MV2MAE: Multi-View Video Masked Autoencoders

Ketul Shah[1]   Robert Crandall[2]   Jie Xu[2]   Peng Zhou[2]   Marian George[2]

Mayank Bansal[2]   Rama Chellappa[1]

[1]Johns Hopkins University   [2]Amazon

{kshah33, rchella4}@jhu.edu {rcrandal, xumji, zhoupz, margeo, maybans}@amazon.com

Abstract

Videos captured from multiple viewpoints can help in perceiving the 3D structure of the world and benefit computer vision tasks such as action recognition, tracking, etc. In this paper, we present a method for self-supervised learning from synchronized multi-view videos. We use a cross-view reconstruction task to inject geometry information in the model. Our approach is based on the masked autoencoder (MAE) framework. In addition to the same-view decoder, we introduce a separate cross-view decoder which leverages cross-attention mechanism to reconstruct a target viewpoint video using a video from source viewpoint. This helps learn representations robust to viewpoint changes. For videos, static regions can be reconstructed trivially which hinders learning meaningful representations. To tackle this, we introduce a motion-weighted reconstruction loss which improves temporal modeling. We report state-of-the-art results on the NTU-60, NTU-120 and ETRI datasets, as well as in the transfer learning setting on NUCLA, PKU-MMD-II and ROCOG-v2 datasets, demonstrating the robustness of our approach. Code will be made available.

1. Introduction

Multiple viewpoints of the same event are crucial to its understanding. Humans move around and obtain different viewpoints of objects and scenes, and develop a representation robust to viewpoint changes [26]. Different viewpoints often have very different appearance, which can help address challenges due to occlusion, lighting variations and limited field-of-view. In many real world scenarios, we have videos captured from multiple viewpoints, *e.g.* sports videos [45], elderly care [27], self-driving [69], complex robotic manipulation tasks [47] and security videos [9]. Learning a robust pre-trained model from large amounts of unlabeled synchronised multi-view data is of significant value for these applications. Such a model which is aware of the 3D geometry will be robust to changes in viewpoint

and can be effectively used as a foundation for downstream finetuning on smaller datasets for different tasks.

There has been significant progress in video self-supervised learning [46] for the single-view case, *i.e.* where synchronized multi-view data is not available. Recently, Masked Autoencoders (MAEs) as a paradigm for self-supervised learning has seen growing interest, and it has been successfully extended to video domain [17, 54, 62]. MAE-based methods achieve superior performance [54] on standard datasets such as Kinetics-400 [29] and Something-Something-v2 [20], compared to contrastive learning methods [16]. However, existing MAE-based pre-training approaches are not explicitly designed to be robust to viewpoint changes. View-invariant learning from multi-view videos has been widely studied using NTU [34, 50] and ETRI [27] datasets. However, most of these methods are based on 3D human pose, which is difficult to accurately capture or annotate for in-the-wild scenarios. Recently, there has been a growing interest in RGB-based self-supervised learning approaches leveraging multi-view videos [10, 31, 39, 57], facilitated by the availability of large-scale multi-view datasets [27, 34, 50]. ViewCLR [10], which achieves state-of-the-art results, introduces a latent viewpoint generator as a learnable augmentation for generating positives in a contrastive learning [7] framework. However, this method is memory intensive as it requires storing two copies of the feature extractor and two queues of features, and also requires multi-stage training. Considering the recent success of MAEs for video SSL, it is desirable to explore its potential in the multi-view video SSL scenario.

In this paper, we aim to learn a self-supervised video representation which is robust to viewpoint shifts. Humans learn a view invariant representation for tasks such as action recognition and are able to *visualize how an action looks from different viewpoints* [26]. We integrate this task of using features of one viewpoint to predict the appearance of a video from a different viewpoint in the standard MAE framework. More specifically, given a video of an activity from one viewpoint, it is patchified and a high fraction of the patches are masked out. The visible patches are en-
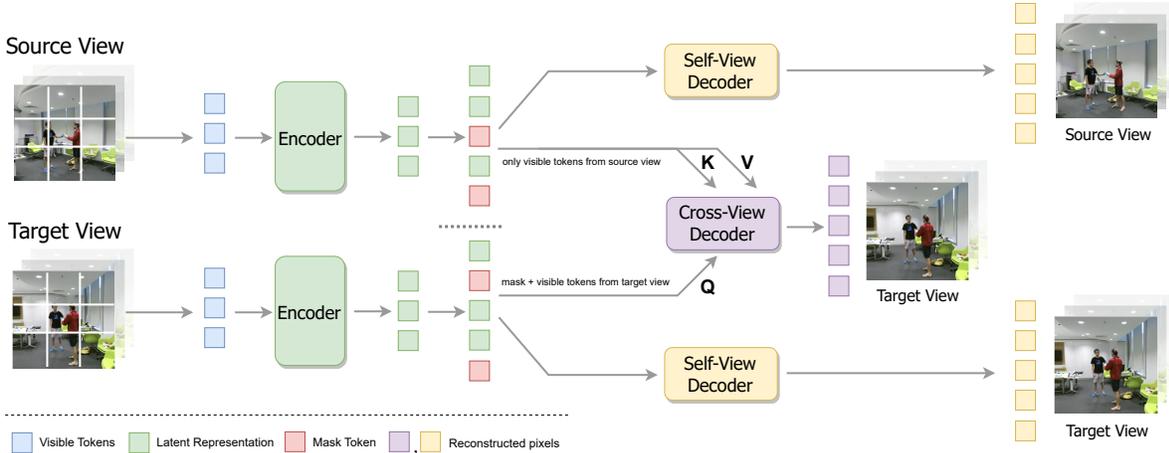
Figure 1. **Multi-View Video Masked Autoencoder (MV2MAE).** Source and target viewpoint videos are tokenized, the tokens are masked using random masking, and visible tokens are encoded for each view using a shared encoder. The cross-view decoder uses source view tokens to reconstruct the target viewpoint, whereas the standard decoder reconstructs each view separately.

coded, which the decoder uses (along with MASK tokens for missing patches) to reconstruct the given video. We introduce an additional cross-view decoder, which is tasked with reconstructing the masked patches of a target viewpoint by using the visible regions of from source view. This would require the model to understand the geometric relations between different views, enabling the construction of a robust pre-trained model. A challenge with MAE in videos is that they contain a lot of temporal redundancy, making it easier to reconstruct the static, background regions by simply copy pasting from adjacent frames where those are visible. Existing solutions for this problem involve specialized masking strategies using extra learnable modules [3, 24] or use tube masking [54, 62], which are not effective in certain scenarios, *e.g.* when motion is localized in a small region of the frame. We propose a simple solution without introducing additional learnable parameters by modifying the reconstruction loss to focus on moving regions. We can control the relative weights of moving and static regions using a temperature parameter.

We perform experiments on three multi-view video datasets: NTU-60 [50], NTU-120 [34], ETRI [27] for pre-training. Our method achieves SOTA fine-tuning accuracy on all benchmarks of these datasets. The robustness of our representation is shown in the transfer learning results on smaller datasets. We achieve SOTA results on NUCLA [59], ROCOG-v2 [44] and PKU-MMD-II [33] datasets in the transfer learning setting.

Our main contributions can be summarized as follows:

- We present an approach for self-supervised pre-training from multi-view videos using the MAE framework, and achieve state-of-the-art results on a variety on bench-

marks under full finetuning and transfer learning settings.

- Our approach uses cross-view reconstruction to inject geometry information in the model. This is done via a separate decoder with cross-attention mechanism which reconstructs target view from source view.

- We introduce a simple motion-focused reconstruction loss for improved temporal modeling, while also allowing us to specify the degree to which to focus on moving regions.

2. Related Work

2.1. Self-Supervised Learning from Videos

Pretext Learning. Many pretext tasks have been proposed for learning self-supervised video representations, initially inspired from the progress in SSL for images. Tasks such as video rotation prediction [28], solving spatio-temporal jigsaw [1], predicting motion and appearance statistics [60] were direct extensions of their image counterparts, and showed impressive performance. Methods leveraging the temporal order in videos for constructing pretext tasks such as frame ordering [67] and odd-one-out learning [18] were also proposed. These methods were outperformed by contrastive learning approaches.

Contrastive Learning. These methods create augmented versions of the input (positives) which preserve the semantic content of the input. The contrastive loss is used to pull these closer together in the feature space, while simultaneously pushing them away from other samples (negatives). Numerous ways of generating positive pairs were proposed such as using random clips from the same video, clips of different frame rates [61], choosing nearby clips [41], and

using optical flow [21], among others.

Masked Video Modeling. Recently, masked video modeling has emerged as a promising area for self-supervised learning. Methods such as BEVT [63], MaskFeat [65], VideoMAE [54], MAE-ST [17] show superior performance on the standard video self-supervised learning benchmarks. Different reconstruction targets have been studied, such as MVD [64] which uses distillation from pre-trained features, and MME [53] which reconstructs motion trajectories. To tackle trivial reconstruction solution via copy-paste in videos, which becomes and issue due to high redundancy, different masking strategies have been proposed. MGMAE [24] uses motion-guided masking based on motion vectors, VideoMAE [54] proposed using tube masking, AdaMAE [3] introduces a neural network for mask sampling. Orthogonal to these, we propose to tackle the issue by using a motion-weighted reconstruction loss. Moreover, unlike our approach, existing MAE pre-training approaches are not explicitly designed to be robust to viewpoint shifts.

2.2. Multi-View Action Recognition

Early work in this area designed hand-crafted features which were robust to viewpoint shifts [39, 43, 66]. Many unsupervised learning approaches have been proposed for learning representations robust to changes in viewpoint. A large number of methods leverage 3D human pose information, which greatly aids in achieving view invariance. Methods based on RGB modality [10, 31, 57] have gained increasing popularity. These can be broadly divided into two categories:

One trend is to enforce the latent representations of different viewpoints to be close. Along this line, [71] follows a dictionary learning approach and encourages videos of different views to have the same sparse representation. [42] fits a 3D human model to a mocap sequence and generates videos from multiple viewpoints, which are forced to predict the same label. More recently, methods based on contrastive learning have been proposed such as ViewCLR [10] which achieves remarkable performance. They add a latent viewpoint generator module which is used to generate positives in the latent space corresponding to different views.

Another line of work uses one viewpoint to predict another. [31] uses cross-view prediction in 3D flow space by using depth as an additional input to provide view information. Their approach also uses a gradient reversal layer for enforcing view invariance. [57] uses the encoded source view features to render same video from unseen viewpoint and a random start time. Their approach hence needs to be able to predict across time and viewpoint shifts. They leverage a view embedding which requires information of camera height, distance and angle. In contrast the these approaches which rely on view embedding or depth for providing viewpoint information, the view information is inherently available in the visible patches of the viewpoints in our approach.

3. Method

3.1. Preliminary: Masked Video Modeling

Here we revisit the masked autoencoder (MAE) framework for videos. Given a video, we first sample T frames with stride τ to get the input clip: $\mathbf{I} \in \mathbb{R}^{C \times T \times H \times W}$. Here, $H \times W$ is the spatial resolution, T denotes the number of frames sampled, and C is the number of input (RGB) channels. The standard MAE architecture has three main components: tokenizer, encoder, decoder.

Tokenizer. The input clip is first converted into N patches using a patch size of $t \times h \times w$, where $N = \frac{T}{t} \times \frac{H}{h} \times \frac{W}{w}$. The tokenizer returns N tokens of dimension d by first linearly embedding these N patches. This is implemented in practice using a strided 3D convolution layer. Next, we provide position information to these tokens by adding positional embeddings [56].

Encoder. A high fraction of these N tokens are dropped with a masking ratio $\rho \in (0,1)$. Different masking strategies [3, 24, 54] have been explored for choosing which tokens to mask out. Next, the remaining small fraction of visible tokens are passed through the encoder ($\Phi_{\texttt{enc}}$) to obtain latent representations. The encoder is a vanilla ViT [15] with joint space-time attention [54]. These latent representations need to capture the semantics in order to reconstruct the masked patches.

Decoder. The encoded latent representations of the visible patches are concatenated with learnable $\texttt{MASK}$ tokens corresponding to masked out patches, resulting in combined tokens $\mathbf{Z}_{\texttt{c}}$. Then the positional embeddings are added for all tokens, and passed through a light-weight decoder ($\Phi_{\texttt{dec}}$) to get the predicted pixel values $\hat{\mathbf{I}} = \Phi_{\texttt{dec}}(\mathbf{Z}_{\texttt{c}})$.

The loss function is the mean squared error (MSE) between the reconstructed values and the normalized pixel values [17, 54], for masked patches Ω.

$$\mathcal{L} = \frac{1}{\rho N} \sum_{i \in \Omega} |\mathbf{I}_i - \hat{\mathbf{I}}_i|^2 \qquad (1)$$

3.2. Cross-View Reconstruction

The goal of cross-view reconstruction is to predict the missing appearance of a video in target viewpoint given videos from one (few) source viewpoint(s). Being able to extrapolate across viewpoints requires understanding the geometric relations between different viewpoints, making it an effective task for learning representations robust to viewpoint variations.

As shown in Figure 1, consider two synchronized videos of an activity, $\mathbf{I}^{\texttt{sv}}$ and $\mathbf{I}^{\texttt{tv}}$, from source view ($\texttt{sv}$) and target view ($\texttt{tv}$) respectively. We first tokenize, mask and
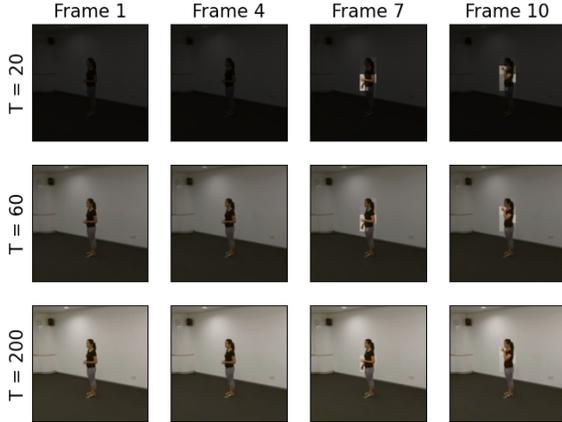
Figure 2. **Motion weights.** Each row shows motion weights overlaid on the input frames for different temperature values. Higher temperature increases the weight on static regions.

encode the visible tokens for each video separately using a shared encoder Φ_{enc}. We introduce a cross-view decoder ($\Phi_{\text{dec}}^{\text{cross-view}}$) which uses a cross-attention mechanism. This decoder additionally uses the visible tokens in the source view to reconstruct the target viewpoint video, $\hat{\mathbf{I}}^{\text{tv}} = \Phi_{\text{dec}}^{\text{cross-view}}(\mathbf{Z}_{\text{c}}^{\text{tv}}, \mathbf{Z}_{\text{vis}}^{\text{sv}})$. More specifically, each block of the cross-view decoder consists of cross-attention and self-attention layers, followed by a feed-forward layer. The tokens from the target view attend to the visible source view tokens using cross-attention, and then to each other using self-attention. A key aspect of methods based on cross-view prediction is how the viewpoint information is provided: [57] conditions the decoder on a viewpoint embedding, while some approaches [31] use extra modalities such as depth to provide information about the target viewpoint. In contrast to these, the visible patches provide the required target viewpoint information in our approach. The *amount* of view information we want to provide can be easily varied by changing the masking ratio. Moreover, the standard decoder (Φ_{dec}) is used to reconstruct video from each viewpoint independently $\hat{\mathbf{I}}^{\text{vp}} = \Phi_{\text{dec}}(\mathbf{Z}_{\text{c}}^{\text{vp}})$ for $\text{vp} \in \{\text{sv}, \text{tv}\}$.

Figure 4 visualizes the cross-view reconstruction quality and cross-attention maps, which demonstrates that the model learns to focus on relevant regions across viewpoints.

3.3. Motion-Weighted Reconstruction Loss

A given video can be decomposed into static and dynamic regions. Static regions typically involve scene background and objects which do not move throughout the video. Patches from such regions are trivial to reconstruct [3, 53] due to temporal redundancy in videos. In order to deal with this, we offer a simple solution by re-weighting the reconstruction loss of each patch proportional to the amount of motion within that patch. The motion weights used for re-

Algorithm 1: PyTorch code for motion weights.

```
# frames    : input frames of shape [B,C,T,H,W]
# patch_size: (p_time, p_height, p_width)
# t         : temperature parameter

fdiff = frames[:,:,1:,:,:] - frames[:,:,:-1,:,:]
fdiff = torch.cat([fdiff[:,:,0:1,:,:], fdiff],
    dim=2)
fdiff = rearrange(fdiff,'b c (t p0) (h p1) (w p2)
    -> b (t h w) (p0 p1 p2 c)', p0=patch_size
    [0], p1=patch_size[1], p2=patch_size[2])
fdiff = torch.abs(fdiff)
fdiff = torch.linalg.vector_norm(fdiff, dim=2,
    keepdim=True) # B N 1
motion_weights = torch.nn.functional.softmax(
    fdiff/t, dim=1) # B N 1
```

weighting are obtained using frame difference for simplicity. Note that other motion features such as optical flow, motion history image, etc can be used in place of frame difference, but frame difference is extremely fast to compute. In order to get the final weights, we take the norm of frame difference within each patch, and apply softmax over all tokens, as shown in Algorithm 1. We can control the extent to which to focus on the moving regions by controlling the temperature parameter. The higher the temperature value, the more uniform the resulting weights. Examples of motion weights overlaid on the original frames for different temperature values are shown in Figure 2. PyTorch-style code for computing the motion weights for a video is presented in Algorithm 1. The final motion-weighted reconstruction loss is given below, where w_i is the weight for i^{th} patch:

$$\mathcal{L} = \frac{1}{\rho N} \sum_{i \in \Omega} w_i \times |\mathbf{I}_i - \hat{\mathbf{I}}_i|^2 \qquad (2)$$

4. Experiments

We evaluate our approach on several common multi-view video datasets: NTU60 [50], NTU120 [34], ETRI [27], NU-CLA [59], PKU-MMD [33], and ROCOG [44]. For NTU and ETRI, we achieve state-of-the-art results by pre-training and fine-tuning on the target domain. On NUCLA, PKU-MMD-II, and ROCOG-v2, we demonstrate excellent transfer learning performance by pre-training only on NTU, and fine-tuning on the target dataset.

4.1. Datasets

NTU RGB+D 60. [50] is a large-scale multi-view action recognition dataset, consisting of 56,880 videos from 60 distinct action classes. These videos were recorded from 40 subjects using Kinect-v2. Each activity instance is simultaneously captured from three viewpoints. The dataset consists of two benchmarks outlined in [50]: (1) Cross-Subject (xsub) and (2) Cross-View (xview). In the cross-subject benchmark, the 40 subjects are divided into training

4

Table 1. Comparison with state-of-the-art on cross-view and cross-subject benchmarks of NTU-60 dataset. **Top:** supervised methods using multiple modalities, **Middle:** supervised methods using only RGB modality, **Bottom:** unsupervised methods using any modality. Labels ✓: Supervised methods

Method	Modality	Labels	NTU-60 (%) xview	xsub
STA-Hands [4]	RGB+Pose	✓	88.6	82.5
Separable STA [11]	RGB+Pose	✓	94.6	92.2
VPN [12]	RGB+Pose	✓	96.2	93.5
ESE-FN [51]	RGB+Pose	✓	96.7	92.4
DA-Net [58]	RGB	✓	75.3	–
Zhang et al. [70]	RGB	✓	70.6	63.3
Glimpse Clouds [5]	RGB	✓	93.2	86.6
DMCL [19]	RGB	✓	–	83.6
Debnath et al. [13]	RGB	✓	–	87.2
FSA-CNN [27]	RGB	✓	92.2	88.1
Piergiovanni et al. [40]	RGB	✓	93.7	–
ViewCon [49]	RGB	✓	98.0	91.4
Li et al. [31]	Flow	✗	83.4	80.9
HaLP [48]	Pose	✗	88.6	82.1
Vyas et al. [57]	RGB	✗	86.3	82.3
ViewCLR [10]	RGB	✗	94.1	89.7
MV2MAE (Ours)	RGB	✗	**95.9**	**90.0**

Table 2. Comparison with state-of-the-art on cross-setup and cross-subject benchmarks of NTU-120 dataset. **Top:** supervised methods using multiple modalities, **Middle:** supervised methods using single modality, **Bottom:** unsupervised methods. MV2MAE outperforms previous SOTA unsupervised methods using any modality. Labels ✓: Supervised methods

Method	Modality	Labels	NTU-120 (%) xset	xsub
Hu et al. [23]	RGB+Depth	✓	44.9	36.3
Hu et al. [22]	RGB+Depth	✓	54.7	50.8
Separable STA [11]	RGB+Pose	✓	82.5	83.8
VPN [12]	RGB+Pose	✓	87.8	86.3
PEM [35]	Pose	✓	66.9	64.6
2s-AGCN [30]	Pose	✓	84.9	82.9
MS-G3D Net [36]	Pose	✓	88.4	86.9
CTR-GCN [8]	Pose	✓	90.6	88.9
Two-streams [52]	RGB	✓	54.8	58.5
Liu et al. [34]	RGB	✓	54.8	58.5
I3D [6]	RGB	✓	80.1	77.0
DMCL [19]	RGB	✓	84.3	–
ViewCon [49]	RGB	✓	87.5	85.6
HaLP [48]	Pose	✗	73.1	72.6
ViewCLR [10]	RGB	✗	86.2	84.5
MV2MAE (Ours)	RGB	✗	**87.1**	**85.3**

and testing sets, with 20 subjects in each. In the cross-view scenario, videos from cameras 2 and 3 are used for training, while testing is performed on videos from camera 1.

NTU RGB+D 120. [34] is the extended version of the NTU-60 dataset which contains 114,480 videos spanning 120 action categories. Our evaluation follows the established protocols outlined in [34]: (1) Cross-Subject (xsub) and (2) Cross-Setup (xset). In the cross-subject scenario, subjects are partitioned into training and testing groups, while in the cross-setup setting, the data is divided into training and testing subsets based on the setup ID.

ETRI. [27] is another large-scale multi-view action recognition dataset consisting of activities of daily living for elderly care. It has 112,620 videos captured from 55 action classes. All activity instances are recorded from 8 synchronized viewpoints. [27] describes a cross-subject benchmark which we use to evaluate our approach.

4.2. Implementation Details

We sample a clip of 16 RGB frames with a stride of 4 from each video. We downsample the resolution of frames to 128×128 following [10]. During pre-training, we only apply random resized crops as augmentation. We use a tempo-

Table 3. Comparison with state-of-the-art cross-subject benchmark of ETRI dataset. MV2MAE performs better than prior work, which are all supervised approaches.

Method	Modality	Labels	ETRI (%) xsub
ESE-FN [51]	RGB+Pose	✓	95.9
FSA-CNN [27]	RGB	✓	90.6
ConViViT [14]	RGB	✓	95.1
MV2MAE (Ours)	RGB	✗	**96.5**

ral patch size of 2 and a spatial patch size of 16×16, which results in 512 tokens. A masking ratio of 0.7 is used unless otherwise specified. We choose fixed sinusoidal spatio-temporal positional position embedding following [3, 54]. All of our experiments use the vanilla ViT-S/16 [55] architecture as the encoder (unless otherwise noted), trained using AdamW optimizer [37]. The pre-training is carried out for 1600 epochs. Please refer to the supplementary material for more details.

We evaluate our pre-trained models using two settings: 1) end-to-end fine-tuning on the same datasets and 2) trans-

Table 4. Transfer learning on NUCLA. Unsupervised methods (Labels: ✗) have been pre-trained on NTU-60 dataset. MV2MAE significantly outperforms other methods showing remarkable transfer capability of our representations.

Method	Modality	Labels	NUCLA (%) xview
STA [11]	RGB+Pose	✓	92.4
VPN [12]	RGB+Pose	✓	93.5
DA-Net [58]	RGB	✓	86.5
Glimpse Cloud [5]	RGB	✓	90.1
I3D [6]	RGB	✓	88.8
ViewCon [49]	RGB	✓	91.7
MS^{2}L [32]	Pose	✗	86.8
Li *et al.* [31]	Depth	✗	62.5
Colorization [68]	Depth	✗	94.0
Vyas *et al.* [57]	RGB	✗	83.1
ViewCLR [10]	RGB	✗	89.1
MV2MAE (Ours)	RGB	✗	**97.6**

Table 5. Transfer learning on PKU-MMD-II. All methods use NTU-120 dataset for pre-training. MV2MAE surpasses other unsupervised methods, all of which use Pose modality.

Method	Modality	Labels	PKU-MMD-II (%) xsub
CrosSCLR-B [72]	Pose	✗	52.8
CMD [38]	Pose	✗	57.0
HaLP [48]	Pose	✗	57.3
MV2MAE (Ours)	RGB	✗	**60.1**

Table 6. Transfer learning on ROCOG-v2 ground dataset.

Method	Modality	ROCOG-v2 (%)
Reddy *et al.* [44]	RGB	87.0
MV2MAE (Ours)	RGB	**89.0**

fer learning on smaller datasets. We discard the decoders and attach a classifier head which uses the global average pooled features for classification. For testing, we sample 5 temporal clips, and use 10 crops from each following [10], and the final prediction is the average of these.

4.3. Comparison with state-of-the-art

We compare our approach with previous SOTA methods on the cross-subject (xsub) and cross-view (xview) benchmarks of the commonly used NTU-60 and NTU-120 datasets. We also present our results on the ETRI dataset, which only has a cross-subject benchmark.

Table 1 and Table 2 show results on the NTU-60 and NTU-120 datasets. We outperform all previous unsupervised methods based on RGB, Flow or Pose modality on both cross-view and cross-subject benchmarks of the two datasets. On NTU-120, our method approaches the performance of RGB-based *supervised* methods. In the xsub setting, we see an improvement of +0.3% and +1.2% on NTU-60 and NTU-120 respectively, and in the xview setting, observe an improvement of +1.8% and +0.9% respectively. Our approach is also faster to train [54] and more memory efficient compared to ViewCLR [10], which uses a MoCo [7] framework and requires storing two copies of the model and two queues in memory. [57] uses a cross-view prediction paradigm but performs poorly (86.3% vs 95.9% on xview and 82.3% vs 90.0% on xsub) despite using more parameters (∼72M vs ∼22M). Unlike their approach which relies only on viewpoint embeddings for information of the

target viewpoint, we implicitly have that information in the visible patches from target view, shows the effectiveness of our pre-training mechanism.

4.4. Transfer Learning Results

Transfer learning is an important setting for evaluating the generalization capabilities of pre-trained models. The model is initialized using pre-trained weights and fine-tuned on smaller datasets. We perform transfer learning experiments on three action recognition datasets: 1) NUCLA, 2) PKU-MMD-II, and 3) ROCOG-v2.

NUCLA [59] is a multi-view action recognition dataset consisting of 1493 videos spanning 10 action classes. Each activity has been captured from three viewpoints, and we follow the cross-view protocol for our experiments. PKU-MMD-II [33] is another dataset for 3D action understanding, consisting of 6945 videos from 51 activity classes. Following prior work [48], we use the phase 2 of the dataset and evaluate our approach on the cross-subject setting. ROCOG-v2 [44] is a gesture recognition dataset consisting of 304 videos from ground viewpoint from 7 gestures.

As shown in Table 4, our method achieves significantly better performance than prior *supervised and unsupervised* methods on the NUCLA dataset. We improve by 8.5% upon the previous RGB-based SOTA approach [10]. On the PKU-MMD-II dataset, our method outperforms prior work, all of which are based on Pose modality, by 2.8% as shown in Table 5. It is interesting to note that although the Pose modality shows superior performance in supervised setting (Table 2), it lags behind when used for self-supervised learning in both in-domain fine-tuning (Table 2)

and transfer learning (Table 5) settings. Finally, we show results on the ROCOG-v2 dataset in Table 6, where we gain an improvement of 2%. These transfer learning results clearly demonstrates that the representations learnt using our approach generalize well.

4.5. Ablation Study and Analysis

How much emphasis to place on reconstructing moving patches? In our approach, the motion weights can be adjusted to modulate the emphasis on moving patches using the temperature parameter in Algorithm 1. As shown in Figure 2, lower temperature value places more focus on reconstructing patches with more motion, and increasing the temperature increases the weight given to the background pixels. Figure 3 shows the influence of the temperature parameter on accuracy. From the plot, we see that a temperature value of 60 performs best, which is used in all our experiments. Increasing the weights of background patches by increasing temperature degrades the performance. This is because it is trivial to reconstruct the background patches by copy-paste from nearby frames. The performance degrades significantly to 82.46% if each patch is weighted equally.
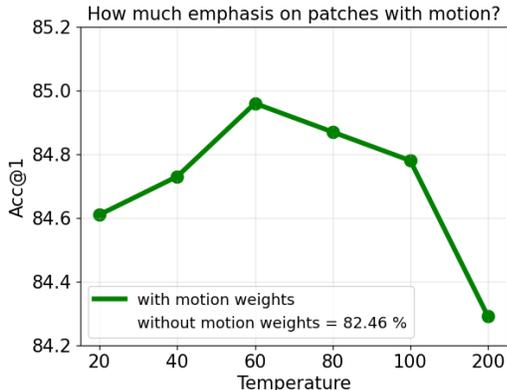

Figure 3. **Temperature parameter** of motion weights modulates the focus on static vs moving regions as visualized in Figure 2.

Masking Ratio. We study the impact of masking ratio in Table 7. We note that the optimal masking ratio is lower in our multi-view setting than the single-view setting in [54], which we hypothesize is because the model needs more information from each individual view to effectively infer cross-view geometry.

Model Scaling. To study how the performance scales with models of different capacities, we compare the fine-tuning performance of pre-trained models with ViT-T, ViT-S, and ViT-B in Table 8 on the NTU-120 cross-subject setting. Our approach effectively pre-trains larger models using the same amount of data.

Visualizing Cross-Attention Maps and Reconstructions.

Table 7. **Masking Ratio.** MV2MAE performs best with a masking ratio of 0.7 which is needed for effectively inferring cross-view geometry.

Masking ratio (ρ)	0.6	0.7	0.8	0.9
NTU-120 xsub (%)	83.6	**85.3**	84.3	83.4

Table 8. **Increasing Model Capacity**. We observe that our approach scales effectively with bigger models.

Backbone	ViT-T	ViT-S	ViT-B
NTU-120 xsub (%)	82.0	83.4	85.1

Here, we analyze the cross-view decoder by visualizing the cross-attention maps and the cross-view reconstruction quality. The cross-attention maps are visualized in Figure 4. The first and second rows show the input and masked input frames from the target viewpoint, with the masked query token circled in red. The third row shows the reconstructed target view from the cross-view decoder. The last row shows the cross-attention map for the query overlaid on the source view frames. We can see that model is able to find matching regions in the source view, demonstrating the learnt geometry.

How many source views to use? For the cross-view decoder, we study the effect of number of source viewpoints used in Table 9. For these experiments, all viewpoints used are chosen randomly from available synchronized views. The performance is similar when using one or two source viewpoints. We observe that the fine-tuning accuracy drops if we use more source viewpoints for reconstructing the target viewpoint, by making the reconstruction task easier.

Table 9. **Number of Source Views.** Having more source views makes reconstruction task easier and degrades performance.

# Source Views	1	2	3
ETRI xsub (%)	94.0	93.9	93.1

How different should the views be? A natural question that arises is which viewpoint should we use? In other words, given a target viewpoint, how far should the source view be? We study this by fixing the target view to be View1, and varying the source view to be View2, View3 or View4, as shown in Figure 5. The results are reported in Table 10, which shows that the performance drops if the chosen target and source viewpoints are separated by a lot.
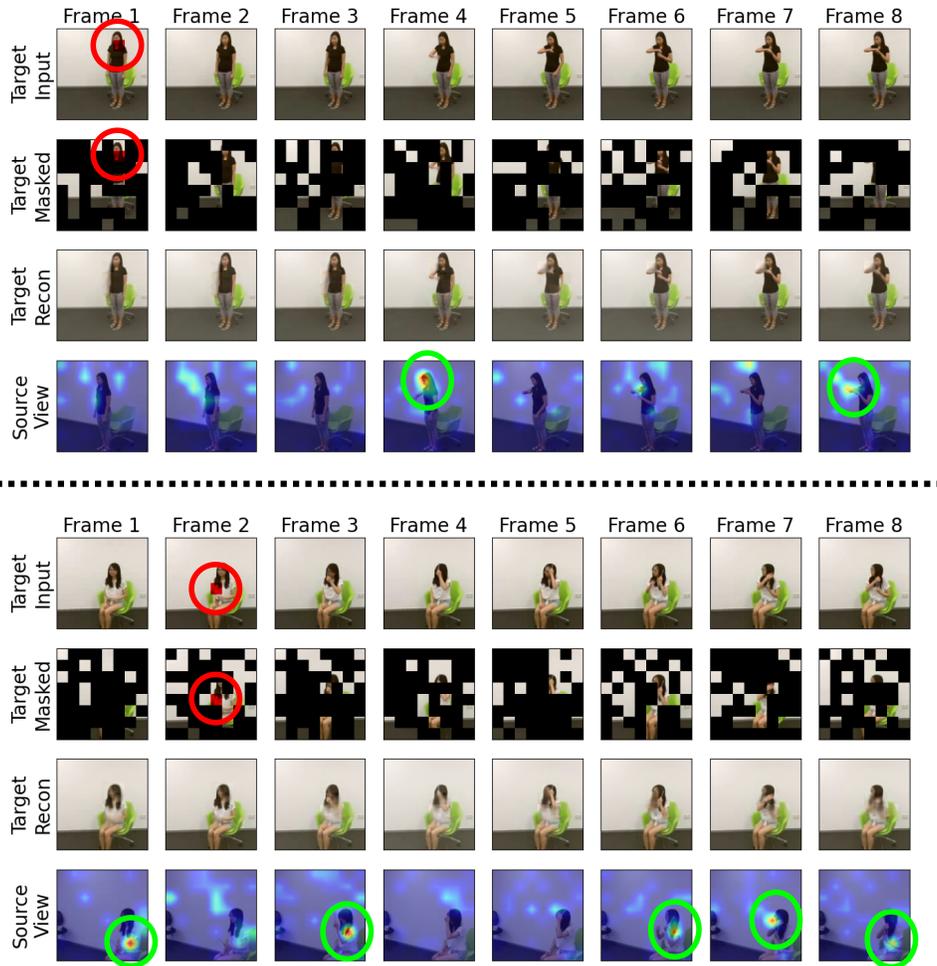
Figure 4. **Cross-View Decoder Qualitative Analysis.** We visualize the reconstructions and cross-attention maps from the cross-view decoder. **First row:** Target viewpoint input frames, **Second row:** Masked input frames from target viewpoint, **Third row:** Reconstruction of target view from the cross-decoder, **Last row:** Cross-attention maps visualized on source view frames. The red circle indicates the query token whose attention maps are visualized. green circles shows that the model is able find matching regions across viewpoints.

Table 10. **Which views to choose?** We study the effect of the distance between the source and target views. Keeping the target view fixed (View1), we vary the choice of source view and find that the performance degrades if the two views are very different from each other.

Source View	View2	View3	View4
ETRI xsub (%)	94.7	94.4	94.3

Figure 5. Example of synced views from ETRI dataset.

5. Conclusion

In conclusion, this paper proposes a self-supervised learning approach for harnessing the power of multi-view videos within the masked autoencoder framework. Our method integrates a cross-view reconstruction task, leveraging a dedicated decoder equipped with cross-attention mechanism to instill essential geometry information into the model. The introduction of a motion-focused reconstruction loss further enhances temporal modeling. Through comprehensive evaluation using full fine-tuning and transfer learning settings on multiple datasets, our approach consistently exhibits remarkable efficacy.

References

[1] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189. IEEE, 2019.

[2] 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*, pages 6836–6846, 2021.

[3] Wele Gedara Chaminda Bandara, Naman Patel, Ali Gholami, Mehdi Nikkhah, Motilal Agrawal, and Vishal M Patel. Adamae: Adaptive masking for efficient spatiotemporal learning with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14507–14517, 2023.

[4] Fabien Baradel, Christian Wolf, and Julien Mille. Human action recognition: Pose-based attention draws focus to hands. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 604–613, 2017.

[5] Fabien Baradel, Christian Wolf, Julien Mille, and Graham W. Taylor. Glimpse clouds: Human activity recognition from unstructured feature points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[8] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021.

[9] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1060–1068, 2021.

[10] Srijan Das and Michael S Ryoo. ViewCLR: Learning self-supervised video representation for unseen viewpoints. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5573–5583, 2023.

[11] Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[12] Srijan Das, Saurav Sharma, Rui Dai, Francois Bremond, and Monique Thonnat. VPN: Learning video-pose embedding for activities of daily living. In *European Conference on Computer Vision*, pages 72–90. Springer, 2020.

[13] Bappaditya Debnath, Mary O'brien, Swagat Kumar, and Ardhendu Behera. Attentional learn-able pooling for human activity recognition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13049–13055. IEEE, 2021.

[14] Rachid Reda Dokkar, Faten Chaieb, Hassen Drira, and Arezki Aberkane. Convivit–a deep neural network combining convolutions and factorized self-attention for human activity recognition. *arXiv preprint arXiv:2310.14416*, 2023.

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

[16] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross B. Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3298–3308, 2021.

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

[18] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017.

[19] Nuno Cruz Garcia, Sarah Adel Bargal, Vitaly Ablavsky, Pietro Morerio, Vittorio Murino, and Stan Sclaroff. Distillation multiple choice learning for multimodal action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2755–2764, 2021.

[20] 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.

[21] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *Advances in Neural Information Processing Systems*, 33:5679–5690, 2020.

[22] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for RGB-D activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5344–5352, 2015.

[23] Jian-Fang Hu, Wei-Shi Zheng, Lianyang Ma, Gang Wang, Jianhuang Lai, and Jianguo Zhang. Early action prediction by soft regression. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2568–2583, 2018.

[24] 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 In-*

ternational Conference on Computer Vision, pages 13493–13504, 2023.

[25] Hochul Hwang, Cheongjae Jang, Geonwoo Park, Junghyun Cho, and Ig-Jae Kim. Eldersim: A synthetic data generation platform for human action recognition in eldercare applications. *IEEE Access*, 2021.

[26] Leyla Isik, Andrea Tacchetti, and Tomaso Poggio. A fast, invariant representation for human action in the visual system. *Journal of neurophysiology*, 119(2):631–640, 2018.

[27] Jinhyeok Jang, Dohyung Kim, Cheonshu Park, Minsu Jang, Jaeyeon Lee, and Jaehong Kim. ETRI-activity3D: A large-scale rgb-d dataset for robots to recognize daily activities of the elderly. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10990–10997. IEEE, 2020.

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

[29] 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.

[30] Shi Lei, Zhang Yifan, Cheng Jian, and Lu Hanqing. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[31] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Unsupervised learning of view-invariant action representations. *Advances in neural information processing systems*, 31, 2018.

[32] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 2490–2498, New York, NY, USA, 2020. Association for Computing Machinery.

[33] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017.

[34] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. NTU RGB+D 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701, 2020.

[35] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[36] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020.

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

[38] Yunyao Mao, Wengang Zhou, Zhenbo Lu, Jiajun Deng, and Houqiang Li. Cmd: Self-supervised 3d action representation learning with cross-modal mutual distillation. In *European Conference on Computer Vision (ECCV)*, 2022.

[39] Vasu Parameswaran and Rama Chellappa. View invariance for human action recognition. *International Journal of Computer Vision*, 66(1):83–101, 2006.

[40] AJ Piergiovanni and Michael S. Ryoo. Recognizing actions in videos from unseen viewpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4124–4132, 2021.

[41] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6960–6970. IEEE, 2021.

[42] Hossein Rahmani and Ajmal Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2458–2466, 2015.

[43] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50:203–226, 2002.

[44] Arun V Reddy, Ketul Shah, William Paul, Rohita Mocharla, Judy Hoffman, Kapil D Katyal, Dinesh Manocha, Celso M de Melo, and Rama Chellappa. Synthetic-to-real domain adaptation for action recognition: A dataset and baseline performances. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

[45] H. Saito, N. Inamoto, and S. Iwase. Sports scene analysis and visualization from multiple-view video. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, pages 1395–1398 Vol.2, 2004.

[46] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 55(13s):1–37, 2023.

[47] Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. *arXiv preprint arXiv:2302.02408*, 2023.

[48] Anshul Shah, Aniket Roy, Ketul Shah, Shlok Mishra, David Jacobs, Anoop Cherian, and Rama Chellappa. HaLP: Hallucinating latent positives for skeleton-based self-supervised learning of actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18846–18856, 2023.

[49] Ketul Shah, Anshul Shah, Chun Pong Lau, Celso M de Melo, and Rama Chellappa. Multi-view action recognition using contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3381–3391, 2023.

[50] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[51] Xiangbo Shu, Jiawen Yang, Rui Yan, and Yan Song. Expansion-squeeze-excitation fusion network for elderly activity recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(8):5281–5292, 2022.

[52] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

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

[54] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[55] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[57] Shruti Vyas, Yogesh S Rawat, and Mubarak Shah. Multiview action recognition using cross-view video prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 427–444. Springer, 2020.

[58] Dongang Wang, Wanli Ouyang, Wen Li, and Dong Xu. Dividing and aggregating network for multi-view action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 451–467, 2018.

[59] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2649–2656, 2014.

[60] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4006–4015, 2019.

[61] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 504–521. Springer, 2020.

[62] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023.

[63] 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*, pages 14733–14743, 2022.

[64] 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*, pages 6312–6322, 2023.

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

[66] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 20–27. IEEE, 2012.

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

[68] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C. Kot. Skeleton cloud colorization for unsupervised 3d action representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13423–13433, 2021.

[69] Senthil Yogamani, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O'Dea, Michal Uricár, Stefan Milz, Martin Simon, Karl Amende, et al. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9308–9318, 2019.

[70] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Adding attentiveness to the neurons in recurrent neural networks. In *proceedings of the European conference on computer vision (ECCV)*, pages 135–151, 2018.

[71] Jingjing Zheng, Zhuolin Jiang, P Jonathon Phillips, and Rama Chellappa. Cross-view action recognition via a transferable dictionary pair. In *bmvc*, page 7, 2012.

[72] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox. Crossclr: Cross-modal contrastive learning for multi-modal video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1450–1459, 2021.

Appendix

A. Architecture details

The detailed asymmetric architecture of the encoder and decoders is shown in Table 11, Table 12 and Table 13. We have two decoders in our architecture: 1) self-view decoder and 2) cross-view decoder. The self-view decoder only uses self-attention to reconstruct same view whereas the cross-view decoder uses cross-attention in addition to self-attention for reconstructing target viewpoint while also using source viewpoint. These decoders are discarded during fine-tuning. We use 16 frame input and choose ViT-S/16 as our default encoder. We adopt the joint space-time attention [2] for the encoder.

Stage	Vision Transformer (Small)	Output Sizes
data	stride $4 \times 1 \times 1$	$3 \times 16 \times 128 \times 128$
cube	$2 \times 16 \times 16$, 384 stride $2 \times 16 \times 16$	$384 \times 8 \times 64$
mask	random mask *mask ratio = ρ*	$384 \times 8 \times [64 \times (1-\rho)]$
encoder	MHA(384) MLP(1536) $\times 12$	$384 \times 8 \times [64 \times (1-\rho)]$
projector	MLP(192) & *concat learnable tokens*	$192 \times 8 \times 64$

Table 11. **Encoder of MV2MAE.** The encoder processes 16-frame input clips from source and target views, and the encoded representations of the visible tokens are combined with the learnable mask tokens, before passing through the decoder.

Stage	Transformer	Output Sizes
self-view decoder	MHA(192) MLP(768) $\times 4$	$192 \times 8 \times 64$
projector	MLP(1536)	$1536 \times 8 \times 64$
reshape	*from 1536 to $3 \times 2 \times 16 \times 16$*	$3 \times 16 \times 128 \times 128$

Table 12. **Self-view decoder of MV2MAE.** It takes the source and target view tokens and reconstructs both the views independently.

Stage	Transformer	Output Sizes
cross-view decoder	MHCA(192) MHA(192) MLP(768) $\times 4$	$192 \times 8 \times 64$
projector	MLP(1536)	$1536 \times 8 \times 64$
reshape	*from 1536 to $3 \times 2 \times 16 \times 16$*	$3 \times 16 \times 128 \times 128$

Table 13. **Cross-view decoder of MV2MAE.** The cross-view decoder uses the visible tokens from the source view to reconstruct the missing patches in the target view. This cross-view information is pulled in using the cross-attention block (MHCA).

config	NTU60	NTU120	ETRI
optimizer		AdamW	
base learning rate		1e-3	
weight decay		0.05	
optimizer momentum		$\beta_1, \beta_2 = 0.9, 0.95$	
batch size		1024	
learning rate schedule		cosine decay	
warmup epochs	320	160	160
total epochs	3200	1600	1600
augmentation		MultiScaleCrop	

Table 14. **Pre-training setting.**

config	NTU60	NTU120	ETRI
optimizer		AdamW	
base learning rate		1e-3	
weight decay		0.1	
optimizer momentum		$\beta_1, \beta_2 = 0.9, 0.999$	
batch size		1024	
learning rate schedule		cosine decay	
warmup epochs	5	10	10
training epochs	35	120	120
repeated augmentation		6	
flip augmentation		*yes*	
RandAug		(7, 0.5)	
label smoothing		0.1	
drop path		0.1	
layer-wise lr decay		0.9	

Table 15. **End-to-end fine-tuning setting**

B. Implementation details

The pre-training and fine-tuning hyper-parameter settings for NTU-60, NTU-120 and ETRI datasets are given in Table 14 and Table 15.

C. Comparison with VideoMAE

VideoMAE [54] proposes to use the tube masking for dealing with the temporal redundancy in videos. Instead, we use a motion re-weighted reconstruction loss to deal with this issue. Table 16 shows that our approach to tackle temporal redundancy is superior to using tube masking.

Method	NTU-120 xsub (%)
Tube masking (VideoMAE)	79.7
Motion-weighted rec. loss (MV2MAE)	84.8

Table 16. Motion-weighted reconstruction loss in MV2MAE is a more effective way of combating temporal redundancy in videos compared to tube masking of VideoMAE.

D. Single-View vs Multi-View Inference

At test time, multiple viewpoints of an activity are available in the cross-subject setting. However, evaluation in prior work is carried out using single-view at a time, following the original benchmark [50]. Though in most practical scenarios, it would be natural to combine the predictions from available synchronized viewpoints for a given activity. We show this comparison of single-view and multi-view inference in Table 17. For multi-view inference, the predictions are combined using late fusion strategy.

Method	Cross-Subject (%)	
	NTU-60	NTU-120
Single-View Inference	90.0	85.3
Multi-View Inference	91.9	87.9

Table 17. SV vs MV inference. We perform late fusion for multi-view inference.

E. Synthetic data for pre-training

Real multi-view videos can be difficult to acquire and can pose privacy concerns. Here we explore using synthetic multi-view action recognition data as an alternative. In these experiments, the pre-training is done on synthetic data (SynADL [25] dataset) while fine-tuning and inference is done on the real data (ETRI [27] dataset). We compare synthetic pre-training (green) with real pre-training (orange). We observe that if the amount of synthetic data used is same (1x) as the amount of real data, there is a performance drop due to the domain difference. Interestingly, if we increase the amount of synthetic data used for pre-training, synthetic pre-training can outperform real pre-training, as seen in Figure 6.
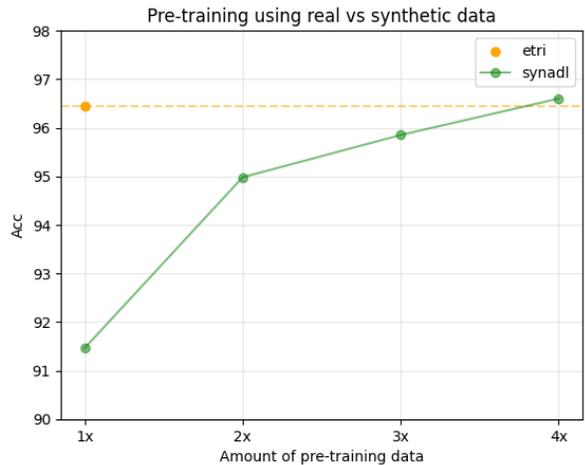

Figure 6. **Pre-training using synthetic data.** Pre-training using more (4x) synthetic data beats pre-training using real data on the same real test set.