Motion-Guided Masking for Spatiotemporal Representation Learning

David Fan[1] Jue Wang[1] Shuai Liao[2] Yi Zhu[3] Vimal Bhat[1]
Hector Santos-Villalobos[1] Rohith MV[1] Xinyu Li[1]
[1] Amazon Prime Video [2] Amazon Fulfillment Technology [3] AWS AI Labs
{fandavi, juewangn, uliaoshu, yzaws, vimalb, hsantosv, kurohith, xxnl}@amazon.com

Abstract

Several recent works have directly extended the image masked autoencoder (MAE) with random masking into video domain, achieving promising results. However, unlike images, both spatial and temporal information are important for video understanding. This suggests that the random masking strategy that is inherited from the image MAE is less effective for video MAE. This motivates the design of a novel masking algorithm that can more efficiently make use of video saliency. Specifically, we propose a motion-guided masking algorithm (MGM) which leverages motion vectors to guide the position of each mask over time. Crucially, these motion-based correspondences can be directly obtained from information stored in the compressed format of the video, which makes our method efficient and scalable. On two challenging large-scale video benchmarks (Kinetics-400 and Something-Something V2), we equip video MAE with our MGM and achieve up to +1.3% *improvement compared to previous state-of-the-art methods. Additionally, our MGM achieves equivalent performance to previous video MAE using up to* 66% *fewer training epochs. Lastly, we show that MGM generalizes better to downstream transfer learning and domain adaptation tasks on the UCF101, HMDB51, and Diving48 datasets, achieving up to* +4.9% *improvement compared to baseline methods.*

1. Introduction

Video transformers [1, 24, 25, 28, 43] have achieved state-of-the-art for a variety of video understanding tasks, mirroring the success of image transformers such as ViT [9]. However, many video transformers heavily rely on large-scale supervised pretraining from image datasets such as ImageNet21K [8] and JFT-300M [35], which is data-inefficient.

Self-supervised learning (SSL) [5, 15, 17, 29, 42] is one promising paradigm for eliminating the dependency on large-scale supervised pretraining. The denoising / masked autoencoder (MAE) [41], which was originally popularized by BERT [20] for natural language modeling, has recently re-

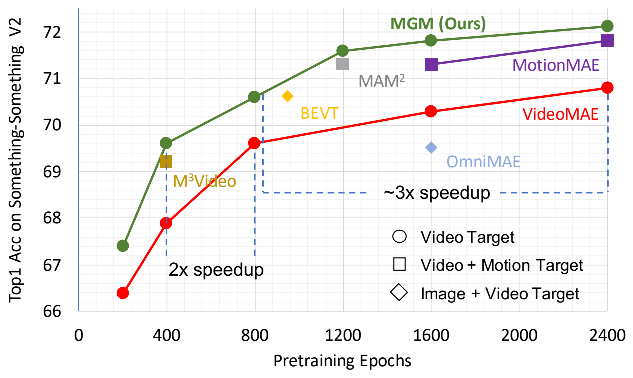


Figure 1: Top-1 accuracy on Something-Something V2. Our proposed MGM outperforms all previous masked-video methods (M^{3}Video [36], MAM^2 [33], BEVT [45], OmniMAE [13], MotionMAE [50], and VideoMAE [39]) with improved training efficiency. Marker shape denotes reconstruction target.

emerged as a promising representation learning method for vision tasks. Image MAE [2, 16] has achieved state-of-the-art in image domain via learning to reconstruct randomly masked image patches. Recently, similar works [11, 39] that reconstruct randomly masked video have obtained encouraging results in video domain. However, few SSL works focus on effectively learning video saliency. In this paper, we improve upon the previous video MAEs [11, 39] via video saliency.

We argue that the random masking strategy which is inherited from image MAE is not optimal for video. The optimal random masking ratio for video MAE [11, 39] is higher than that for image MAE [16] (0.9 vs. 0.75). This can be understood as a consequence of the natural temporal coherence within videos, which leads to the existence of similar video patches in other frames. When using random masking, many of these correlated video patches may be visible to the encoder which would make reconstruction easier. This necessitates the use of a high masking ratio in order to reduce redundancy and make the reconstruction task sufficiently challenging [11]. However, increasing the masking ratio has the side-effect of leaving fewer visible patches for the MAE encoder to learn spatiotemporal saliency from, which we hypothesize limits the learned representation.

We thus propose to guide the model to learn to reconstruct the most salient regions of video. As humans and objects are key to understanding video, one natural idea is to track the bounding boxes in each video frame and mask the content within. This is also consistent with the observation that video pixels evolve continuously frame-by-frame, and therefore, the MAE should be able to reconstruct this spatiotemporal continuity. However, generating bounding boxes for each frame is impractical for large-scale video datasets.

To this end, we hypothesize that motion is an effective guide for detecting spatiotemporal saliency. To test this hypothesis, we define the saliency score as r_{bbox} / r_{non_bbox}, or the ratio of average motion magnitude within bounding boxes to average motion magnitude outside bounding boxes. The saliency score is 1.47 for Something-Something v2 [14] and 1.28 for Kinetics-400 [19] respectively, suggesting that regions of higher motion overlap with spatially salient regions more often than regions of lesser motion.

We thus propose an improved masking algorithm, MGM, which uses motion to continuously guide the mask to cover the most salient spatiotemporal regions. MGM obtains cheap motion correspondence by exploiting the H.264 codec [30] which prevails in popular video formats such as MP4. During the encoding phase of H.264, motion vectors are precomputed and stored as part of the codec. Thus during the decoding phase, motion vectors can be obtained "for free" along with RGB frames. The use of readily available motion vectors instead of expensive optical flow enables us to efficiently achieve scale. While some supervised works have used motion vectors [21, 43, 47], we are the first to use motion vectors for MAE pretraining.

Our MGM outperforms previous VideoMAE [39] by 1.2% on Something-Something V2 (SSv2) [14] and 0.2% on Kinetics-400 (K400) [19], demonstrating that our MGM is effective. MGM can also achieve the same performance as [39] with 50% fewer training epochs as shown in Fig. 1, further making MAE pretraining more data-efficient. We also show that MGM generalizes well to small datasets (UCF101 [34], HMDB51 [22], and Diving48 [23]) in both full finetune and linear probe evaluation as well as domain adaptation settings with up to 4.9% performance improvement over VideoMAE [39]. This shows that the features learned by MGM contain richer semantics that transfer well to video recognition tasks. In summary, our contributions are:

1. MGM, an efficient and effective self-supervised algorithm for 3D masking that continuously models motion trajectories.
2. Applying motion vectors which are directly available during video decoding - unlike optical flow - to provide efficient motion guidance in the MAE framework.
3. New state-of-the-art or comparable results on two large-scale datasets and various downstream tasks on three small datasets, as well as detailed ablations and insights.

2. Related Works

Image Masked Auto-Encoders. iGPT [4] was one of the first recent attempts to revisit masked image modeling using Transformer architecture rather than CNNs. iGPT reconstructs at the pixel-level. Other recent works try different reconstruction targets. Some works such as BEIT [2] explore using tokenization as a prediction target, such as tokens from pretrained dVAE [40]. MaskFeat [46] explores using HoG features [7] as a reconstruction target. ImageMAE [16] reconstructs normalized image patches and focuses on studying decoder design, discovering that larger decoders enable the encoder to process only visible patches, thus enabling the use of a larger masking ratio to achieve both better performance and computational efficiency. Unlike these works, our MGM is designed specifically for video.

Video Masked Auto-Encoders. Some works such as VIMPAC [37] and BEVT [45] try to reconstruct video tokens and thus require strong tokenizers using either extra pretraining data or domain-specific knowledge. VideoMAE [11, 39] bypasses this requirement by applying image MAE [16] to reconstruct 3D RGB video patches, achieving promising results on video benchmarks. Other works such as OmniMAE [13] use both image and video masked modeling to achieve improved performance on both image and video tasks. A line of very recent work tries to improve VideoMAE by designing explicit motion cues by reconstructing optical flow or frame difference based targets [33, 36, 50].

These existing video MAE works all inherit random masking from image MAE [16] where the mask is either discontinuous such as in ST-MAE [11] or random tubes such as in VideoMAE [39]. In the former, the mask is randomly distributed both spatially and temporally, and in the latter, the mask is randomly distributed spatially but temporally static. In contrast, we re-examine this assumption and design a new masking algorithm that continuously masks out video motion to force the model to focus on spatiotemporal saliency. We make no changes to the reconstruction target nor model architecture.

VideoMAE [39] initially generates a random mask and statically propagates the mask across time. The position of the mask is the same across frames, forming small tubelets which are spatially discontinuous but temporally continuous.

ST-MAE [11] independently generates a random mask per-frame which is neither spatially nor temporally continuous.

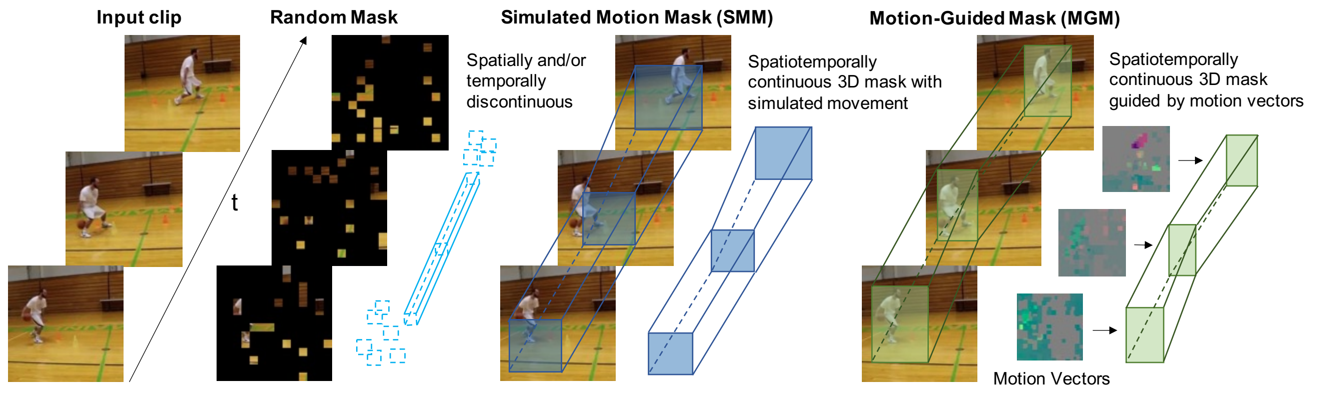

Input clip
Random Mask
Simulated Motion Mask (SMM)
Motion-Guided Mask (MGM)
t
Spatially and/or temporally discontinuous
Spatiotemporally continuous 3D mask with simulated movement
Spatiotemporally continuous 3D mask guided by motion vectors
Motion Vectors

Figure 2: An overview of a) the proof-of-concept simulated-motion masking (SMM) vs. b) motion-guided masking (MGM). Our MGM produces 3D masks that achieves both spatial and temporal continuity, while capturing the true motion information in the video. Random masking (left-most) is both spatially and temporally discontinuous.

3. Methodology

As detailed in Sec. 1, motion is concentrated in salient video regions. Random masking does not effectively cover these regions because motion is not uniformly distributed throughout the video. Furthermore, spatially salient entities such as people and objects are continuous rather than disjoint patches. In general, this leads us to believe random masking may miss certain semantics that are useful for learning spatiotemporal representations. The goal of this work is to address these deficiencies to make masked video modeling more effective for video representation learning. Because video is temporally cohesive, the MAE should be able to reconstruct a continuous spatiotemporal volume comprised of the salient regions. This motivates our following approach.

3.1. Preliminaries

Given a video $V \in \mathbb{R}^{T\times H\times W\times C}$, where T, H, W, C denote the number of frames, height, width, and RGB-channels, the video is typically first split into patches of size $t \times h \times w \times C$ and processed with a patch embedding layer $\mathcal{P}$ to obtain a sequence of token embeddings V_p.

$$V_p = \mathcal{P}(V); V_p \in \mathbb{R}^{\frac{T}{t}\times\frac{H}{h}\times\frac{W}{w}} \tag{1}$$

Next, a masking function η (e.g. random masking [11, 39]) generates a binary mask M which is applied to select a set of visible patches with mask ratio γ.

$$\begin{aligned} M &= \eta(V_p, \gamma);\ dim(M) = dim(V_p) \\ V_{\text{p\_visible}} &= V_p \cdot (\sim M) \\ V_{\text{p\_masked}} &= V_p \cdot M \end{aligned} \tag{2}$$

The encoder ϕ then processes only the visible patches $V_{\text{p\_visible}}$ while the decoder ξ processes the full set of encoded patches and masked tokens $\phi(V_{\text{p\_visible}}) \cup V_{\text{p\_masked}}$ to reconstruct the video. This work uses the same asymmetric encoder-decoder design as [11, 16, 39].

$$E = \phi(V_{\text{p\_visible}}) \tag{3}$$

$$V' = \xi(E \cup V_{\text{p\_masked}}) \tag{4}$$

Finally, the model is trained with MSE reconstruction loss between V and V'. The model is then transferred to downstream tasks such as classification via finetuning with cross-entropy loss. Note that this work focuses on the mask generator $\eta(\gamma)$, which is complementary to recent works which look into the loss function and reconstruction targets [33, 36, 45, 50].

3.2. Simulated-Motion Masking (SMM)

To help the encoder learn spatiotemporal semantics throughout the video, we propose to generate a spatiotemporally continuous moving 3D mask, as shown in Fig. 2. This differs from random masking [11] and random tube masking [39] which break spatiotemporal continuity.

We define the mask as a vector $[x, y, w, h]$ where x and y define the top-left coordinate and w and h define the width and height of the mask. We first explore the use of random motion to generate a dense moving 3D mask with masking ratio γ. To do this, we initialize a rectangle shaped mask in the first frame with dimensions $w_0 = h_0 = \sqrt{\gamma * \frac{H}{h} * \frac{W}{w}}$. x_0 and y_0 are then randomly initialized in $rand(0, \frac{H}{h} - h_0)$. We then propagate the mask across time using the following recurrent relation:

$$\begin{aligned} [x_t, y_t, w_t, h_t] =& [x_{t-1} + v_{xt}, y_{t-1} + v_{yt}, \\ & w_{t-1} + s_t, h_{t-1} + s_t] \\ t \in [1, T),& \sum_t s_t = 0 \end{aligned} \tag{5}$$

where v_{xt}, v_{yt} are the random horizontal and vertical velocity components, and s_t is a random scaling factor. To ensure that the masking ratio for each video is consistent, we enforce that $\sum_t s_t = 0$; in other words, the number of masked patches remains fixed for a given γ.

3.3. Motion-Guided Masking (MGM)

While SMM leads to an improvement over random masking, SMM generates masks that may not reflect true motion as it is not context-aware. For example in Figure 2, the mask may cover static background rather than the basketball player depending on the random initialization and velocity. The most straightforward approach to address this issue would be to guide the mask using precomputed object bounding boxes or optical flow. However, this requires additional data and cost and would make such a method less scalable.

We propose to use motion vectors as guidance to ensure that the mask movement consistently covers motion across frames. Motion vectors [30, 47] can be decoded from raw video (with H.264 or H.265 codecs) with no extra cost, which ensures that our proposed approach adds no additional cost and scales well to larger datasets. A motion vector $M_t[x, y]$ for a given pixel position x, y at frame t, is defined as $[d_x, d_y]$ where d_x and d_y denote the displacement of pixel x, y from frame $t-1$. Motion vectors are typically sparsely assigned to 8×8 pixel grids [30, 47], so the overall motion vector map is similar to optical flow but at a lower resolution. In order to use motion vectors to guide the masking, we need pixel-level correspondence between the motion vectors and the video. We accomplish this by upsampling the motion vector map to match the spatial resolution of the video.

Given a motion vector map $M_t \in \mathbb{R}^{T \times \frac{H}{8} \times \frac{W}{8}}$, we first use nearest-neighbor upsampling (U_{nearest}) to get $M_{\text{t\_scaled}} \in \mathbb{R}^{T \times H \times W}$.

$$\mathrm{M}_{\text{t\_scaled}} = U_{\text{nearest}}(M_t, (H, W)) \tag{6}$$

Next, we initialize a dense rectangle mask for frame t_0 with dimensions $w_0 = h_0 = \sqrt{\gamma * \frac{H}{h} * \frac{W}{w}}$. x_0 and y_0 are then randomly initialized in $rand(0, \frac{H}{h} - h_0)$. The mask is then generated for following time-steps as:

$$\begin{aligned} x_{\text{t\_center}}, y_{\text{t\_center}} &= Pool_{\text{topK}}(M_{\text{t\_scaled}}) \\ w_t &= w_{t-1} + s_{xt} \\ h_t &= h_{t-1} + s_{yt} \\ [x_t, y_t, w_t, h_t] &= [x_{\text{t\_center}} - \frac{w_t}{2}, y_{\text{t\_center}} - \frac{y_t}{2}, w_t, h_t] \\ t \in [1, T), \sum_t s_{xt} &= 0, \sum_t s_{yt} = 0 \end{aligned} \tag{7}$$

where $Pool_{\text{topK}}$ is the topK (top-1 in our work) magnitude pooling operation that gives the argmax indexes $x_{\text{t\_center}}, y_{\text{t\_center}}$, which the mask is centered around. We also apply jitter s_{xt}, s_{yt} to both the mask height and width.

In other words, MGM produces a moving 3D mask where the movement of the mask is centered around parts of the video with the highest magnitude of motion, as shown in Fig. 2. This makes the reconstruction task more focused on spatiotemporal semantics than random masking.

4. Experimental Results

4.1. Datasets

We conduct experiments on five commonly used datasets: **Something-Something V2 (SSv2)** [14] contains 220K videos with 174 action classes. SSv2 is considered a motion heavy dataset, as most of the labels are defined by the motion and directionality of the actual action. **Kinetics-400 (K400)** [19] is the de-facto standard dataset used to evaluate video recognition. It contains 240K Internet videos with 400 action classes. **UCF101** [34] is a dataset containing 13K Internet short videos with 101 action classes. **HMDB51** [22] is a dataset containing 5K short movie clips from 51 action classes. **Diving48** [23] contains 18K untrimmed video clips from 48 action classes, all of which are types of dives. We report the top-1 accuracy on the evaluation set for all datasets following standard practices [12]. Only UCF101 and HMDB51 have multiple split versions; we use split 1.

4.2. Implementation Details

Model Configuration: We experiment with both 2D and 3D transformer backbones. The default backbone is ViT-Base [9] with global joint space-time attention. We also experimented with TimesFormer [3] with divided space-time attention. For fair comparison, we use the same input patch size of $2 \times 16 \times 16$ for all models following [39].

Pre-Processing: We pretrain with clips of 16 frames sampled at a temporal stride of 4 and 2 for K400 and SSv2 respectively following [39]. We use a fixed spatial resolution of 224×224 for all experiments. We apply multi-scale-crop and horizontal flip augmentation by default (flip is not applied to SSv2). We follow [39] to use AdamW [27] optimizer with a base learning rate $1.5e-4$, weight decay of 0.05, $\beta = [0.9, 0.95]$, and cosine learning rate decay.

Finetuning and Evaluation: We use the same 16-frame clip for finetuning and multi-view evaluation protocol following standard practice [12]. We use TSN-style sampling [44] on SSv2 dataset with 2 temporal $\times$ 3 spatial views during test-time following [39] for fair comparison. For Kinetics-400, UCF101, HMDB51 and Diving48, we use 5 temporal $\times$ 3 spatial views during test-time following [39] for fair comparison. See the supplementary material for hyperparameter details which are mostly the same as [39].

4.3. Main Results

Something-Something V2. We first show our MGM's performance on SSv2 in Table 1, and compare with previous

			finetune eval on SSv2			*finetune eval on K400*		
Model	**Backbone**	**GFLOPs**	**Epochs**	**Pretrain**	**Top1**	**Epochs**	**Pretrain**	**Top1**
Supervised								
SlowFast [12]	R101 + NL	106	196	K400	63.1	-	-	79.8
TimeSformer [3]	ViT-B	196	15	IN21K	59.5	15	IN21K	78.0
VidTr [51]	ViT-B	351	50	K400	63.0	-	-	79.1
MotionFormer [28]	ViT-B	370	90+35	IN21K + K400	66.5	90	IN21K	79.7
MViTv1 [10]	MViT-B	455	200	K400	67.7	-	-	81.2
MViTv2 [24]	MViT-B	225	200	K400	70.5	-	-	82.9
Self-supervised								
BEVT [45]	Swin-B	282	800+150	IN1K + K400	70.6	800+150	IN1K + K400	80.6
M^3Video [36]	ViT-B	180	400	SSv2	69.2	400	K400	79.7
MGM	ViT-B	180	400	SSv2	69.6	400	K400	80.3
VideoMAE [39]	ViT-B	180	800	SSv2	69.6	800	K400	80.0
MGM	ViT-B	180	800	SSv2	70.6	800	K400	80.8
MAM^2 [33]	ViT-B + VQGAN	180	1200	SSv2	71.3	800	K400	82.3
MGM	ViT-B	180	1200	SSv2	71.6	1200	K400	81.2
OmniMAE [13]	ViT-B	180	1600	IN1K + SSv2	69.5	1600	IN1K + K400	80.8
VideoMAE [39]	ViT-B	180	1600	SSv2	70.3	1600	K400	81.5
VideoMAE [39]	ViT-B	180	2400	SSv2	70.8	-	-	-
STMAE [11]	ViT-B	180	1600	K400	N/A	1600	K400	81.3
MotionMAE [50]	ViT-B	180	2400	SSv2	71.8	1600	K400	81.7
MGM	ViT-B	180	1600	SSv2	71.8	1600	K400	81.7
MGM	ViT-B	180	2400	SSv2	72.1	-	-	-

Table 1: Results comparison on Something-SomethingV2 (SSv2) and Kinetics-400 (K400). GFLOPs is listed as a single view's. All results are with fine-tuning. We apply repeated augmentation [18] = 2 for this table to compare against SOTA.

models trained in a supervised manner: Our MGM using vanilla ViT-B pretrained on only SSv2 outperforms previous SOTA such as MViTv1 [10] (+4.4%) and MViTv2 [10] (+1.6%) which uses a heavy hierarchical 3D transformer backbone and pretrains on Kinetics-400. This demonstrates the efficiency and effectiveness of our proposed method compared to supervised 3D transformer backbones. MGM also significantly outperforms previous supervised learning methods that use the same backbone as us: +5.6% over MotionFormer [28] and +12.6% over TimeSformer [3]. This demonstrates that the proposed MGM is effective for spatiotemporal modeling.

We then compare MGM with SOTA video MAE methods. With the same backbone and training schema, MGM achieves 1.3% higher accuracy comparing to VideoMAE [39]. Note that with $3\times$ less training time, our MGM pretrained for 800 epochs is able to achieve nearly the same performance as VideoMAE pretrained for 2400 epochs ($\sim$ 66% fewer epochs), as illustrated in Fig. 1. This demonstrates that motion-guided masking is a more efficient strategy for self-supervised spatiotemporal learning due to learning video saliency. Compared to OmniMAE [13] which does masked modeling on both images and video, MGM achieves +2.6% improvement. Compared to BEVT [45] which also learns from both images and video and additionally uses Swin Transformer [25, 26] – a heavy 3D Transformer architecture – MGM achieves +1.5% improvement. This is despite the fact that our method only uses video and is pretrained from scratch. Note that STMAE [11] only provides results for ViT-Large backbone on SSv2.

We finally compare our MGM with some most recent works that also utilize motion information for video pretraining. MGM outperforms M^3Video [36] (+0.4% with the same 400 epochs of pretraining) which reconstructs motion trajectories generated from optical flow. We achieve equivalent performance to MotionMAE [50] using 50% fewer epochs (1600 vs. 2400) even though it utilizes frame difference as an additional reconstruction target which incorporates explicit motion information. The results show that motion-guided masking alone is sufficient for improving spatiotemporal learning. It is worth reiterating that our MGM uses motion vectors for masking guidance, which is already computed during video encoding, and thus directly available during video decoding [30, 43, 47], making our method more efficient and scalable comparing to works such as [33, 36, 50] which use optical flow and/or frame difference.

Kinetics-400 (K400). We further conduct experiments on K400 and show results in Table 1. Similar trends are observed. Our MGM is able to outperform supervised models with the same backbone such as TimeSformer [3] by +3.7%

and MotionFormer [28] by +2.0%. MGM slightly underperforms some hierarchical 3D transformer backbones such as MViTv2 [24] by -1.2% on K400 but MViTv2 is trained with 2× more frames (32) and uses significantly more FLOPS.

Comparing with other MAE based methods, our MGM consistently outperforms VideoMAE by +1.4% at 800 epochs of pretraining. MGM also achieves better performance comparing to other recent works, at no additional cost. Our MGM outperforms M^{3}Video [36] at 400 epochs of pretraining by 0.6%, while being more efficient as our method does not use frame difference nor optical flow. Compared to OmniMAE [13] which uses both images and video, MGM achieves 0.9% improvement. Compared to BEVT [45], MGM achieves 1.1% improvement.

We note that the performance gain MGM obtains over other methods on K400 is slightly lower compared to on SSv2. This can be explained by the fact that a great proportion of actions in K400 dataset can be differentiated by the appearance of a few key frames, for example, "skiing" *vs*. "tennis". In contrast, recognising actions in SSv2 dataset, *e.g*. "moving something up" *vs*. "moving something down", requires the model to understand the subtle differences in motion along temporal dimension. Therefore, we hypothesize that spatiotemporal modeling is less impactful on spatial-heavy datasets such as K400 and the strength of MGM is not fully exhibited. Motion serves as a appearance-agnostic differentiator between classes.

4.4. Transfer Learning

We next evaluate MGM's performance when transferred to smaller datasets: UCF101 [34], HMDB51 [22], and Diving48 [23]. Performance in transfer learning is commonly used as an indicator of feature quality and representativeness [17]. We use ViT-Base model pretrained on K400 and SSv2 by 800 epochs for all transfer learning experiments.

Finetune on downstream tasks. We first present fine-tuning results from MGM pretrained on unlabeled K400 in Table 2. Our MGM consistently outperforms VideoMAE with same backbone and pretraining setup on all three datasets (+0.9% on UCF101, +1.1% on HMDB51, +4.9% on Diving48). This shows that our approach learns more representative semantics and thus generalizes better to small-scale downstream tasks. It is worth mentioning that MGM even outperforms K400 supervised pretraining on HMDB51 (+3.2%) and on Diving48 (+7.5%), demonstrating the effectiveness of our proposed method.

Our MGM is also able to consistently outperform VideoMAE [39] with pretraining on unlabeled SSv2 in the finetune setting (+1.3% on UCF101, +1% on HMDB51, +2.6% on Diving48). Our MGM even outperforms SSv2 supervised pretraining by a large margin (+14.1% on UCF101, +13.4% on HMDB51, +46.9% on Diving48). Finetuning MGM on SSv2 (same model from Tab. 1) does not lead to a large

	UCF	HMDB	Diving48
Supervised (K400) †	95.1	71.4	62.7
VideoMAE [39]	93.3	73.5	65.3
MGM	94.2	74.6	70.2
+ K400 Finetune (Tab. 1)	97.7	81.0	82.6

(a) **Pretrained on K400** and finetuned on downstream tasks.

	UCF	HMDB	Diving48
Supervised (SSv2) †	77.8	56.3	36.2
VideoMAE [39]	90.6	68.7	80.5
MGM	91.9	69.7	83.1
+ SSv2 Finetune (Tab. 1)	93.2	74.7	83.4

(b) **Pretrained on SSv2** and finetuned on downstream tasks.

	UCF	HMDB	Diving48
Supervised (K400) †	94.1	65.4	25.2
VideoMAE [39]	70.5	45.4	10.0
MGM	77.5	49.2	14.2

(c) **Pretrained on K400** and evaluated with linear probing on downstream tasks.

	UCF	HMDB	Diving48
Supervised (SSv2) †	72.9	51.8	13.2
VideoMAE [39]	65.3	41.2	10.7
MGM	69.1	44.8	10.2

(d) **Pretrained on SSv2** and evaluated with linear probing on downstream tasks.

Table 2: MGM generalizes well to various downstream datasets (UCF101, HMDB51 and Diving48) on two downstream tasks (finetune, linear-probe). We test VideoMAE and MGM pretrained on Kinetics-400 (K400) for 800 epochs. † The supervised baseline is ViT-B trained from scratch on K400 and SSv2 using the recipe from MViT [10, 24].

boost, indicating that our model is able to learn the majority of semantics just from unsupervised pretraining, and that unsupervised pretraining is able to make better use of the information in SSv2 than supervised training.

Linear Probe on downstream tasks. Previous work [2, 11, 16] argues that generative approaches such as MAE generally perform worse on linear probing tasks as there is a larger gap between the reconstruction task and downstream evaluation compared to other pretraining methods. Nevertheless, we compare linear probe performance between MGM and VideoMAE [39] and show that our method achieves significantly better performance. With K400 pretraining, we outperform VideoMAE by +7% on UCF101, +3.8% on HMDB51, and +4.2% on Diving48. With SSv2 pretraining, we outperform by +3.8% on UCF101, +3.6% on HMDB51, and underperform by -0.5% on Diving48. This demonstrates that motion-modeling helps reduce the gap between the reconstruction task and downstream tasks, as linear probe only

	UCF14 → HMDB7	HMDB14 → UCF7
VideoMAE [39]	55.7	69.6
MGM	57.6	71.4
+ K400 Finetune (Tab. 1)	79.1	96.2

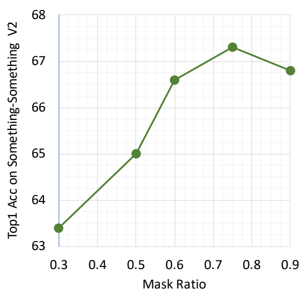
Table 3: MGM works well when trained on one dataset and evaluated on another dataset for an overlapping set of classes. The model sees no samples from test dataset, indicating motion understanding as motion is agnostic to spatial appearance. We tested VideoMAE and MGM pretrained on Kinetics-400 (K400) for 800 epochs.

trains a linear clasifier on top of features extracted from the frozen backbone; linear layers cannot learn new semantics.

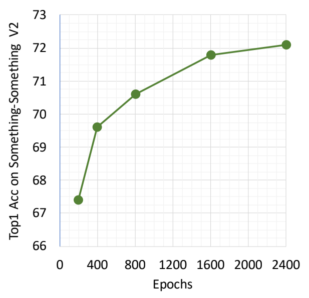
We note that there is a large gap of over 10% with K400 supervised pretraining on linear probe. This is a general deficiency of generative methods such as MAE-based pretraining [11] and we leave further exploration of this problem to future works. One way to address this would be to combine MAE with discriminative methods such as contrastive learning to obtain a balanced representation.

Domain adaptation. We further evaluate MGM on partial domain adaptation tasks following [49], in which the labels for the target dataset are a subset of the labels for the source dataset. The model is finetuned on the source dataset and directly evaluated on the target dataset without seeing any samples from the target dataset. This is more challenging than conventional domain adaptation as the model may incorrectly label samples in the target dataset as the labels from the source dataset which are not present in the target dataset ("negative transfer") [49]. Previous work has showed that optical flow is useful for action recognition because it is invariant to appearance [32]. Several works such as [6, 31] improve unsupervised domain adaptation in video by reducing background and appearance bias to capture the essence of an action via shuffling and mixup respectively. By similar reasoning, domain adaptation is a good evaluation protocol to assess motion learning as the source and target datasets have very different visual appearances, and motion serves as a common link between classes that is agnostic to spatial appearances.

We use the training sets provided by [49] for UCF14 → HMDB7 and HMDB14 → UCF7. Our MGM consistently outperform VideoMAE with K400 pretraining in both settings (+1.9% on HMDB7 and +1.8% on UCF7) indicating that MGM can also outperform in a domain adaptation setting. While there is a significant gap of over 20% between Table 3 and Table 2, finetuning our model on K400 labels bridges the gap. This suggests that our model has learned video saliency and can perform well without any large-scale label supervision, but additional label supervision is helpful for challenging settings such as domain adaptation where spatial information is very different and difficult to learn from limited samples. There are only ~1000 training samples in each setting.



(a) More mask ratios.



(b) Pretrain duration in epochs.

Figure 3: Ablations on mask ratio 3a and pretraining duration 3b. Mask ratio 0.75 is optimal and performance does not saturate with longer pretraining.

backbone	*SSv2*
ViT-S	61.1
ViT-B	**67.3**
TimesFormer [3]	64.7

(a) Backbone generalization.

guidance	*SSv2*
optical flow	66.0
motion vector	**67.3**

(b) Source of motion info.

aspect ratio	*SSv2*
no	67.1
yes	**67.3**

(c) Mask jitter.

pretrain data	*samples*	*K400*
Mini-K200 [48]	80K	78.2
K400 [19]	240K	78.6

(d) Dataset scale generalization.

	temporal propagation			***spatial continuity***		
Mask	**static**	**simulated**	**motion**	**sparse**	**dense**	Acc
Random Block	✓				✓	62.9
SMM (Sparse)		✓		✓		64.6
Random Tube	✓			✓		65.0
MGM (Sparse)			✓	✓		65.1
SMM (Dense)		✓			✓	65.9
MGM (Dense)			✓		✓	**67.3**

(e) Masking strategies differentiated by initial position, temporal propagation, and spatial continuity. Spatiotemporally continuous masks with motion guidance perform the best.

Table 4: Ablations on backbone 4a, source of motion guidance 4b, mask jitter 4c, pretrain data scale 4d, and mask type 4e. All ablations pretrain for 200 epochs on SSv2 unless otherwise indicated.

4.5. Ablations

We perform ablations on the Something-Something V2 dataset, with MGM with ViT-B backbone pretrained for 200 epochs using our motion-guided masking, unless otherwise specified.

Impact of masking ratio. Figure 3a studies the impact of different masking ratio on SSv2. We noticed that 0.75 masking ratio works best for the proposed MGM, which is lower than that of other works [11, 39]. This is because motion-guided masking leads to a more "challenging" reconstruction task compared to random masking, since the masked regions of videos are most informative and therefore cannot be easily inferred from retained regions.

Impact of pretraining length. We show the impact of different pretraining epochs on SSv2 in Figure 3b. The proposed MGM consistently improves with more training and we achieve state-of-the-art performance on SSv2 with 2400 epochs of pretraining. Performance does not seem to saturate with longer pretraining.

Generalization to different backbones. MGM also generalizes well to backbones of different sizes and types (Table 4a). We test vanilla ViT-S [9] and ViT-B [9], and TimeSformer [3] with joint space-time attention. Note that our TimeSformer ablation outperforms the supervised result reported by [3] of 59.5% on SSv2 which uses IN21K pretraining. In contrast, we do not use any extra data. ViT-S's lower performance may be explained by its low model capacity which may not be sufficient for learning spatiotemporal dynamics. As other methods do not use ViT-S, we cannot compare our result to other methods. We thus use ViT-B to compare fairly against other MAE works which use ViT-B.

Impact of motion source. In Table 4b we compare the performance of MGM when using optical flow vs. motion vectors as the source of motion guidance. Even though optical flow is more fine-grained and precise than motion vectors, performance of MGM drops when the mask is guided by optical flow. This could be related to the fact that optical flow is calculated on a per-pixel basis whereas motion vectors are computed in a block-wise manner and we utilize a block-wise mask strategy. We thus conveniently choose to use motion vector as it is also directly obtainable from compressed video. We extract optical flow using RAFT-Small [38]. On the SSv2 dataset, reading motion vectors is roughly 30× faster than computing optical flow (80 milliseconds vs. 2.5 seconds per video on average).

Impact of mask aspect ratio jitter. In Table 4c we ablate the use of aspect ratio jittering from Eq. (7). We see that aspect ratio jittering gives a small 0.2% boost in performance but is not a critical component of MGM.

Impact of pretrain dataset scale. In Table 4d we ablate the amount of pretraining data used to see if MGM can generalize to different dataset scales. We pretrain MGM on MiniKinetics-200 [48] and Kinetics-400 (K400) [19] which contain ∼80K videos and ∼240K videos respectively for 200 epochs. MiniKinetics-200 contains videos belonging to a subset of 200 of the most common classes in K400. We then evaluate top-1 accuracy on K400. We find that there is a small 0.4% drop in performance when pretraining on the smaller MiniKinetics-200. We expect performance to further improve when pretraining on datasets larger than K400.

Effectiveness of 3D motion masking. Table 4e studies different masking strategies which are broken down into categories based on degree of motion guidance and spatial continuity. We first compare dense to sparse masking, where sparse means that the mask may be spatially discontinuous. The dense variant of SMM and MGM outperform the respective sparse counterpart; dense SMM outperforms sparse SMM by 1.3% and dense MGM outperforms sparse MGM by 2.2%. This suggests the importance of maintaining spatial continuity in the masked reconstruction. Next, we compare MGM to SMM. Sparse MGM outperforms sparse SMM by 0.5% and dense MGM outperforms dense SMM by 1.4%. When motion is completely removed, performance is the worst even when the mask is spatially continuous (static block gets 62.9%). Dense MGM outperforms all masking strategies and specifically the random masking baseline by 2.3%. All masking strategies except block and sparse SMM outperform random masking. Random masking is spatiotemporally discontinuous. This validates our intuition that forming continuous 3D masks which model the true motion information within video is useful. We thus pick dense MGM for all other experiments and refer to "MGM" as the dense variant by default. For a visualization of these masking strategies, see Figure 4.

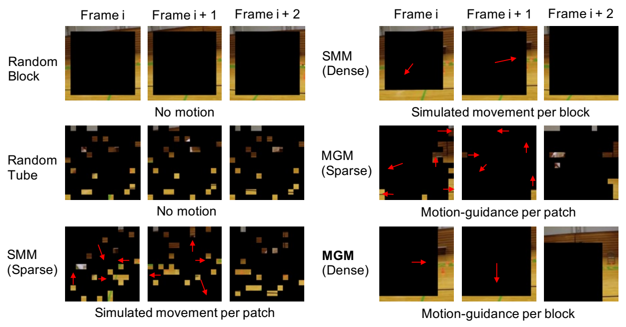


Figure 4: Visualization of different mask algorithms.

4.6. Visualization

In Figure 5, we provide visualizations from two perspectives using MGM pretrained on K400. Note that visualizations are subjective and should not be used as a formal explanation for model behavior. Our intention is to provide additional insight into the model to complement our quantitative results. We first visualize the RGB frames with bounding boxes, motion-guided masks, and motion vectors. We see that the motion-guided masks overlap with the majority of spatiotemporally salient regions. Second, we visualize the attention map using the center patch of the center frame as a query. We observe that the model mostly attends to the salient regions. For more visualizations, see the supplementary.

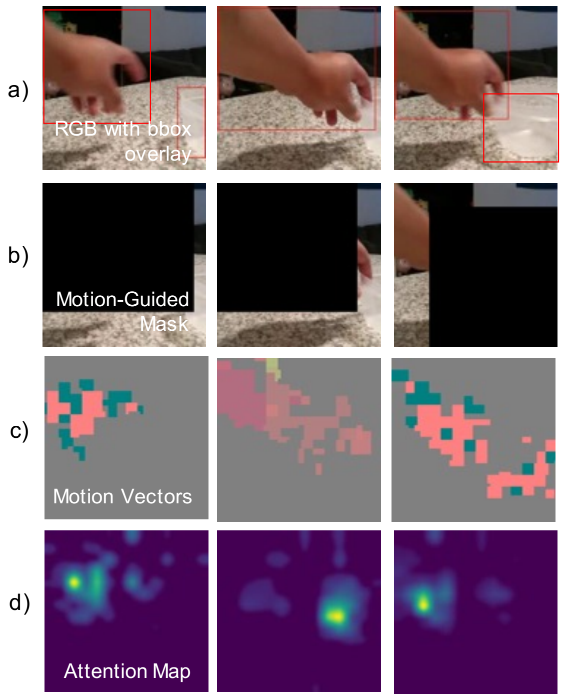


Figure 5: a) RGB frames with bounding boxes for visualization. b) Our MGM masks continuously cover the moving hand. c) Motion vector maps. d) Encoder attention map using center patch of center frame as query.

5. Discussion and Conclusion

Limitations. Our method efficiently leverages motion cues from compressed video format. However, MGM's effectiveness on the video benchmarks that are not human centric or have tremendous camera motion has yet to be examined. To the best of our knowledge, we did not find such a video benchmark, so we will leave this to future work.

Potential negative impact. As our model is trained on primarily Internet video datasets, our model will learn the bias inherent to that data. There could be unintended consequences and we advocate for complying with the law.

Conclusion. This paper introduces motion-guided masking (MGM), an algorithm which produces motion-aware 3D masks to improve spatiotemporal learning. Our use of motion information is much more efficient than previous optical flow as we leverage motion vectors that naturally exist in the video codec. We achieve new state-of-the-art or comparable performance on two challenging large-scale video benchmarks and achieve previous state-of-the-art results with 50% less pretraining time, making our method more efficient. We also achieve better generalization than previous video MAE works on three small-scale datasets. We believe our method has the potential to enable more efficient video training.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, October 2021. 1

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2021. 1, 2, 6

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 4, 5, 7, 8

[4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 2

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1

[6] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In *European Conference on Computer Vision*, pages 678–695. Springer, 2020. 7

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 1, 4, 8

[10] 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*, pages 6824–6835, 2021. 5, 6

[11] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *ArXiv*, abs/2205.09113, 2022. 1, 2, 3, 5, 6, 7, 8

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 4, 5

[13] Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Omnimae: Single model masked pretraining on images and videos. *ArXiv*, abs/2206.08356, 2022. 1, 2, 5, 6

[14] 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*, pages 5842–5850, 2017. 2, 4

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 1

[16] 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*, pages 16000–16009, 2022. 1, 2, 3, 6

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 1, 6

[18] 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. 5

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 4, 7, 8

[20] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 1

[21] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6232–6242, 2019. 2

[22] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 2, 4, 6

[23] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018. 2, 4, 6

[24] 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*, pages 4804–4814, 2022. 1, 5, 6

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 5

[26] 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*, pages 3202–3211, 2022. 5

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 4

[28] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506, 2021. 1, 5, 6

[29] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Patraucean, Florent Altch'e, Michael Valko, Jean-Bastien Grill, Aäron van den Oord, and Andrew Zisserman. Broaden your views for self-supervised video learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1235–1245, 2021. 1

[30] Iain E Richardson. *Video codec design: developing image and video compression systems*. John Wiley & Sons, 2002. 2, 4, 5

[31] Aadarsh Sahoo, Rutav Shah, Rameswar Panda, Kate Saenko, and Abir Das. Contrast and mix: Temporal contrastive video domain adaptation with background mixing. *Advances in Neural Information Processing Systems*, 34:23386–23400, 2021. 7

[32] Laura Sevilla-Lara, Yiyi Liao, Fatma Güney, Varun Jampani, Andreas Geiger, and Michael J Black. On the integration of optical flow and action recognition. In *German conference on pattern recognition*, pages 281–297. Springer, 2018. 7

[33] Yuxin Song, Min Yang, Wenhao Wu, Dongliang He, Fu Li, and Jingdong Wang. It takes two: Masked appearance-motion modeling for self-supervised video transformer pre-training. *arXiv preprint arXiv:2210.05234*, 2022. 1, 2, 3, 5

[34] 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. 2, 4, 6

[35] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 1

[36] Xinyu Sun, Peihao Chen, Liang-Chieh Chen, Thomas H. Li, Mingkui Tan, and Chuang Gan. M3video: Masked motion modeling for self-supervised video representation learning. *ArXiv*, abs/2210.06096, 2022. 1, 2, 3, 5, 6

[37] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021. 2

[38] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 8

[39] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *ArXiv*, abs/2203.12602, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[40] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[41] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 1

[42] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14010–14020, 2022. 1

[43] Jue Wang and Lorenzo Torresani. Deformable video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14053–14062, 2022. 1, 2, 5

[44] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018. 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. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14713–14723, 2022. 1, 2, 3, 5, 6

[46] Chen Wei, Haoqi Fan, Saining Xie, Chaoxia Wu, Alan Loddon Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14648–14658, 2022. 2

[47] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6026–6035, 2018. 2, 4, 5

[48] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 7, 8

[49] Yuecong Xu, Jianfei Yang, Haozhi Cao, Zhenghua Chen, Qi Li, and Kezhi Mao. Partial video domain adaptation with partial adversarial temporal attentive network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9332–9341, 2021. 7

[50] 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, 3, 5

[51] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021. 5