Time Is MattEr: Temporal Self-supervision for Video Transformers

Sukmin Yun [1] **Jaehyung Kim** [1] **Dongyoon Han** [2] **Hwanjun Song** [2] **Jung-Woo Ha** [2] **Jinwoo Shin** [1 3]

Abstract

Understanding temporal dynamics of video is an essential aspect of learning better video representations. Recently, transformer-based architectural designs have been extensively explored for video tasks due to their capability to capture long-term dependency of input sequences. However, we found that these Video Transformers are still biased to learn spatial dynamics rather than temporal ones, and debiasing the spurious correlation is critical for their performance. Based on the observations, we design simple yet effective self-supervised tasks for video models to learn temporal dynamics better. Specifically, for debiasing the spatial bias, our method learns the temporal order of video frames as extra self-supervision and enforces the randomly shuffled frames to have low-confidence outputs. Also, our method learns the temporal flow direction of video tokens among consecutive frames for enhancing the correlation toward temporal dynamics. Under various video action recognition tasks, we demonstrate the effectiveness of our method and its compatibility with state-of-the-art Video Transformers.

1. Introduction

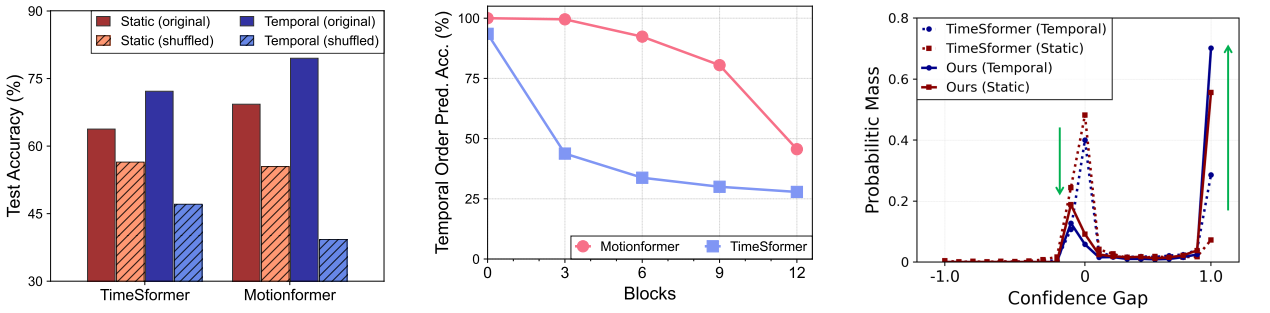
Understanding videos for action or event recognition is a challenging yet crucial task that has gotten significant attention in computer vision communities (Lin et al., 2019; Cheng et al., 2021). Compared to image data, temporal dynamics between video frames provide additional information that is essential for recognition, as the actions and events generally occur over multiple consecutive frames. Hence, designing video-specific architectures for capturing such temporal dynamics has been a common theme in learning better video representations (Simonyan & Zisserman, 2014; Tran et al., 2015; 2018; Feichtenhofer et al., 2019). Recently, Transformer-based (Vaswani et al., 2017) video models, so-called *Video Transformers*, have been extensively explored due to their capability to capture long-term dependency among the input sequence; for example, Bertasius et al. (2021) and Patrick et al. (2021) introduce divided space-time and trajectory attentions, respectively.

However, it is still questionable *whether these architectural advances are enough to fully capture the temporal dynamics in a video.* For example, Fan et al. (2021) shows that a well-designed image classifier with a flattened video along time dimension could outperform the several state-of-the-art video models on the representative action recognition tasks. As additional clues, we observed that Video Transformers often predict a video action correctly with high confidence even when input video frames are randomly shuffled, *i.e.,* the shuffled video does not contain correct temporal dynamics (see Figure 1(a) and 1(c)). Furthermore, as shown in Figure 1(b), Video Transformers also fail to capture the temporal order of video frames as their layers go deeper. These observations reveal that the recent Video Transformers are likely to be biased to learn spatial dynamics, despite their efforts of designing a better architecture for learning the temporal one. Hence, this limitation inspires us to investigate an independent and complementary direction other than architectural advance, to improve the quality of learned video representations via better temporal modeling.

Contribution. In this paper, we design simple yet effective *frame-* and *token-level* self-supervised tasks, coined TIME (**T**ime **I**s **M**att**E**r), for video models which learn temporal dynamics better. First, we train the models to learn two frame-level tasks for debiasing the spurious correlation learned from spatial dynamics. Specifically, (a) we keep the temporal information within the video frame-by-frame by assigning the correct frame order as a self-supervised label to predict the temporal order of video frames (to avoid losing the temporal information), and (b) we simultaneously train the video models to be not able to output high-confident predictions when the input video does not contain the correct temporal order, *i.e.,* randomly shuffled video frames. Moreover, we train the models with a token-level task for enhancing the correlation toward temporal dynamics by predicting the temporal flow direction of video tokens among consecutive frames. To be specific, we adapt an attention-

[1]School of Electrical Engineering, KAIST, South Korea [2]NAVER AI Lab, South Korea [3]Graduate School of AI, KAIST, South Korea. Correspondence to: Sukmin Yun <sukmin.yun@kaist.ac.kr>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.



(a) Spurious correlation on spatial dynamics (b) Vanishing temporal information (c) Debiasing via temporal self-supervision

Figure 1. Experimental results on SSv2 test dataset supporting our motivation. (a) Comparison of the accuracy for original and shuffled videos, respectively. Two different types of classes, *Static* and *Temporal*, are used to consider the relatively different importance of temporal information, following Sevilla-Lara et al. (2021). Here, high accuracy is retained after shuffling frames, due to the models' bias toward spatial dynamics. (b) Variation of temporal information within each Transformer block, measured with the accuracy of temporal order prediction; temporal information vanishes as the blocks deeper. (c) Probabilistic mass of the subset samples based on their confidence gap between original and shuffled videos. 20 bins are used to calculate the probabilistic mass. One could observe that the overall confidence gap is significantly increased from the proposed temporal self-supervised tasks, *i.e.*, the model is successfully debiased.

based module on the final representations of tokens in consecutive frames to predict their nine types of flow direction in the time axis (eight angular directions and the center; see Eq.(10)) by assigning their self-supervised labels obtained by Gunnar Farnebäck's algorithm (Farnebäck, 2003). We provide an overall illustration of our scheme in Figure 2. It is worth noting that our scheme can be adopted to any Video Transformers in a plug-in manner and is beneficial to various video downstream tasks including action recognition without additional human-annotated supervision. Somewhat interestingly, our approach also can be extended to the image domain for alleviating background bias.

To demonstrate the effectiveness of the proposed temporal self-supervised tasks, we incorporate our method with various Video Transformers and mainly evaluate on Something-Something-v2 (SSv2) (Goyal et al., 2017) benchmark.[1] Despite its simplicity, our method consistently improves the recent state-of-the-art Video Transformers by debiasing the spurious correlation between spatial and temporal dynamics successfully. For example, ours improves the accuracy of TimeSformer (Bertasius et al., 2021) and X-ViT (Bulat et al., 2021) from 62.1% to 63.7% (+1.6%) and 60.1% to 63.5% (+3.4%) for SSv2, respectively. Furthermore, we also found that our self-supervised idea of debiasing can be naturally extended to image classification models to reduce the spurious correlation learned from the image backgrounds, *e.g.*, our method improves the model generalization and robustness on ImageNet-9 (Xiao et al., 2021). For example, our idea not only improves DeiT-Ti/16 (Touvron et al., 2020) from 77.3% to 83.3% (+6.0%) on the original dataset but also from 50.3% to 58.9% (+8.6%) on the Only-FG (*i.e.*, remaining foregrounds only and removing backgrounds) in the background shift benchmarks (Xiao et al., 2021).

Overall, our work highlights the importance of debiasing the spurious correlation of visual transformer models with respect to the temporal or spatial dynamics. We believe our work could inspire researchers to rethink the under-explored, yet important problem and provide a new research direction.

[1] As Sevilla-Lara et al. (2021) stated, Something-Something-v2 (Goyal et al., 2017) is known to contain a larger proportion of temporal classes requiring temporal information to be recognized. On the other hand, Kinetics (Kay et al., 2017) is not the case, where it is arguably much less suitable for modeling temporal dynamics well, *e.g.,* see Figure 3 and Section 5 for more details. Hence, our method should have marginal gains on Kinetics.

2. Related Work

Architectural advances for video action recognition. 3D convolutional neural networks (3D-CNNs) were originally considered to learn deep video representations by inflating pre-trained ImageNet weights (Carreira & Zisserman, 2017). 3D-CNN models (Tran et al., 2018; Feichtenhofer et al., 2019) extract spatio-temporal representations via their own temporal modeling methods; for example, SlowFast (Feichtenhofer et al., 2019) captures short- and long-range of time dependencies by using two different speed of pathways for the video. However, such 3D convolutional designs are limited to capture long-term dependency of video with its small receptive field.

Due to the ability to capture long-term dependency of the self-attention mechanism, transformer-based models (Neimark et al., 2021; Bertasius et al., 2021; Arnab et al., 2021; Patrick et al., 2021; Bulat et al., 2021; Fan et al., 2021;

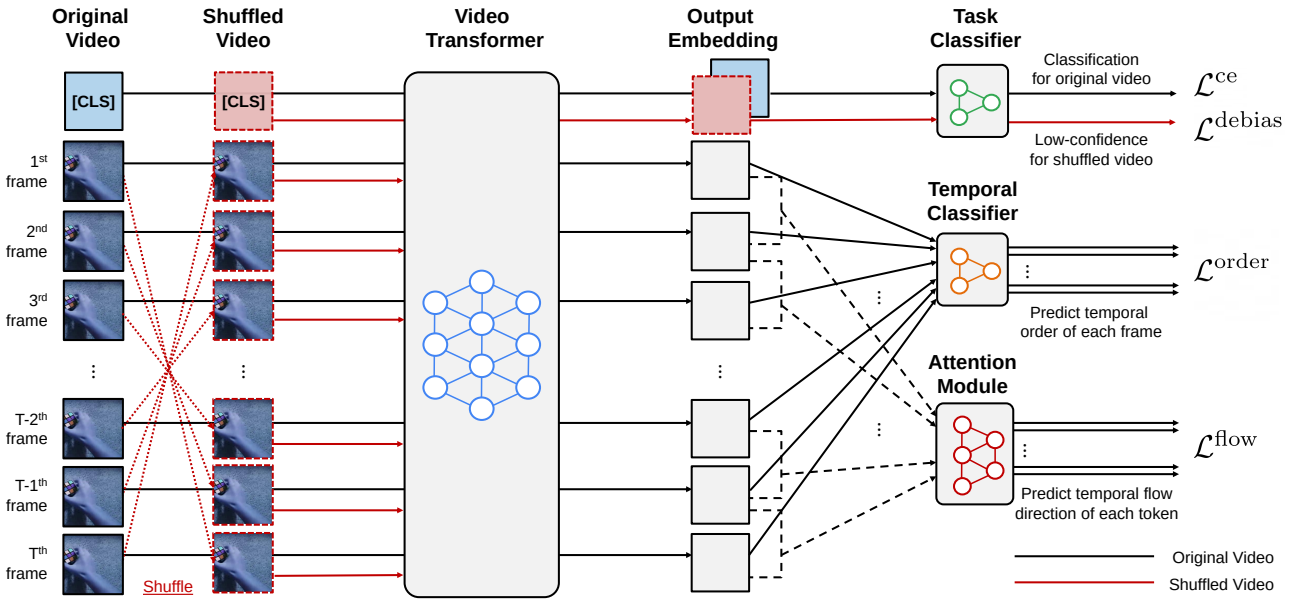


Figure 2. Illustration of the proposed scheme, TIME. We use three types of temporal self-supervision for learning better video representations by (a) reducing a risk of learning the spurious correlation from spatial dynamics (*i.e.,* $\mathcal{L}^{\text{debias}}$), (b) keeping temporal order information in deeper blocks (*i.e.,* $\mathcal{L}^{\text{order}}$), and (c) enhancing the correlation toward temporal dynamics (*i.e.,* $\mathcal{L}^{\text{flow}}$). Specifically, we train the model to (a) output low-confident predictions for shuffled videos, (b) predict the correct frame order of videos, and (c) predict nine types of temporal flow direction (eight angular directions and the center; see Eq.(10)) of video tokens in the consecutive frames.

Li et al., 2022) have been recently explored for video action recognition by following the success of Vision Transformer (Dosovitskiy et al., 2021), which has shown competitive performances against CNNs in image domains. In particular, Video Transformers such as TimeSformer (Bertasius et al., 2021) and ViViT (Arnab et al., 2021) propose the use of a temporal attention module with the existing ViT to better understand temporal dynamics of videos.

Besides, several works attempt to develop a more efficient and powerful attention module to mitigate the quadratic complexity of the self-attention in learning from videos. For example, TimeSformer suggests divided-space-time attention module by decomposing the time and space dimensions separately, X-ViT (Bulat et al., 2021) further restricts time attention to a local temporal window, *i.e.,* local space-time attention. Motionformer (Patrick et al., 2021) introduces a trajectory attention to model the probabilistic path of a token between frames over the entire space-time feature volume.

Static biases in video. In video action recognition, it is essential capturing long-term dependency of temporal dynamics. However, several prior works (Li & Vasconcelos, 2019; Sevilla-Lara et al., 2021; Huang et al., 2018) have shown that video models are often biased to learn spatial dynamics rather than the temporal one due to the presence of static classes, which contain informative frames to predict the action class labels without understanding overall temporal information. In particular, Sevilla-Lara et al. (2021) categorizes action classes in video datasets as temporal and static classes with respect to whether they necessarily require understanding temporal information to recognize them, and shows that training on temporal classes can lead video models to avoid spatial bias and generalize better.

In the end, most recent works have mainly focused on architectural advances to capture temporal dynamics in videos. On the contrary, our method aims to handle this issue by designing self-supervised tasks not only for capturing stronger temporal dynamics, but also for debiasing the spurious correlation learned from spatial dynamics. Meanwhile, some CNN-based works (Misra et al., 2016; Lee et al., 2017; Hu et al., 2021) have also adapted a binary classification task that predicts the correct order of randomly chosen video frames or clips, however, our idea is simple and computationally efficient for video transformers; their capabilities to capture long-term dependency enable us to infer all the absolute temporal order frame-by-frame, while the prior works under CNNs do the binary order one-by-one.

3. Motivation: Bias toward Spatial Dynamics

3.1. Preliminaries: Video Transformers

Let $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_T] \in \mathbb{R}^{T \times H \times W \times C}$ be a video where (H, W) is the resolution of video $\mathbf{x}$ with T frames, and

Model	Input frame	Tokenization	Top-1 ↑	Top-5 ↑	Shuffled Top-1 ↓	Shuffled Top-5 ↓
TimeSformer	8	$1 \times 16 \times 16$	59.1	85.6	46.4	78.8
TimeSformer-HR	16	$1 \times 16 \times 16$	61.8	86.9	49.7	81.5
TimeSformer-L	64	$1 \times 16 \times 16$	62.0	87.5	55.1	84.7
Motionformer	16	$2 \times 16 \times 16$	66.5	90.1	43.9	75.6
Motionformer-L	32	$2 \times 16 \times 16$	68.1	91.2	40.7	73.3

Table 1. Evaluation of pre-trained Video Transformers on SSv2 dataset. Top-1 and Top-5 denote top-1 and top-5 test accuracy with the original input video, respectively. Shuffled Top-1 and Shuffled Top-5 denote the accuracy of frame-shuffled videos, respectively.

C is the number of channels. Video Transformers treat the input video $\mathbf{x}$ as a sequence of hwt tokens $\{\mathbf{x}^{(i)} \in \mathbb{R}^{(T/t)\times(H/h)\times(W/w)\times C}\}_{i=1}^{hwt}$. Then, the tokens are linearly transformed to D-dimensional patch embeddings $\mathbf{e}^{(i)} = E\mathbf{x}^{(i)} + E_{\text{pos}}^{(i)} \in \mathbb{R}^D$ where $E \in \mathbb{R}^{D\times(THWC/thw)}$ is a linear projection and $E_{\text{pos}}^{(i)} \in \mathbb{R}^D$ is the positional embedding for the i-th token $\mathbf{x}^{(i)}$. Video Transformers also prepend [CLS] token, which is used to represent the entire sequence of tokens (*i.e.,* the given video $\mathbf{x}$), to the token sequence with a learnable embedding $\mathbf{e}^{[\texttt{CLS}]} \in \mathbb{R}^D$. Overall, the resulting sequence of the input video $\mathbf{x}$ is $\mathbf{e} = [\mathbf{e}^{[\texttt{CLS}]}; \mathbf{e}^{(1,1)}; \mathbf{e}^{(1,2)}; \ldots; \mathbf{e}^{(s,t)}]$, where $s = hw$. Then, Video Transformers take the sequence $\mathbf{e}$ as inputs and then output the same size contextualized embeddings with their own spatial-temporal attention module.

Spatio-temporal attentions. Spatio-temporal attention is an extension of the self-attention that operates over space and time dimensions in parallel. For a video input sequence $\mathbf{e} \in \mathbb{R}^{st\times D}$ with a space-time location st, joint spatio-temporal attention is a natural extension that can be defined:

$$\mathbf{q}_{st} = \mathbf{e}Q, \mathbf{k}_{st} = \mathbf{e}K, \mathbf{v}_{st} = \mathbf{e}V, \tag{1}$$

$$\text{(joint space-time)} \sum_{s't'} \mathbf{v}_{s't'} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{s't'}\rangle}{\sum_{\bar{s}\bar{t}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{\bar{s}\bar{t}}\rangle}, \tag{2}$$

where $Q, K, V \in \mathbb{R}^{D\times D}$ are query, key, and value matrices. Here, the joint attention computes attentions of all keys $\mathbf{k}_{s't'}$ for each query $\mathbf{q}_{st}$, and it has a limitation of quadratic complexity in both space and time, *i.e.,* $\mathcal{O}(s^2t^2)$. To address this limitation, several Video Transformers (Bertasius et al., 2021; Arnab et al., 2021) propose *divided attention*, which restricts space or time dimension as below:

$$\text{(space only)} \sum_{s'} \mathbf{v}_{s't} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{s't}\rangle}{\sum_{\bar{s}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{\bar{s}t}\rangle}, \tag{3}$$

$$\text{(time only)} \sum_{t'} \mathbf{v}_{st'} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{st'}\rangle}{\sum_{\bar{t}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{s\bar{t}}\rangle}. \tag{4}$$

The divided attention reduces the complexity to $\mathcal{O}(s^2t) + \mathcal{O}(st^2)$, and TimeSformer (Bertasius et al., 2021) and ViViT (Arnab et al., 2021) utilize this approach alternately for getting spatio-temporal features. On the other hand, Motionformer (Patrick et al., 2021) also introduces trajectory attention, which is designed to capture temporal dynamics by modeling a set of trajectory tokens computed across the frames, with a complexity of $\mathcal{O}(s^2t^2)$.

For conciseness, we use f_θ to denote the whole process of a Video Transformers parameterized by θ as follows:[2]

$$\begin{aligned} f_\theta(\mathbf{x}) &= f_\theta([\mathbf{e}^{[\texttt{CLS}]}; \mathbf{e}^{(1,1)}; \mathbf{e}^{(1,2)}; \ldots; \mathbf{e}^{(s,t)}]) \\ &:= [f_\theta^{[\texttt{CLS}]}(\mathbf{x}); f_\theta^{(1,1)}(\mathbf{x}); \ldots; f_\theta^{(s,t)}(\mathbf{x})], \end{aligned} \tag{5}$$

where $f_\theta^{[\texttt{CLS}]}(\mathbf{x})$ and $f_\theta^{(i,j)}(\mathbf{x})$ are the final representations of the [CLS] token and the (i,j)-th token, respectively. We remark that $f_\theta^{[\texttt{CLS}]}(\mathbf{x})$ is generally utilized for solving video-level downstream tasks such as action recognition with a linear classifier head g_θ.

3.2. Observations

In this section, we describe our empirical observations based on the recent Video Transformers, such as TimeSformer (Bertasius et al., 2021) and Motionformer (Patrick et al., 2021), trained to recognize the actions in video using SSv2 dataset (Goyal et al., 2017). Here, our observations reveal that even the recent state-of-the-art video models still struggle to fully exploit the temporal information in video data. These findings serve as a key intuition for designing our temporal self-supervised tasks for video models.

Spurious correlation on spatial dynamics. Our first observation is that the violation of temporal dynamics within video does not lead to the low confident predictions of Video Transformers. Intuitively, if the models are learned to recognize the actions via capturing the temporal dynamics between the frames, their predictions should have low confidence when the input video does not have the correct temporal order, *e.g.*, frames are randomly shuffled (Misra et al., 2016; Sevilla-Lara et al., 2021). To verify such behavior, we measure the accuracy of Video Transformers on SSv2 test dataset with original and shuffled videos in Table 1; the shuffled video $\widetilde{\mathbf{x}}$ is constructed from the original video

[2]Note that θ contains all the parameters of the transformer-based model including embedding parameters E, E_{pos}, and $\mathbf{e}^{[\texttt{CLS}]}$.

Model	Top-1	Top-5
TimeSformer (Bertasius et al., 2021)	62.1	86.4
TimeSformer + TIME	**63.7**	**87.8**
Motionformer (Patrick et al., 2021)	63.8	88.5
Motionformer + TIME	**64.7**	**89.3**
X-ViT (Bulat et al., 2021)	60.1	85.2
X-ViT + TIME	**63.5**	**88.1**

Table 2. Video action recognition performance of the recent Video Transformers. All models share the same training details, and they are fine-tuned on the SSv2 dataset from the ImageNet-1k pre-trained weights. Top-1 and -5 denote test accuracies, respectively.

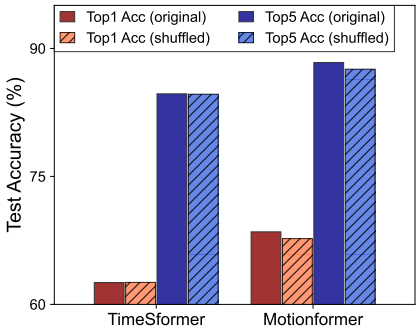


Figure 3. Top-1 and top-5 test accuracy on Kinetics-400 (Kay et al., 2017) with original and shuffled videos, respectively.

$\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ where $\widetilde{\mathbf{x}} = [\mathbf{x}_{p(1)}, \ldots, \mathbf{x}_{p(T)}]$ and $p(i)$ is a permuted order. Here, it is observed that the models generally achieve the high test accuracy regardless of the shuffling of videos; for example, under 64 frames video, TimeSformer achieves 62.0% accuracy with only 6.9% reduction compared to the accuracy of the original video. We note that such high confident predictions on the violation of temporal dynamics (*i.e.,* incorrect temporal order) often occur even for *temporal classes* where temporal information is crucial to recognize them (Sevilla-Lara et al., 2021) (see Figure 1(a)). These results indicate that the Video Transformers are likely to be biased to learn the spatial dynamics rather than temporal one despite their specific architectural designs to learn temporal information better.[3]

Vanishing temporal information. Next, we further observe that the deeper the transformer blocks in Video Transformers even fail to keep the temporal order of video frames. Specifically, we first generate temporal labels $\mathbf{y}_{\text{order}}^{(j)} = j$ as the temporal order of j-th frames. Then, we train additional linear classifier g_θ^{order} using $\mathcal{L}^{\text{order}}$ to predict $\mathbf{y}_{\text{order}}^{(j)}$ based on the frozen output embeddings of input video $\mathbf{x}$ as follow:

$$\bar{f}_\theta^{(j)}(\mathbf{x}) := \frac{1}{s}\sum_{i=1}^{s} f_\theta^{(i,j)}(\mathbf{x}), \tag{6}$$

$$\mathcal{L}^{\text{order}}(\mathbf{x}) := \frac{1}{t}\sum_{j=1}^{t} \text{CE}(g_\theta^{\text{order}}(\bar{f}_\theta^{(j)}(\mathbf{x})), \mathbf{y}_{\text{order}}^{(j)}), \tag{7}$$

where $\bar{f}_\theta^{(j)}(\mathbf{x})$ is an aggregated embedding of $f_\theta^{(i,j)}(\mathbf{x})$ across the space axis, and $\text{CE}(\mathbf{x}, \mathbf{y})$ denotes the standard cross-entropy loss between an input $\mathbf{x}$ and a given label $\mathbf{y}$, respectively. To track the change of temporal information within each transformer block, we train a linear classifier g_θ^{order} on the aggregated embedding of each block and compare the accuracy of the temporal order prediction. As shown in Figure 1(b), the earlier blocks show much higher accuracy, but the performance significantly decreases in the latter blocks; for example, Motionformer achieves 99.5% accuracy at the 3th block, but it decreases to 45.6% at the last block. It might be the results of the learned spurious correlation of video models; as the models are focused on learning the spatial information, the temporal information would be less captured and lost.

Overall, these empirical observations reveal that only designing better video model architecture may not be enough; [4] hence, it motivates us to investigate the independent yet complementary direction for improving the quality of learned video representation.

4. TIME: Temporal Self-supervision for Video

Motivated by the previous observations in Section 3.2, we introduce a simple yet effective self-supervised tasks, coined TIME (**T**ime **I**s **M**att**E**r), to better understand temporal dynamics of videos, which can be beneficial for video recognition in a model-agnostic way. Overall illustration of the proposed scheme is presented in Figure 2.

Debiasing spatial dynamics. First, we reduce a risk of learning the spurious correlation from spatial dynamics by utilizing the shuffled (*i.e.,* temporarily incorrect) for training; we train the video models to output the low confident predictions for the shuffled video. Specifically, we minimize the Kullback-Leibler (KL) divergence from the predictive distribution on randomly shuffled video $\widetilde{\mathbf{x}}$ to the uniform one in order to give less confident predictions as follows:

$$\mathcal{L}^{\text{debias}}(\widetilde{\mathbf{x}}) := \text{KL}\left(\mathcal{U}(\mathbf{y}) \middle\| \texttt{Softmax}(g_\theta(f_\theta^{[\texttt{CLS}]}(\widetilde{\mathbf{x}})))\right), \tag{8}$$

where KL is a the Kullback-Leibler (KL) divergence, `Softmax` is a softmax function, g_θ is a linear classification head, and $\mathcal{U}(\mathbf{y})$ is the uniform distribution. As it prevents

[3] Such spatial bias in video models may come from the ImageNet pre-trained weights. However, even without the pre-trained weights, we empirically found that our method still achieves significant improvements in training from scratch (see Appendix C).

[4] Architectural modifications could be an alternative for keeping temporal information. Nevertheless, we empirically found that our method could further improve the performance (see Appendix D).

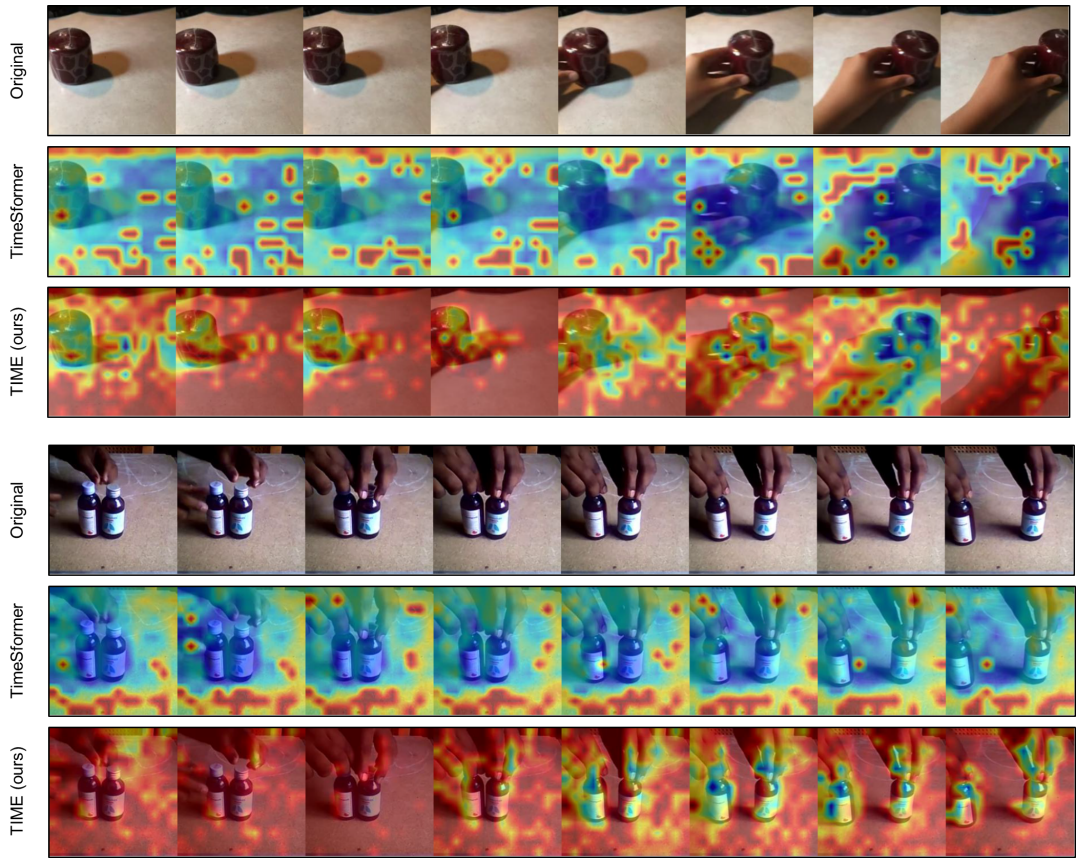


Figure 4. Visualization of learned video models via GradCAM (Selvaraju et al., 2017). Here, we present eight frame input videos that come from *pushing something from left to right* class (**Top**) and *moving something and something away from each other* class (**Bottom**) in the SSv2 test dataset, respectively. Video models are fine-tuned on the SSv2 dataset from the ImageNet-1k pre-trained weights. While TimeSformer fails to focus on the object, TimeSformer + TIME (ours) is successfully tracking its trajectory. Best viewed in color.

the model from biased predictions toward spatial dynamics, the model learns to exploit the temporal dynamics better. We remark that the similar approaches have been utilized in other domains, such as image (Lee et al., 2018; Hendrycks et al., 2019) and language (Moon et al., 2021).

Learning temporal order of frames. To avoid losing temporal order information in deeper blocks, we directly regularize the model to keep such information at the final block. Specifically, we simply add a linear classifier g_θ^{order} on the final aggregated embedding $\bar{f}_\theta^{(j)}(\mathbf{x})$ of each frame and train the model using $\mathcal{L}^{\text{order}}$ (7) to predict the temporal order of input video frames as described in Section 3.2. This regularization preserves the temporal information until the end of blocks; hence, the model could utilize it for solving the target task, such as action recognition.

Learning temporal flow direction of tokens. To enhance the correlation toward temporal dynamics, we train the model to predict the temporal flow direction of tokens in the consecutive frames $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$. To be specific, we adopt an attention-based module h_θ on pairs of final representations of the consecutive frames $\hat{f}^{(j;j+1)}(\mathbf{x})$, and train the model using $\mathcal{L}^{\text{flow}}$ to predict their nine types of flow direction in the time axis (eight angular directions and the center) by assigning self-supervised labels $\mathbf{y}_{\text{flow}}^{(i,j)}$ obtained by Gunnar Farnebäck's algorithm (Farnebäck, 2003) as follow:

$$\{r^{(i,j)}, \phi^{(i,j)}\}_i := \texttt{Polar}\left(\texttt{Farneback}(\mathbf{x}_j, \mathbf{x}_{j+1})\right), \quad (9)$$

$$\mathbf{y}_{\text{flow}}^{(i,j)} := \begin{cases} \lfloor \phi^{(i,j)} \cdot \frac{4}{\pi} \rfloor + 1 & \text{if } r^{(i,j)} \geq \tau, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

$$\hat{f}^{(j;j+1)}(\mathbf{x}) := (\{f_\theta^{(i,j)}(\mathbf{x})\}_i, \{f_\theta^{(i,j+1)}(\mathbf{x})\}_i), \quad (11)$$

$$h_\theta(\hat{f}^{(j;j+1)}(\mathbf{x})) := [h_\theta^{(1,j)}(\mathbf{x}); \dots; h_\theta^{(s,j)}(\mathbf{x})], \quad (12)$$

$$\mathcal{L}^{\text{flow}}(\mathbf{x}) := \frac{1}{s(t-1)} \sum_{j=1}^{t-1} \sum_{i=1}^{s} \text{CE}(h_\theta^{(i,j)}(\mathbf{x})), \mathbf{y}_{\text{flow}}^{(i,j)}), \quad (13)$$

where $r^{(i,j)} > 0$ and $0 \leq \phi^{(i,j)} < 2\pi$ are the magnitude and angle obtained from the frames $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$ via the polarization function `Polar` and the Gunnar Farnebäck's algorithm `Farnebäck`, and τ is a threshold for the center and angular directions of self-supervised labels $\mathbf{y}_{\text{flow}}^{(i,j)}$.

$\mathcal{L}^{\text{order}}$	$\mathcal{L}^{\text{debias}}$	$\mathcal{L}^{\text{flow}}$	SSv2 dataset			Temporal subset			Static subset		
			Original ↑	Shuffled ↓	Gap ↑	Original ↑	Shuffled ↓	Gap ↑	Original ↑	Shuffled ↓	Gap ↑
✗	✗	✗	62.1	41.3	20.8	84.9	57.0	27.9	84.1	84.1	0.0
✓	✗	✗	62.6	39.7	22.9	88.0	51.4	36.6	85.3	83.2	2.1
✗	✓	✗	62.7	**10.9**	**51.8**	88.1	18.8	69.5	84.6	84.5	0.1
✓	✓	✗	63.2	30.4	32.8	90.0	**18.6**	**71.4**	86.3	**40.0**	**46.3**
✗	✗	✓	62.6	39.5	23.1	87.3	56.4	30.9	84.4	84.3	0.1
✓	✗	✓	62.7	40.7	22.0	88.6	52.5	36.1	85.7	83.8	1.9
✗	✓	✓	63.4	13.0	50.4	89.3	22.2	67.1	85.0	84.9	0.1
✓	✓	✓	**63.7**	25.3	38.4	**90.2**	22.1	68.1	**86.9**	69.3	17.6

Table 3. Ablation study on effect of loss components $\mathcal{L}^{\text{order}}$, $\mathcal{L}^{\text{debias}}$ and $\mathcal{L}^{\text{flow}}$. All models share the same training details and are fine-tuned from ImageNet-1k pre-trained weights. "Original" and "Shuffled" denote the top-1 accuracy of the original and shuffled input videos, respectively, and "Gap" denotes the difference between the Original and Shuffled accuracies. "SSv2 dataset", "Temporal subset," and "Static subset" denote training datasets that show relatively different importance of temporal information; configurations of the Temporal and Static subsets are reported in Appendix A. The best scores are in **bold**, and the top 3 scores are underlined.

In summary, our total training loss can be written as follows:

$$\mathcal{L}^{\text{TIME}}(\mathbf{x}, \tilde{\mathbf{x}}) := \lambda^{\text{order}}\mathcal{L}^{\text{order}}(\mathbf{x}) + \lambda^{\text{debias}}\mathcal{L}^{\text{debias}}(\tilde{\mathbf{x}}) + \lambda^{\text{flow}}\mathcal{L}^{\text{flow}}(\mathbf{x}), \quad (14)$$

$$\mathcal{L}^{\text{total}}(\mathbf{x}, \tilde{\mathbf{x}}) := \mathcal{L}^{\text{ce}}(\mathbf{x}) + \mathcal{L}^{\text{TIME}}(\mathbf{x}, \tilde{\mathbf{x}}), \quad (15)$$

where $\mathcal{L}^{\text{ce}}$ is the cross-entropy loss with a linear classification head g_θ for video recognition, λ^{order}, λ^{debias} and λ^{flow} are hyperparameters. We simply set all of them to be 1 in our experiments (see Section 5.2 for analysis on λ).

5. Experiments

In this section, we demonstrate the effectiveness of the proposed temporal self-supervision, TIME. Specifically, we incorporate TIME with the state-of-the-art Video Transformers (Bertasius et al., 2021; Patrick et al., 2021; Bulat et al., 2021) and evaluate their temporal modeling ability on Something-Something-v2 (SSv2) (Goyal et al., 2017), which contains a large proportion of temporal classes than other video dataset like Kinetics-400 (Kay et al., 2017). We also use temporal and static classes in the SSv2 dataset, stated by Sevilla-Lara et al. (2021), for validating the importance of temporal information; temporal classes necessarily require temporal information for action recognition. More details of experimental setups are described in each section.

Video Datasets. We use SSv2 (Goyal et al., 2017) datasets and its temporal and static classes following the categorization of Sevilla-Lara et al. (2021) to evaluate whether video models understand the temporal dynamics well. Notably, SSv2 is a challenging dataset that consists of ∼169k training videos and ∼25k validation videos over 174 classes; in particular, it contains a large proportion of temporal classes requiring temporal information to be recognized (Sevilla-Lara et al., 2021). To investigate behaviors of video models on relatively different importance of temporal information, we further construct "Temporal subset" and "Static subset" as the benchmarks by choosing 6 temporal classes and 16 static classes from the above temporal and static classes, respectively. We report the specific labels of the Temporal and Static subsets in Appendix A, respectively.

Meanwhile, Kinetics dataset (Kay et al., 2017) (*e.g.,* Kinetics-400) is a large-scale video dataset, which consists of ∼240k training videos and ∼20k validation videos in 400 human action categories. However, it arguably contains fewer temporal classes and is comprised of a large amount of static classes (Li & Vasconcelos, 2019; Huang et al., 2018; Sevilla-Lara et al., 2021; Fan et al., 2021); For example, Fan et al. (2021) reports that changing temporal order of kinetic videos does not drop the recognition performance, and we also found similar observations on a 10% subset of Kinetics 400; Figure 3 shows that the state-of-the-art Video Transformers (Bertasius et al., 2021; Patrick et al., 2021) have almost the same accuracy even their input video frames are randomly shuffled, unlike SSv2 in Figure 1(a). To this end, we solely use SSv2 as the main benchmark in a perspective view of validating the importance of temporal modeling.

Baselines. We consider recent Video Transformers as baselines: TimeSformer (Bertasius et al., 2021) with divided space and time attentions, Motionformer (Patrick et al., 2021) with trajectory attention, and X-ViT (Bulat et al., 2021) with space-time mixing attention. All Video Transformers in our experiments are based on ViT-B/16 (Dosovitskiy et al., 2021) (86M parameters), which consists of 12 transformer blocks with 768 embedding dimension. We denote our method built upon an existing method by "+ TIME", *e.g.,* TimeSformer + TIME. For Figure 1(a) and 1(b) and Table 1 in Section 3, we use publicly available pre-trained models and validate their ability of temporal modeling. We remark that Figure 1(a) shows the importance of temporal order information in the temporal classes, but such information vanishes as the model layers deeper as shown in Figure 1(b).

Dataset	Baseline	Baseline + $\mathcal{L}_I^{\text{order}}$	Baseline + $\mathcal{L}_I^{\text{debias}}$	Baseline + $\mathcal{L}_I^{\text{TIME}}$
Original ↑	77.3	82.0 (+4.7)	79.0 (+1.7)	**83.3 (+6.0)**
Only-FG ↑	50.3	54.2 (+3.9)	52.7 (+2.4)	**58.9 (+8.6)**
Mixed-Same ↑	68.6	72.5 (+3.9)	69.7 (+1.1)	**74.0 (+5.4)**
Mixed-Rand ↑	43.7	48.4 (+4.7)	45.1 (+1.4)	**51.0 (+7.3)**
Mixed-Next ↑	39.9	43.6 (+3.7)	40.6 (+0.7)	**46.4 (+6.5)**
BG-Gap ↓	24.8	24.1 (−0.7)	24.6 (−0.2)	**23.0 (−1.8)**

Table 4. Extension of our method to image classification models. Baseline denotes DeiT-Ti/16 (Touvron et al., 2020), and we train all models with 300 training epochs on ImageNet-9 (Xiao et al., 2021) and evaluate them for background shifts (Xiao et al., 2021). Original denotes original ImageNet-9 dataset, Only-FG, Mixed-Same, Mixed-Rand and Mixed-Next denote variation of ImageNet-9 by shifting image background; Only-FG remains only foregrounds and removing backgrounds, Mixed-Same, -Rand and -Next changes backgrounds to random backgrounds from the same, a random, and the next class, respectively. BG-Gap denotes the difference between Mixed-Same and Mixed-Rand. Values in parenthesis are the performance difference between Baseline and Baseline incorporated with each loss term.

Implementation details. In our experiments, we unify different training details of the recent Video Transformers (Bertasius et al., 2021; Patrick et al., 2021; Bulat et al., 2021), and re-implement all the baselines on our setups for a fair comparison.[5] Specifically, we fine-tune all the models from ImageNet (Deng et al., 2009) pre-trained weights of ViT-B/16 (Dosovitskiy et al., 2021) for 35 training epochs with Adamw optimizer (Loshchilov & Hutter, 2018) and learning rate of 0.0001 and a batch size of 64. For data augmentation, we follow RandAugment (Cubuk et al., 2020) policy of Patrick et al. (2021). We use the spatial resolution of 224×224 with patch size of 16×16, and eight frame input videos under the same $1 \times 16 \times 16$ tokenization method, including Motionformer. We set all the loss weights to be 1 (*i.e.,* $\lambda^{\text{order}} = \lambda^{\text{debias}} = \lambda^{\text{flow}} = 1$) unless stated otherwise.

5.1. Temporal modeling on SSv2

In this section, we evaluate our method on the SSv2 benchmark using eight frame input videos by incorporating with the state-of-the-art Video Transformers; TimeSformer (Bertasius et al., 2021), Motionformer (Patrick et al., 2021), and X-ViT (Bulat et al., 2021). Under the same training details; *e.g.,* optimizer, training scheduling, augmentation policy and tokenization, as shown in Table 2, we found that our method, TIME, consistently improves all the backbone architectures with a large margin. For example, TimeSformer + TIME and X-ViT + TIME achieve 1.6% and 3.4% higher top-1 accuracies compared to their baselines TimeSformer (62.1%) and X-ViT (60.1%), respectively. These results not only show the effectiveness of TIME but also demonstrate the high architectural compatibility of TIME and allow us to conjecture that ours can overcome failure modes in the Video Transformers. One can further verify the advantage of our scheme from the provided qualitative examples in Figure 4. Here, with better temporal modeling from the proposed self-supervised tasks, the model can capture the temporal dynamics in a better way. More examples and details are in Appendix B.

[5]Code is available at https://github.com/alinlab/temporal-selfsupervision.

5.2. Ablation study

In this section, we perform an ablation study to understand further how TIME works. Specifically, we perform TimeSformer + TIME using eight frame input videos varying loss components (in Eq. (14)) to demonstrate their effectiveness; (a) learning temporal order of frames $\mathcal{L}^{\text{order}}$, (b) debiasing spatial dynamics $\mathcal{L}^{\text{debias}}$, and (c) learning temporal flow direction of tokens $\mathcal{L}^{\text{flow}}$. To further validate the importance of temporal information in various experimental setups, we train video models with the same training details on the full SSv2 dataset, the Temporal and Static subsets (see Appendix A for their configurations). We report the test accuracies of both original and shuffled videos and their gap as a measurement of avoiding the spatial bias.

Table 3 summarizes the results: our loss components have an orthogonal contribution to the overall improvements in the model generalization (*i.e.*, Original), and TimeSformer + TIME (*i.e.*, using all components $\mathcal{L}^{\text{order}}$, $\mathcal{L}^{\text{debias}}$, and $\mathcal{L}^{\text{flow}}$) consistently improves the performances of Original and Gap on all benchmarks. For example, our method in the last row achieves the best score, which are 1.6%, 5.3%, and 2.8% higher Original accuracies on the SSv2, the Temporal and Static subsets, respectively. TimeSformer + TIME also largely surpasses TimeSformer in the metric of Gap by achieving 17.6, 40.2, and 17.6 improvements on the SSv2, the Temporal and Static subsets, respectively.

In addition, Table 3 shows that our loss components contribute more significantly when the dataset requires more temporal understanding (*e.g.*, the Temporal subset). For example, our method in the last row achieves more significant improvements in the metrics of Original and Gap on the Temporal subset compared to the scores on the SSv2

dataset and Static subset. Interestingly, except for the last row, the performances of Shuffled on the Static subset are often close to the Original ones. It arguably results from the static classes that allow video models to predict class labels without understanding temporal information, while our scheme of debiasing spurious correlation (*i.e.,* $\mathcal{L}^{\text{order}}$ and $\mathcal{L}^{\text{debias}}$) and enhancing temporal correlation (*i.e.,* $\mathcal{L}^{\text{flow}}$) can lead video models to alleviate spatial bias and learn effective temporal modeling. We believe that an elaborate design for utilizing such property might further improve the video understanding, and we leave it for future work.

5.3. Extension to image classification

In this section, we demonstrate an extension of our self-supervised idea of debiasing to image classification models, *e.g.,* Vision Transformer (Dosovitskiy et al., 2021; Touvron et al., 2020), to reduce the spurious correlation learned from the image backgrounds. For adaptation, our goal is to replace (a) learning temporal order of frames with spatial order of patches and (b) debiasing spatial dynamics with image backgrounds. For conciseness, we again use f_θ (5) to denote the whole process of ViTs parameterized by θ where $f_\theta^{[\texttt{CLS}]}(\mathbf{x})$ and $f_\theta^{(i)}(\mathbf{x})$ are the final representations of the $[\texttt{CLS}]$ token and the i-th patch image, respectively. Then, (a) learning spatial order of patches can be written as follow:

$$\mathcal{L}_I^{\text{order}}(\mathbf{x}) := \frac{1}{s}\sum_{i=1}^{s} \text{CE}(g_\theta^{\text{order}}(f_\theta^{(i)}(\mathbf{x})), \mathbf{y}^{(i)}), \tag{16}$$

where g_θ^{order} is an linear classification head. On the other hand, (b) debiasing image backgounds objective also can be written as follow:

$$\mathcal{L}_I^{\text{debias}}(\tilde{\mathbf{x}}) := \text{KL}(\mathcal{U}(\mathbf{y}) \| \texttt{Softmax}(g_\theta(f_\theta^{[\texttt{CLS}]}(\tilde{\mathbf{x}})))), \tag{17}$$

where g_θ is an linear classification head, and $\tilde{\mathbf{x}}$ is a sequence of randomly shuffled patches to reduce the effect of backgrounds. Then, adapted loss objectives $\mathcal{L}_I^{\text{TIME}}$ for image classification models can be written as follows:

$$\mathcal{L}_I^{\text{TIME}}(\mathbf{x}, \tilde{\mathbf{x}}) := \lambda_I^{\text{order}}\mathcal{L}_I^{\text{order}}(\mathbf{x}) + \lambda_I^{\text{debias}}\mathcal{L}_I^{\text{debias}}(\tilde{\mathbf{x}}), \tag{18}$$

where λ_I^{order} and $\lambda_I^{\text{debias}}$ are hyperparameters. In Table 4, we also simply use $\lambda_I^{\text{order}} = \lambda_I^{\text{debias}} = 1$.

Somewhat surprisingly, we found that adapted TIME loss (18) enhances the model generalization and robustness to background shifts when evaluated on Backgrounds Challenge[6] as shown in Table 4. Specifically, we train DeiT-Ti/16 (Touvron et al., 2020) with 300 training epochs on ImageNet-9 (Xiao et al., 2021) dataset, which contains 9 super-classes (370 classes in total) of ImageNet (Deng et al., 2009) for both background shifts experiments (Xiao et al., 2021). For example, Only-FG replaces backgrounds with the black, and Mixed-Same, Mixed-Rand, and Mixed-Next replace backgrounds with random backgrounds from the same, a random, and the next class, respectively, for disentangling foregrounds and backgrounds of the images.

Table 4 summarizes the results on the background shifts: our method consistently and significantly improves Baseline on overall benchmarks; *e.g.,* Baseline + $\mathcal{L}_I^{\text{TIME}}$ not only improves the top-1 accuracy of Baseline from 77.26% to 83.26% on the original dataset but also from 50.27% to 58.91% on the Only-FG benchmark in the background shifts. Also, Table 4 shows that $\mathcal{L}_I^{\text{order}}$ and $\mathcal{L}_I^{\text{debias}}$ have an orthogonal contribution to the overall improvements for alleviating background bias. For example, $\mathcal{L}_I^{\text{order}}$ improves Baseline from 50.27% to 54.15%, and $\mathcal{L}_I^{\text{debias}}$ improves it again from 54.15% to 58.91% on the Only-FG benchmark. Furthermore, comparing performance gains from each loss component and the combined one, it is worth noting that we confirm the remarkable synergy of spatial order prediction and background debiasing for robust image recognition in most benchmarks. This superiority of our method on background shifts shows its merits come from a widely applicable self-supervised idea for the vision domain.

6. Conclusion

We propose simple yet effective temporal self-supervision tasks (TIME) for improving video representations by capturing temporal dynamics of video data. Our key observation is that the existing Video Transformers do ineffective temporal modeling; *e.g.,* learning spurious correlation on spatial dynamics and vanishing temporal order information as the model layers deeper. In order to learn effective temporal modeling, we train the model to output low-confident predictions for temporally violated video data (*e.g.,* randomly shuffled video), and to predict both the correct temporal order of video frames and the temporal flow direction of video tokens. Through the extensive experiments, we highlight the importance of debiasing the spurious correlation of visual transformers with respect to the temporal or spatial dynamics. We believe that our work would provide insights toward the under-explored yet important problem.

Acknowledgements. We thank Seong Hyeon Park for providing helpful feedback and suggestions. This work was mainly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub; No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)). This work was partly experimented on the NAVER Smart Machine Learning (NSML) platform (Sung et al., 2017; Kim et al., 2018). This work was partly supported by KAIST-NAVER Hypercreative AI Center.

[6] https://github.com/MadryLab/backgrounds_challenge

References

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021.

Bertasius, G., Wang, H., and Torresani, L. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Bulat, A., Perez-Rua, J.-M., Sudhakaran, S., Martinez, B., and Tzimiropoulos, G. Space-time mixing attention for video transformer. In *NeurIPS*, 2021.

Carreira, J. and Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Cheng, B., Choudhuri, A., Misra, I., Kirillov, A., Girdhar, R., and Schwing, A. G. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Fan, Q., Panda, R., et al. An image classifier can suffice for video understanding. *arXiv preprint arXiv:2106.14104*, 2021.

Farnebäck, G. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pp. 363–370. Springer, 2003.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

Goyal, R., Kahou, S. E., Michalski, V., Materzyńska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. The "something something" video database for learning and evaluating visual common sense. In *European Conference on Computer Vision (ECCV)*, 2017.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations (ICLR)*, 2019.

Hu, K., Shao, J., Liu, Y., Raj, B., Savvides, M., and Shen, Z. Contrast and order representations for video self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7939–7949, 2021.

Huang, D.-A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L., and Niebles, J. C. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., and Zisserman, A. The kinetics human action video dataset, 2017.

Lee, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE international conference on computer vision*, pp. 667–676, 2017.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations (ICLR)*, 2018.

Li, K., Wang, Y., Gao, P., Song, G., Liu, Y., Li, H., and Qiao, Y. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022.

Li, Y. and Vasconcelos, N. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Lin, J., Gan, C., and Han, S. Tsm: Temporal shift module for efficient video understanding. In *European Conference on Computer Vision (ECCV)*, 2019.

Loshchilov, I. and Hutter, F. Fixing weight decay regularization in adam. 2018. URL https://openreview.net/forum?id=rk6qdGgCZ.

Misra, I., Zitnick, C. L., and Hebert, M. Shuffle and learn: unsupervised learning using temporal order verification. In *International Conference on Computer Vision (ICCV)*. Springer, 2016.

Moon, S. J., Mo, S., Lee, K., Lee, J., and Shin, J. Masker: Masked keyword regularization for reliable text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Neimark, D., Bar, O., Zohar, M., and Asselmann, D. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

Patrick, M., Campbell, D., Asano, Y. M., Metze, I. M. F., Feichtenhofer, C., Vedaldi, A., Henriques, J., et al. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *European Conference on Computer Vision (ECCV)*, 2017.

Sevilla-Lara, L., Zha, S., Yan, Z., Goswami, V., Feiszli, M., and Torresani, L. Only time can tell: Discovering temporal data for temporal modeling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2015.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Xiao, K., Engstrom, L., Ilyas, A., and Madry, A. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations (ICLR)*, 2021.

A. Configurations of Temporal and Static subsets in Something-Something-v2

By following Sevilla-Lara et al. (2021), we use the categorization of 18 temporal classes and 16 static classes of the Something-Sonething-v2 (SSv2) dataset (Goyal et al., 2017).

Temporal classes. "Turning something upside down", "Approaching something with your camera", "Moving something away from the camera", "Moving away from something with your camera", "Moving something towards the camera", "Lifting something with something on it", "Moving something away from something", "Moving something closer to something", "Uncovering something", "Pretending to turn something upside down", "Covering something with something", "Lifting up one end of something, then letting it drop down", "Lifting something up completely without letting it drop down", "Moving something and something closer to each other", "Moving something and something away from each other", "Lifting something up completely, then letting it drop down", "Stuffing something into something", "Moving something and something so they collide with each other".

Specifically, we use 6 temporal classes of "Lifting something up completely without letting it drop down", "Lifting something up completely, then letting it drop down", "Lifting up one end of something, then letting it drop down", "Moving something and something closer to each other", "Moving something and something away from each other", and "Moving something and something so they collide with each other" classes as the Temporal subset.

Static classes. "Folding something", "Turning the camera upwards while filming something", "Holding something next to something, Pouring something into something", "Pretending to throw something", "Squeezing something", "Holding something in front of something", "Touching (without moving) part of something", "Lifting up one end of something without letting it drop down", "Showing something next to something", "Poking something so that it falls over", "Wiping something off of something", "Scooping something up with something", "Letting something roll down a slanted surface", "Sprinkling something onto something", "Pushing something so it spins", "Twisting (wringing) something wet until water comes out". We use all the above 16 static classes as the Static subset.

B. Visualization of learned video models via GradCAM

In this section, we present details of qualitative results in Figure 4, and then provide more examples in Figure 5. To apply GradCAM (Selvaraju et al., 2017), which was originally developed based on CNNs, we use its adaptation for Vision Transformers provided by the original authors.[7] As the code is originally proposed for image data, we slightly modify it for video data by considering one more dimension for the temporal dimension (*i.e.*, frames). Here, one can again identify that our method successfully improves the existing Video Transformers with the proposed temporal self-supervised tasks; *e.g.*, focusing on the movements of objects (see Figure 5(a) and 5(b)) or the turning object (see Figure 5(c)).

C. Temporal modeling of TIME in training from scratch

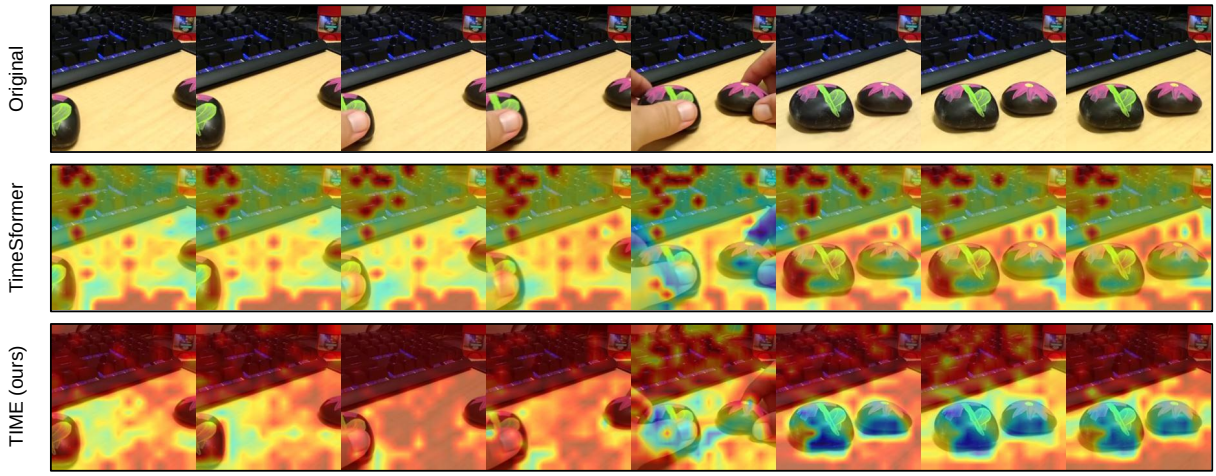
As many recent Video Transformers depend on pre-trained weights from a large-scale image dataset (*e.g.*, ImageNet (Deng et al., 2009)) for better performances, the pre-trained weights could be one of the reasons for spatial bias in the video models. However, even without the pre-trained weights, we empirically found that our method, TIME, still significantly improves TimeSformer from 39.4% to 64.4% in training from scratch on the Temporal subset using eight frame input videos. Again, we remark that static classes in the video datasets can also make video models biased toward learning spatial dynamics. Still, our method can lead them to alleviate spatial bias and learn temporal modeling better.

D. Alternative for keeping temporal information

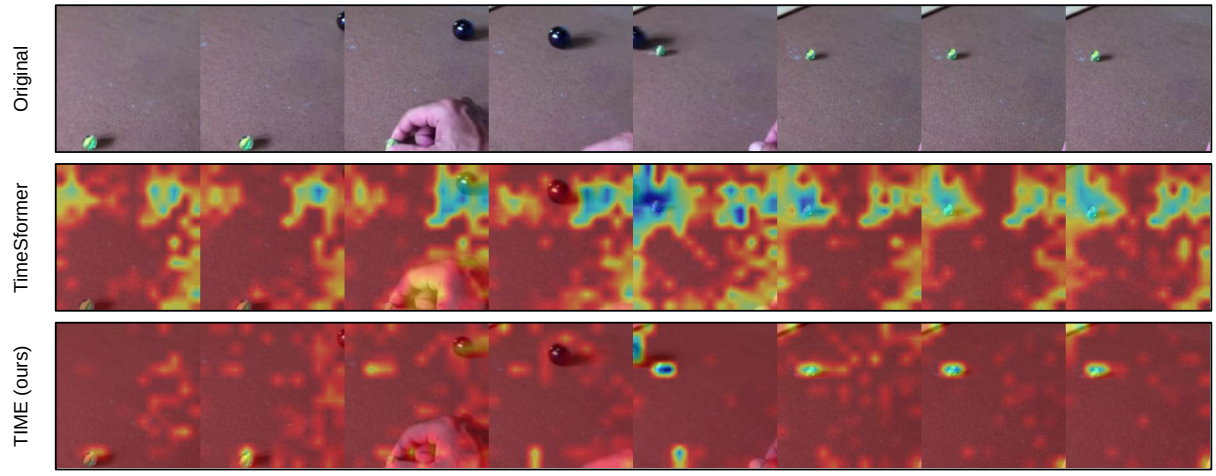
Here, we explore an alternative for keeping temporal information in Video Transformers. Specifically, we modify TimeSformer (Bertasius et al., 2021) to repeatedly re-add their temporal positional encodings before each block to maintain the temporal order of video frames and then train them on the Temporal subset. Interestingly, we observed that the re-adding one still loses temporal information at the final layer (27.8%), as shown in Figure 1(b). Furthermore, we found that learning temporal order $\mathcal{L}^{\text{order}}$ (7) could improve the re-adding one from 88.2% to 90.0% on the Temporal subset using 16 frame input videos.[8] These results show that architectural modifications for solely guiding temporal information may be limited; however, investigating architectural advances to keep temporal information would be a meaningful future direction.

[7] https://github.com/jacobgil/pytorch-grad-cam

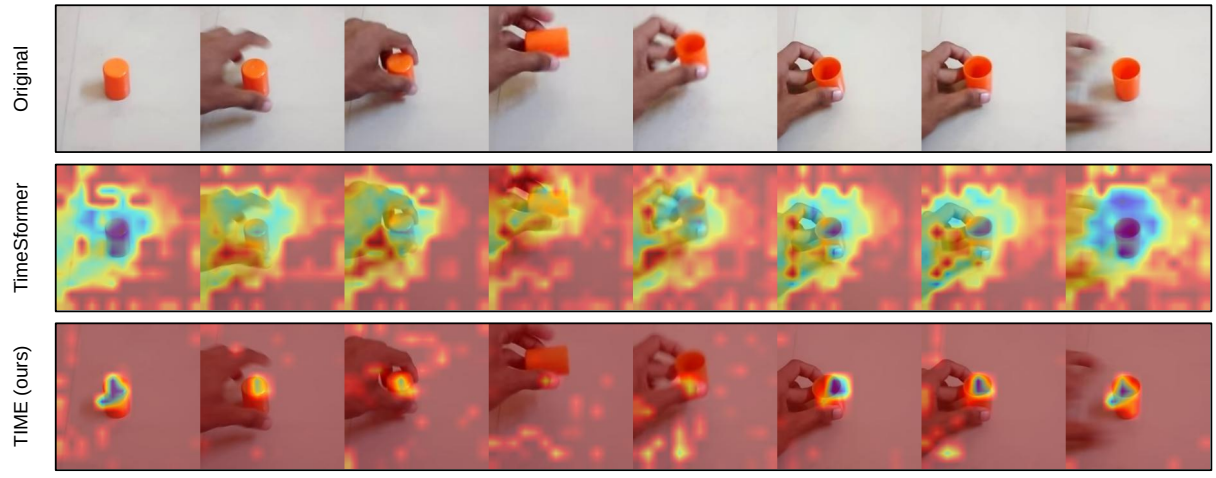
[8] We note the baseline (*i.e.*, the original TimeSformer) and the re-adding one + TIME achieve 87.2% and 90.5%, respectively.



(a) Examples from class *moving something and something closer to each other*.



(b) Examples from class *moving something and something so they collide with each other*.



(c) Examples from class *turning something upside down*.

Figure 5. More qualitative examples on SSv2 test dataset using GradCAM.