B | 1600+400 | K400 | 72.5 | 87 |
| MME [40] | ViT-B | 800 | K400 | 70.5 | 87 |
| SIGMA [36] | ViT-B | 800 | K400 | 71.1 | 87 |
| MGMAE* [16] | ViT-B | 800 | K400 | 68.9 | 87 |
| MGM* [8] | ViT-B | 800 | K400 | 71.1 | 87 |
| SMILE (ours) | ViT-B | 600 | K400 | **72.1** | 87 |
| SMILE (ours) | ViT-B | 1200 | K400 | **72.4** | 87 |

ing k-means clustering with a given number of clusters ($K$), then aligned to ground-truth object masks via the Hungarian algorithm [20]. Segmentation accuracy is quantified through mean Intersection over Union (mIoU). The scenario is labeled as clustering when $K$ equals the actual object count and as overclustering when $K$ surpasses this number. We follow the implementation from [36] and report mIoU on **DAVIS** [31] and **YTVOS** [48].

**Results.** As shown in Table 3, SMILE obtains the best segmentation performance across all settings except for DAVIS clustering where it is the second best. In particular, we outperform prior motion modeling methods MGM and MGMAE by 4% and 6% on YTVOS clustering and by 6% and 7% on YTVOS overclustering. Interestingly, we also beat SIGMA which explicitly clusters the reconstructed features via Sinkhornkoop clustering. This demonstrates the superior motion modeling capability of our approach over standard pixel reconstruction, motion-guided pixel reconstruction, and feature clustering reconstruction approaches.

## B.2. Temporal Action Localization

**Setup.** Temporal action localization (TAL) [24, 52, 55] is a task that aims to identify categories of actions that occur in a video and to locate the start and end timestamps of all action instances. It requires the model to understand not only the spatial semantics within the frames but also the temporal dynamics across frame sequences to capture the action process. We evaluated our SMILE as well as the comparative methods on two representative TAL benchmarks THUMOS-14 [17] and ActivityNet-v1.3 [3]. We used the

| Method | Unsupervised Video Object Segmentation | | | | Temporal Action Localization | |
|---|---|---|---|---|---|---|
| | Clustering | | Overclustering | | | |
| | YTVOS | DAVIS | YTVOS | DAVIS | THUMOS-14 | ActivityNet-v1.3 |
| VideoMAE [43] | 34.1 | 29.5 | 61.3 | 56.2 | 58.5 | 37.3 |
| MGM [8] | 36.6 | **36.5** | 61.2 | 56.6 | 62.0 | 37.6 |
| MGMAE [16] | 34.5 | 31.0 | 60.1 | 57.5 | 56.3 | 37.3 |
| SIGMA [36] | 37.5 | 31.5 | 66.4 | 58.5 | 62.7 | 37.7 |
| SMILE (ours) | **40.5** | 32.7 | **67.0** | **59.5** | **65.6** | **38.0** |

Table 3. **Generalization assessment on Unsupervised Video Object Segmentation and Temporal Action Localization.** All methods are evaluated on the ViT-B backbone pretrained on K400 with their publicly available checkpoints. SMILE outperforms prior masked video modeling works on both tasks demonstrating a better temporal modeling capability for more complex video understanding tasks.

pretrained models from each method as the backbones for video spatio-temporal feature extraction and finetuned them with the TAL method ActionFormer [52] on both datasets using the OpenTAD framework [24]. Following the common practice in the TAL community, we report the average mean average precision (mAP) over various temporal intersection of union (tIoU) values, i.e., 10 tIoU values [0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95] for ActivityNet-v1.3 and 5 tIoU values [0.3, 0.4, 0.5, 0.6, 0.7] for THUMOS-14.

**Results.** As shown in Table 3, our SMILE achieves the highest average mAPs on both benchmarks. Specifically, on THUMOS-14, SMILE outperforms some methods by a large margin *e.g.* VideoMAE by 7% and MGMAE by 9%. On the more challenging and largescale dataset ActivityNet-v1.3, it shows 0.3% improvement over the second-best SIGMA. More notably, on ActivityNet-v1.3, the performance of SMILE is on par with the state-of-the-art TAL performance, which relies on fully supervised finetuning with labeled Kinetics-400 videos (not shown in the table). Overall, the results show that SMILE generalizes better to more complex downstream video tasks than the current masked video modeling approaches.

## C. More comparisons with CLIP adaptations

In this section, we provide further comparisons between our proposed SMILE and existing CLIP adaptation methods for action recognition. Specifically, we consider two distinct categories of CLIP-based adaptations: ***without intermediate pretraining*** and ***with intermediate pretraining***. The former approaches either jointly finetune the CLIP vision and text encoders using labeled video-text pairs of the target dataset (e.g., X-CLIP [28], ViFi-CLIP [33], ILA [44]) or directly finetune the CLIP vision encoder with added specialized spatio-temporal adaptation modules on labeled videos from the target dataset (e.g., AIM [50], DUAL Path [29]). In contrast, intermediate pretraining methods, such as UMT [22] and ViCLIP [47] use the CLIP model and intermediate video-text pairs for further pretraining to align video and text modality using contrastive learning. Detailed results of these comparisons are presented in Table 4.

We observe that SMILE significantly outperforms prior adaptation methods that directly finetune the CLIP model on the target dataset. It achieves the best performance on all target datasets except K400, where it achieves comparable results. This highlights the effectiveness of our approach as a superior CLIP adaptation strategy which relies only on the unlabeled videos for adaptation. Moreover, as demonstrated in the main paper, SMILE surpasses adaptation methods that also employ intermediate pretraining, UMT, and ViCLIP. Notably, the performance gains are particularly pronounced on motion-intensive datasets like GYM99, SSv2, and EPIC, underscoring the critical importance of explicit motion modeling an aspect often neglected in existing CLIP adaptations. Our intuition is that CLIP features contain object information like shape, boundaries, and location, which guides the reconstruction task to focus on the overlaid objects as well as the original video semantics.

In summary, SMILE provides a robust CLIP adaptation by reconstructing masked video inputs directly within the CLIP visual feature space while explicitly integrating synthetic motion cues.

## D. Learning without Natural Videos

In the main paper, we show that our method can learn video representations by overlaying object motions on clips from natural videos, single frames from natural videos, single natural images, or even black and noise images. This raises the question about the effectiveness of using object motions alone and how they compare with learning from natural videos. To answer this we compare the performance of learning from object motions only with learning from natural video data. Specifically, we generate video clips by adding our synthetic object motions to randomly generated noise images. As in the main paper, we take a noise image, duplicate it $T$ times to form a static video clip, and then

| | | Finetuning | | | | | Linear Probing | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | Intermediate Dataset | K400 | UCF | GYM | SSv2 | EPIC | UCF | GYM | SSv2 | EPIC |
| CLIP [32] | - | 81.8 | 93.6 | 88.0 | 66.7 | 50.3 | 77.5 | 20.7 | 11.3 | 25.1 |
| ***Without intermediate pretraining*** | | | | | | | | | | |
| X-CLIP [28] | - | 83.8 | 92.0 | 75.2 | 57.4 | 52.7 | - | - | - | - |
| ViFi-CLIP [33] | - | 83.9 | 94.6 | 81.5 | 48.6 | 48.9 | - | - | - | - |
| AIM [50] | - | 83.9 | 94.0 | 90.3 | 66.4 | 58.4 | - | - | - | - |
| ILA [44] | - | 84.0 | 94.2 | 82.7 | 65.0 | 60.8 | - | - | - | - |
| DUAL-Path [29] | - | **85.4** | - | - | 69.6 | - | - | - | - | - |
| ***With intermediate pretraining*** | | | | | | | | | | |
| ViCLIP [47] | Intervid-10M [47] | 82.4 | 95.2 | 89.7 | 67.9 | 55.0 | 86.7 | 27.3 | 18.9 | 27.3 |
| UMT [22] | K700 [4] | 81.7 | 96.0 | 89.9 | 70.1 | 50.1 | **88.0** | 26.4 | 18.8 | 28.2 |
| SMILE (ours) | K400 [18] | 83.1 | **96.4** | **90.8** | **71.9** | **63.3** | 83.8 | **30.2** | **23.7** | **34.6** |

Table 4. **More comparison with CLIP adaptations.** SMILE learns better video representations than both types of CLIP adaptations: with and without intermediate pretraining. The performance gap is wider on motion-focused domains.

overlay objects with motion on top of it. We compare its performance with learning from natural videos of the K400 dataset. We use ViT-S and ViT-B backbones for this experiment and results are shown in Table 5.

We observe that learning with such augmented videos significantly improves on no pretraining. Compared to the VideoMAE baseline which uses 240K natural videos of K400, learning from our object motions only with pixel reconstruction shows a small gap in performance. This highlights the effectiveness of video representations learned only from the proposed object motions via unnatural videos created on the fly. The gap is further reduced when feature reconstruction is used instead of pixel reconstruction, demonstrating the impact of using CLIP feature projections over raw pixels, even for such unnatural data. Overall, our proposed synthetic object motions can act as a strong supervisory signal in a standalone to learn video representations with masked video modeling. We leave the scaling of such learning without natural videos for larger models to future works.

Table 5. **Learning video representations with only object motions.** We train VideoMAE baseline on Kinetics-400 videos and ours with augmented clips generated by overlaying noise images with object motions. All settings train a ViT-S for 400 and ViT-B for 600 epochs. Our method trained without any natural videos lags only by a small margin compared to the VideoMAE baseline trained with natural videos from the Kinetics-400 dataset.

| Method | Data | Target | K400 | SSv2 | GYM |
|---|---|---|---|---|---|
| ViT-S | | | | | |
| No Pretrain. | - | - | 65.7 | 52.2 | 55.1 |
| Ours | Noise + Motion | Pixel | 72.7 | 59.1 | 72.5 |
| Ours | Noise + Motion | Features | 73.7 | 61.0 | 74.6 |
| VideoMAE | K400 | Pixel | 75.9 | 62.7 | 75.1 |
| ViT-B | | | | | |
| No Pretrain. | - | - | 69.1 | 49.8 | 50.0 |
| Ours | Noise + Motion | Pixel | 74.5 | 60.0 | 77.2 |
| Ours | Noise + Motion | Features | 77.5 | 64.1 | 83.0 |
| VideoMAE | K400 | Pixel | 79.0 | 67.0 | 86.6 |

## E. Additional Experiments

### E.1. Performance with fixed training budget

We now compare our method with VideoMAE [43] baseline for different dataset scales with a fixed training budget, *i.e.* total number of training iterations $n = s \times e$, where $s$ is the dataset size and $e$ is the number of epochs. Using the ViT-S backbone, we pretrain on four subsets of K400, namely, 12.5%, 25%, 50%, and 100% of the full scale. For the whole dataset $s = 100\%$, we pretrain for 200 epochs; smaller subsets are trained for proportionally more epochs *i.e.*, $s = 50\%$ for 400 epochs, $s = 25\%$ for 800, and $s = 12.5\%$ for 1600. Results in Figure 1 show that our method consistently outperforms the VideoMAE baseline by a large margin for various data scales under the same training budget. This highlights the robustness of our method to dataset size and training duration.

Table 6. **Full-Scale ablation.** Ablating our main contributions on a larger backbone and a bigger pretraining dataset, i.e., the original K400. Reconstructing features and adding synthetic motions shows consistent improvements for a larger backbone (ViT-B) and scaling to a bigger pretraining dataset.

| Backbone | Target | Synth. | K400 | SSv2 |
|---|---|---|---|---|
| ViT-B | Pixels | w/o | 78.3 | 67.2 |
| ViT-B | Pixels | w/ | 78.9 | 67.7 |
| ViT-B | Features | w/o | 81.2 | 70.8 |
| ViT-B | Features | w/ | **81.7** | **71.2** |

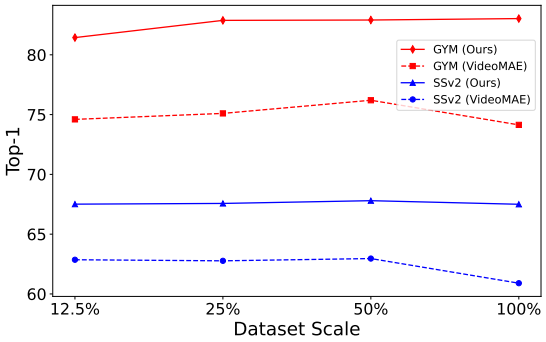


Figure 1. **Performance comparison with a fixed training budget.** We evaluate on SSv2 and GYM for full finetuning. Our method consistently outperforms VideoMAE [43] across all data scales with the same training budget.

### E.2. Full-scale ablation

All main paper ablations are conducted with the smaller $K400_m$ pretraining and a ViT-S backbone. To reinforce the validity of our key contributions—feature target reconstruction and synthetic object motion, we extend the ablations to full-scale K400 pretraining using a larger ViT-B backbone. Full finetuning is performed on the complete K400 and SSv2 datasets. We adopt the best configurations from the small-scale ablations, including 80% masking, trajectory masking, two object overlays, CLIP feature reconstruction, and a 300-epoch training schedule unless stated otherwise.

Table 6 presents the results for full-scale ablations. Consistent with the small-scale ablations in the main paper, incorporating synthetic motions boosts downstream performance. Specifically, our feature reconstruction improves the downstream performance by 2.9% on K400 and 3.6% on SSv2 over pixel reconstruction. By adding object motions, pixel reconstruction improves by 0.6% on K400 and 0.5% on SSv2, and feature reconstruction sees gains of 0.5% on K400 and 0.4% on SSv2. Our best configuration—feature reconstruction with synthetic motion—achieves the highest performance, reinforcing the robustness and scalability of our method across larger backbones and pretraining datasets.

### E.3. Qualitative analysis

In Figure 2, we extend the qualitative analysis to compare with more prior video SSL works. As before, we observe that the features of different frames have larger differences for our model, indicating better temporal awareness. In particular, our feature similarity is consistent with moth motion-aware methods like MGM, MGMAE and MME demonstrating the motion focus of our method too.

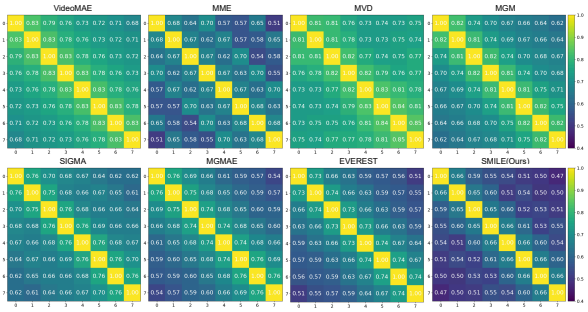


Figure 2. **Feature similarity across different frames for different SSL methods.** We compute this on K400 validation videos.

## F. Experimental Details

### F.1. Datasets for main results

For linear probing and full finetuning SoTA experiments, we evaluate action recognition task with standard action recognition datasets *i.e.* Kinetics-400 [18] (K400), SomethingSomething V2 [12] (SSv2), UCF-101 [39] (UCF), HMDB-51 [19] (HMDB), FineGYM [37] (GYM), and EPIC-Kitchens-100 [6] (EPIC). More details about the datasets are in Table 8 and following:

**Kinetics-400 [18]** Kinetics-400 (K400) is a comprehensive benchmark designed for video action recognition tasks. It consists of over 306,000 concise video clips sourced from YouTube, spanning an impressive 400 distinct action categories. As one of the largest and most widely adopted datasets in this field, K400 plays a pivotal role in assessing and advancing cutting-edge models for understanding actions in video content.

**SomethingSomething V2 [12]** SomethingSomething V2 (SSv2) is a collaboratively sourced dataset consisting of first-person video recordings, specifically crafted to facilitate the development of common-sense reasoning capabilities. In terms of visual composition and perspective, it markedly diverges from Kinetics-400. The dataset comprises 168,913 training samples and 24,777 testing samples, distributed across 174 unique action categories.

**UCF-101 [39]** UCF-101 is a widely recognized benchmark in video self-supervised learning research. It comprises a diverse set of 9537 training and 3783 testing samples, sourced from YouTube videos grouped into 101 action categories, characterized by coarse granularity. Many of these categories show a substantial overlap with the action types included in Kinetics-400.

**HMDB-51 [19]** HMDB-51 (HMDB) is a widely-used benchmark for action recognition research. It features a total of 6,766 video clips, carefully selected from a variety of sources, such as films, the Prelinger Archive, YouTube, and Google Videos. The dataset is categorized into 51 unique action classes, with each class comprising no fewer than

Table 7. **SEVERE benchmark.** Details of all the experimental subsets in the benchmark. We follow the configurations from the original work [42].

| Evaluation Setup | Experiment | Dataset | Task | #Classes | #Finetuning | #Testing | Eval Metric |
|---|---|---|---|---|---|---|---|
| | Gym99 | FineGym [37] | Action Class. | 99 | 20,484 | 8,521 | Top-1 Acc. |
| **Sample Efficiency** | UCF ($10^3$) | UCF 101 [39] | Action Class. | 101 | 1,000 | 3,783 | Top-1 Acc. |
| | Gym ($10^3$) | FineGym [37] | Action Class. | 99 | 1,000 | 8,521 | Top-1 Acc. |
| **Action Granularity** | FX-S1 | FineGym [37] | Action Class. | 11 | 1,882 | 777 | Mean-per-class |
| | UB-S1 | FineGym [37] | Action Class. | 15 | 3,511 | 1,471 | Mean-per-class |
| **Task Shift** | UCF-RC | UCFRep [54] | Repetition Counting | - | 421 | 105 | Mean Error |
| | Charades | Charades [38] | Multi-label Class. | 157 | 7,985 | 1,863 | mAP |

Table 8. **Datasets.** Details of the datasets used for evaluation showing the corresponding number of classes, training, and testing samples for each.

| Dataset | #Classes | #Train | #Test |
|---|---|---|---|
| K400 | 400 | 240K | 19K |
| UCF | 101 | 9.5K | 3.8K |
| HMDB | 51 | 4.8K | 2K |
| SSv2 | 174 | 169K | 24.8K |
| GYM | 99 | 20.5K | 8.5K |
| EPIC | 97 | 67.2K | 9.7K |

Table 9. **Linear-Evaluation setting.**

| config | K400 | UCF | HMDB | SSv2 | GYM | EPIC |
|---|---|---|---|---|---|---|
| optimizer | AdamW[26] | | | | | |
| base learning rate | 1.e-3 | | | | | |
| weight decay | 0.05 | | | | | |
| optimizer momentum | $\beta_1, \beta_2 = 0.9, 0.999$ | | | | | |
| layer-wise lr decay [1] | 0.75 | | | | | |
| batch size | 128 | | | | | |
| learning rate schedule | cosine decay | | | | | |
| training epochs | 30 | 100 | 100 | 50 | 100 | 100 |
| flip augmentation | *yes* | *yes* | *yes* | *no* | *yes* | *yes* |

Table 10. **Full finetuning evaluation setup.**

| config | SSv2 | K400 | SEVERE |
|---|---|---|---|
| optimizer | AdamW | | |
| base learning rate | 1.0e-3 | | |
| weight decay | 0.05 | | |
| optimizer momentum | $\beta_1, \beta_2 = 0.9, 0.999$ | | |
| layer-wise lr decay[1] | 0.75 | | |
| batch size | 32 | 16 | 16 |
| learning rate schedule | cosine decay | | |
| warmup epochs | 5 | | |
| training epochs | 40 | 100 | 100 |
| flip augmentation | *no* | *yes* | *yes* |
| RandAug [5] | (9,0.5) | | |
| label smoothing[41] | 0.1 | | |
| mixup [53] | 0.8 | | |
| cutmix [51] | 1.0 | | |
| drop path | 0.1 | | |

100 video samples.

**FineGYM [37]** FineGYM (GYM) is a benchmark designed for fine-grained action analysis in gymnastics competitions. For our study, we specifically select the Gym-99 subset, which consists of 99 unique action categories. This subset provides 20,484 training samples and 8,521 testing samples.

**EPIC-Kitchens-100 [6]** EPIC-Kitchens-100 (EPIC) is a large egocentric dataset capturing daily kitchen activities, annotated with 97 verbs and 300 nouns, where actions are defined as combinations of both. Similar to SS-v2, EK-100 differs significantly from Kinetics-400 in its unique visual style and first-person perspective. Using the standard splits provided in [13], it includes 67,217 training samples and 9,668 for validation, with our study focusing solely on recognizing the 97 verb classes.

## F.2. Datasets for SEVERE Benchmark

**SEVERE Benchmark[42]** SEVERE-Benchmark spans eight experimental settings across four datasets *i.e.* Something-Something V2, UCF, FineGYM, and Charades [38]. Table 7 provides detailed configurations for each subset.

## F.3. Training and Evaluation Details

**Pretraining details.** We pretrain on Kinetics-400 (K400) [18] and Something-Something V2 (SSv2) [12] datasets. To generate our segmented object set $O$, we follow [7] to utilize Stable Diffusion [34] and X-paste [56], generating 60 samples for each of the 1203 categories in the LVIS dataset [13]. Following VideoMAE [43] we use a temporal stride of 2 for SSv2 and a stride of 4 for K400. Each clip contains 16 frames sampled at a resolution of $224 \times 224$ pixels. Space-time tube embeddings are extracted using a 3D convolution layer, treating each $2 \times 16 \times 16$ cube as a token. We use both tube and trajectory mask-

Table 11. **Pretraining details.**

| config | SSv2 | K400 |
|---|---|---|
| optimizer | AdamW | |
| base learning rate | 1.5e-4 | |
| weight decay | 0.05 | |
| optimizer momentum | $\beta_1, \beta_2 = 0.9, 0.95$ | |
| batch size | 256 | |
| learning rate schedule | cosine decay | |
| warmup epochs | 40 | |
| flip augmentation | *no* | *yes* |
| augmentation | MultiScaleCrop | |
| Epochs | 800 | 600 |

ing with a ratio $m = 80\%$. We employ multiple sampling based on [15] during the pretraining which effectively samples two input clips from the same video for reconstruction. This decreases the training time by almost half without any performance drop. We always count epochs as "effective epochs = *No. of epochs* $\times$ *No. of samples per video*", i.e., how many times each video is sampled and processed throughout training. We employ our progressive pretraining strategy for 600 epochs on K400 and 800 on SSv2, 300 and 400 in each stage, respectively. Table 11 shows the rest of the configuration. We train our models with 8 NVIDIA V100 GPUs. For downstream evaluation, we only use the student network without its decoder and attach a task-dependent head to the pretrained student encoder e.g., a classification layer for action recognition.

**Linear Probing details.** Table 9 shows the settings for linear probing. We use 4 NVIDIA V100 GPUs for linear probing.

**Full Finetuning details.** Table 10 shows the settings for full finetuning, following [43]. We use 4 NVIDIA V100 GPUS for fine-tuning.

**SEVERE Benchmark evaluation.** We compare our method to recent masked video modeling approaches, using the SEVERE codebase [42] and keeping identical training and evaluation setups for fair comparison. Official models for each comparative method are used.

## References


[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2022. 6

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 1, 2

[3] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[4] 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. 4

[5] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *arXiv preprint arXiv:1909.13719*, 2019. 6

[6] 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. 5, 6

[7] 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. 6

[8] 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

[9] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[10] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022. 1, 2

[11] 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. 1, 2

[12] 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. 5, 6

[13] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[14] 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. 1

[15] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of*

*the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138, 2020. 7

[16] 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

[17] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 2

[18] 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. 4, 5, 6

[19] 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. 5

[20] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2

[21] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022. 1, 2

[22] 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. 3, 4

[23] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[24] Shuming Liu, Chen Zhao, Fatimah Zohra, Mattia Soldan, Alejandro Pardo, Mengmeng Xu, Lama Alssum, Merey Ramazanova, Juan León Alcázar, Anthony Cioppa, Silvio Giancola, Carlos Hinojosa, and Bernard Ghanem. Opentad: A unified framework and comprehensive study of temporal action detection. *arXiv preprint arXiv:2502.20361*, 2025. 2, 3

[25] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019. 6

[27] Chengze Lu, Xiaojie Jin, Zhicheng Huang, Qibin Hou, Ming-Ming Cheng, and Jiashi Feng. CMAE-V: contrastive masked autoencoders for video action recognition. *CoRR*, abs/2301.06018, 2023. 1, 2

[28] 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. 3, 4

[29] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Dual-path adaptation from image to video transformers. *arXiv preprint arXiv:2303.09857*, 2023. 3, 4

[30] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2

[31] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 2

[32] 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. 4

[33] 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. 3, 4

[34] 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. 6

[35] Mohammadreza Salehi, Efstratios Gavves, Cees G. M. Snoek, and Yuki M Asano. Time does tell: Self-supervised time-tuning of dense image representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[36] 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. 1, 2, 3

[37] 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. 5, 6

[38] 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. 6

[39] 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. 5, 6

[40] Xinyu Sun, Peihao Chen, Liangwei Chen, Changhao Li, Thomas H. Li, Mingkui Tan, and Chuang Gan. Masked motion encoding for self-supervised video representation learn-

ing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015. 6

[42] 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. 6, 7

[43] 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, 4, 5, 6, 7

[44] 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. 3, 4

[45] 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. 2

[46] 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. 1, 2

[47] 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. 3, 4

[48] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 2

[49] 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. 1, 2

[50] 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. 3, 4

[51] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *arXiv preprint arXiv:1905.04899*, 2019. 6

[52] Chenlin Zhang, Jianxin Wu, and Yin Li. ActionFormer: Localizing moments of actions with transformers. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 3

[53] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2018. 6

[54] 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. 6

[55] Chen Zhao, Shuming Liu, Karttikeya Mangalam, and Bernard Ghanem. Re$^2$TAL: Rewiring pretrained video backbones for reversible temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[56] 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. 6