Video Swin Transformer

Ze Liu[*12]   Jia Ning[*13]   Yue Cao[1†]   Yixuan Wei[14]   Zheng Zhang[1]   Stephen Lin[1]   Han Hu[1†]

[1]Microsoft Research Asia   [2]University of Science and Technology of China
[3]Huazhong University of Science and Technology   [4]Tsinghua University

{t-liuze,v-jianing,yuecao,t-yixuanwei,zhez,stevelin,hanhu}@microsoft.com

Abstract

The vision community is witnessing a modeling shift from CNNs to Transformers, where pure Transformer architectures have attained top accuracy on the major video recognition benchmarks. These video models are all built on Transformer layers that globally connect patches across the spatial and temporal dimensions. In this paper, we instead advocate an inductive bias of locality in video Transformers, which leads to a better speed-accuracy trade-off compared to previous approaches which compute self-attention globally even with spatial-temporal factorization. The locality of the proposed video architecture is realized by adapting the Swin Transformer designed for the image domain, while continuing to leverage the power of pre-trained image models. Our approach achieves state-of-the-art accuracy on a broad range of video recognition benchmarks, including on action recognition (84.9 top-1 accuracy on Kinetics-400 and 85.9 top-1 accuracy on Kinetics-600 with ~20× less pre-training data and ~3× smaller model size) and temporal modeling (69.6 top-1 accuracy on Something-Something v2).

1. Introduction

Convolution-based backbone architectures have long dominated visual modeling in computer vision [14, 18, 22, 24, 32, 33]. However, a modeling shift is currently underway on backbone architectures for image classification, from Convolutional Neural Networks (CNNs) to Transformers [8, 28, 34]. This trend began with the introduction of Vision Transformer (ViT) [8, 34], which globally models spatial relationships on non-overlapping image patches with the standard Transformer encoder [38]. The great success of ViT on images has led to investigation of Transformer-based architectures for video-based recognition tasks [1, 3].

Previously for convolutional models, backbone architec-

tures for video were adapted from those for images simply by extending the modeling through the temporal axis. For example, 3D convolution [35] is a direct extension of 2D convolution for joint spatial and temporal modeling at the operator level. As joint spatiotemporal modeling is not economical or easy to optimize, factorization of the spatial and temporal domains was proposed to achieve a better speed-accuracy tradeoff [30, 41]. In the initial attempts at Transformer-based video recognition, a factorization approach is also employed, via a factorized encoder [1] or factorized self-attention [1, 3]. This has been shown to greatly reduce model size without a substantial drop in performance.

In this paper, we present a pure-transformer backbone architecture for video recognition that is found to surpass the factorized models in efficiency. It achieves this by taking advantage of the inherent spatiotemporal locality of videos, in which pixels that are closer to each other in spatiotemporal distance are more likely to be correlated. Because of this property, full spatiotemporal self-attention can be well-approximated by self-attention computed locally, at a significant saving in computation and model size.

We implement this approach through a spatiotemporal adaptation of Swin Transformer [28], which was recently introduced as a general-purpose vision backbone for image understanding. Swin Transformer incorporates inductive bias for spatial locality, as well as for hierarchy and translation invariance. Our model, called Video Swin Transformer, strictly follows the hierarchical structure of the original Swin Transformer, but extends the scope of local attention computation from only the spatial domain to the spatiotemporal domain. As the local attention is computed on non-overlapping windows, the shifted window mechanism of the original Swin Transformer is also reformulated to process spatiotemporal input.

As our architecture is adapted from Swin Transformer, it can readily be initialized with a strong model pre-trained on a large-scale image dataset. With a model pre-trained on ImageNet-21K, we interestingly find that the learning rate of the backbone architecture needs to be smaller (e.g. 0.1×)

---

* Equal Contribution. † Equal Advising. The work is done when Ze Liu, Jia Ning and Yixuan Wei are interns at Microsoft Research Asia.

than that of the head, which is randomly initialized. As a result, the backbone forgets the pre-trained parameters and data slowly while fitting the new video input, leading to better generalization. This observation suggests a direction for further study on how to better utilize pre-trained weights.

The proposed approach shows strong performance on the video recognition tasks of action recognition on Kinetics-400/Kinetics-600 and temporal modeling on Something-Something v2 (abbreviated as SSv2). For video action recognition, its 84.9% top-1 accuracy on Kinetics-400 and 85.9% top-1 accuracy on Kinetics-600 slightly surpasses the previous state-of-the-art results (ViViT [1]) both by +0.1 points, with a smaller model size (200.0M params for Swin-L vs. 647.5M params for ViViT-H) and a smaller pre-training dataset (ImageNet-21K vs. JFT-300M). For temporal modeling on SSv2, it obtains 69.6% top-1 accuracy, an improvement of +0.9 points over previous state-of-the-art (MViT [9]).

2. Related Works

CNN and variants In computer vision, convolutional networks have long been the standard for backbone architectures. For 3D modeling, C3D [35] is a pioneering work that devises a 11-layer deep network with 3D convolutions. The work on I3D [5] reveals that inflating the 2D convolutions in Inception V1 to 3D convolutions, with initialization by ImageNet pretrained weights, achieves good results on large-scale Kinetics datasets. In P3D [30], S3D [41] and R(2+1)D [37], it is found that disentangling spatial and temporal convolution leads to a speed-accuracy tradeoff better than the original 3D convolution. The potential of convolution based approaches is limited by the small receptive field of the convolution operator. With a self-attention mechanism, the receptive field can be broadened with fewer parameters and lower computation costs, which leads to better performance of vision Transformers on video recognition.

Self-attention/Transformers to complement CNNs NLNet [40] is the first work to adopt self-attention to model pixel-level long-range dependency for visual recognition tasks. GCNet [4] presents an observation that the accuracy improvement of NLNet can mainly be ascribed to its global context modeling, and thus it simplifies the NL block into a lightweight global context block which matches NLNet in performance but with fewer parameters and less computation. DNL [43] on the contrary attempts to alleviate this degeneration problem by a disentangled design that allows learning of different contexts for different pixels while preserving the shared global context. All these approaches provide a complementary component to CNNs for modeling long range dependency. In our work, we show that a pure-transformer based approach more fully

captures the power of self-attention, leading to superior performance.

Vision Transformers A shift in backbone architectures for computer vision, from CNNs to Transformers, began recently with Vision Transformer (ViT) [8, 34]. This seminal work has led to subsequent research that aims to improve its utility. DeiT [34] integrates several training strategies that allow ViT to also be effective using the smaller ImageNet-1K dataset. Swin Transformer [28] further introduces the inductive biases of locality, hierarchy and translation invariance, which enable it to serve as a general-purpose backbone for various image recognition tasks.

The great success of image Transformers has led to investigation of Transformer-based architectures for video-based recognition tasks [1, 3, 9, 25, 29]. VTN [29] proposes to add a temporal attention encoder on top of the pre-trained ViT, which yields good performance on video action recognition. TimeSformer [3] studies five different variants of space-time attention and suggests a factorized space-time attention for its strong speed-accuracy tradeoff. ViViT [1] examines four factorized designs of spatial and temporal attention for the pre-trained ViT model, and suggests an architecture similar to VTN that achieves state-of-the-art performance on the Kinetics dataset. MViT [9] is a multi-scale vision transformer for video recognition trained from scratch that reduces computation by pooling attention for spatiotemporal modeling, which leads to state-of-the-art results on SSv2. All these studies are based on global self-attention modules. In this paper, we first investigate spatiotemporal locality and then empirically show that the Video Swin Transformer with spatiotemporal locality bias surpasses the performance of all the other vision Transformers on various video recognition tasks.

3. Video Swin Transformer

3.1. Overall Architecture

The overall architecture of the proposed Video Swin Transformer is shown in Figure 2, which illustrates its tiny version (Swin-T). The input video is defined to be of size $T \times H \times W \times 3$, consisting of T frames which each contain $H \times W \times 3$ pixels. In Video Swin Transformer, we treat each 3D patch of size $2 \times 4 \times 4 \times 3$ as a token. Thus, the 3D patch partitioning layer obtains $\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4}$ 3D tokens, with each patch/token consisting of a 96-dimensional feature. A linear embedding layer is then applied to project the features of each token to an arbitrary dimension denoted by C.

Following the prior art [11, 12, 30, 41], we do not downsample along the temporal dimension. This allows us to strictly follow the hierarchical architecture of the original Swin Transformer [28], which consists of four stages and performs $2 \times$ spatial downsampling in the patch merging
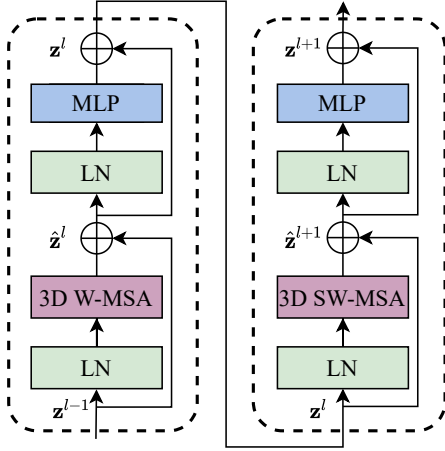
Figure 1. An illustration of two successive Video Swin Transformer blocks.

layer of each stage. The patch merging layer concatenates the features of each group of 2×2 spatially neighboring patches and applies a linear layer to project the concatenated features to half of their dimension. For example, the linear layer in the second stage projects $4C$-dimensional features for each token to $2C$ dimensions.

The major component of the architecture is the Video Swin Transformer block, which is built by replacing the multi-head self-attention (MSA) module in the standard Transformer layer with the 3D shifted window based multi-head self-attention module (presented in Section 3.2) and keeping the other components unchanged. Specifically, a video transformer block consists of a 3D shifted window based MSA module followed by a feed-forward network, specifically a 2-layer MLP, with GELU non-linearity in between. Layer Normalization (LN) is applied before each MSA module and FFN, and a residual connection is applied after each module. The computational formulas of the Video Swin Transformer block are given in Eqn. (1).

3.2. 3D Shifted Window based MSA Module

Compared to images, videos require a much larger number of input tokens to represent them, as videos additionally have a temporal dimension. A global self-attention module would thus be unsuitable for video tasks as this would lead to enormous computation and memory costs. Here, we follow Swin Transformer by introducing a locality inductive bias to the self-attention module, which is later shown to be effective for video recognition.

Multi-head self-attention on non-overlapping 3D windows Multi-head self-attention (MSA) mechanisms on each non-overlapping 2D window has been shown to be both effective and efficient for image recognition. Here,

we straightforwardly extend this design to process video input. Given a video composed of $T'\times H'\times W'$ 3D tokens and a 3D window size of $P\times M\times M$, the windows are arranged to evenly partition the video input in a non-overlapping manner. That is, the input tokens are partitioned into $\lceil\frac{T'}{P}\rceil\times\lceil\frac{H'}{M}\rceil\times\lceil\frac{W'}{M}\rceil$ non-overlapping 3D windows. For example, as shown in Figure 3, for an input size of $8\times8\times8$ tokens and a window size of $4\times4\times4$, the number of windows in layer l would be $2\times2\times2=8$. And the multi-head self-attention is performed within each 3D window.

3D Shifted Windows As the multi-head self-attention mechanism is applied within each non-overlapping 3D window, there lacks connections across different windows, which may limit the representation power of the architecture. Thus, we extend the shifted 2D window mechanism of Swin Transformer to 3D windows for the purpose of introducing cross-window connections while maintaining the efficient computation of non-overlapping window based self-attention.

Given that the number of input 3D tokens is $T'\times H'\times W'$ and the size of each 3D window is $P\times M\times M$, for two consecutive layers, the self-attention module in the first layer uses the regular window partition strategy such that we obtain $\lceil\frac{T'}{P}\rceil\times\lceil\frac{H'}{M}\rceil\times\lceil\frac{W'}{M}\rceil$ non-overlapping 3D windows. For the self-attention module in the second layer, the window partition configuration is shifted along the temporal, height and width axes by $(\frac{P}{2},\frac{M}{2},\frac{M}{2})$ tokens from that of the preceding layer's self-attention module.

We illustrate this with an example in Figure 3. The input size is $8\times8\times8$, and the window size is $4\times4\times4$. As layer l adopts regular window partitioning, the number of windows in layer l is $2\times2\times2=8$. For layer $l+1$, as the windows are shifted by $(\frac{P}{2},\frac{M}{2},\frac{M}{2})=(2, 2, 2)$ tokens, the number of windows becomes $3\times3\times3=27$. Though the number of windows is increased, the efficient batch computation in [28] for the shifted configuration can be followed, such that the final number of windows for computation is still 8.

With the shifted window partitioning approach, two consecutive Video Swin Transformer blocks are computed as

$$
\begin{aligned}
\hat{\mathbf{z}}^l &= \text{3DW-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1}, \\
\mathbf{z}^l &= \text{FFN}\left(\text{LN}\left(\hat{\mathbf{z}}^l\right)\right) + \hat{\mathbf{z}}^l, \\
\hat{\mathbf{z}}^{l+1} &= \text{3DSW-MSA}\left(\text{LN}\left(\mathbf{z}^l\right)\right) + \mathbf{z}^l, \\
\mathbf{z}^{l+1} &= \text{FFN}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1},
\end{aligned}
\tag{1}
$$

where $\hat{\mathbf{z}}^l$ and $\mathbf{z}^l$ denote the output features of the 3D(S)W-MSA module and the FFN module for block l, respectively; 3DW-MSA and 3DSW-MSA denote 3D window based multi-head self-attention using regular and shifted window partitioning configurations, respectively.
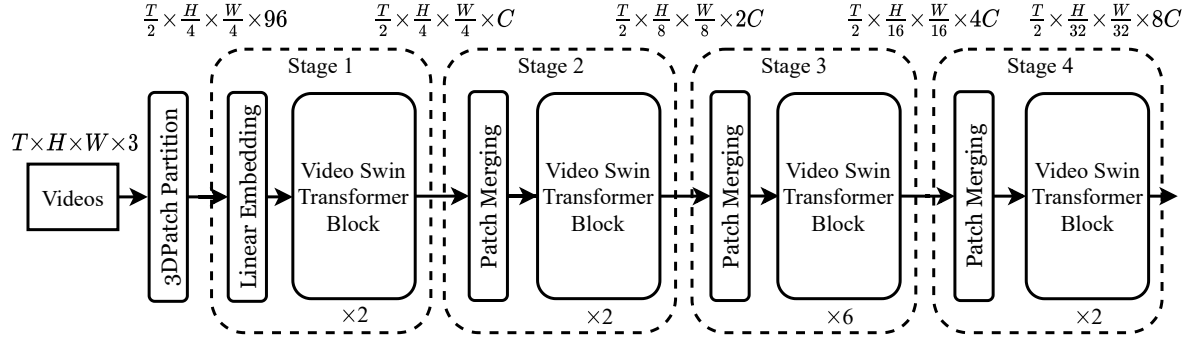
$T \times H \times W \times 3$ ... $\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4} \times 96$ ... $\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4} \times C$ ... $\frac{T}{2} \times \frac{H}{8} \times \frac{W}{8} \times 2C$ ... $\frac{T}{2} \times \frac{H}{16} \times \frac{W}{16} \times 4C$ ... $\frac{T}{2} \times \frac{H}{32} \times \frac{W}{32} \times 8C$

Figure 2. Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

3D tokens: T'×H'×W' = 8×8×8
Window size: P×M×M = 4×4×4

Layer l
window: 2×2×2=8

Layer l+1
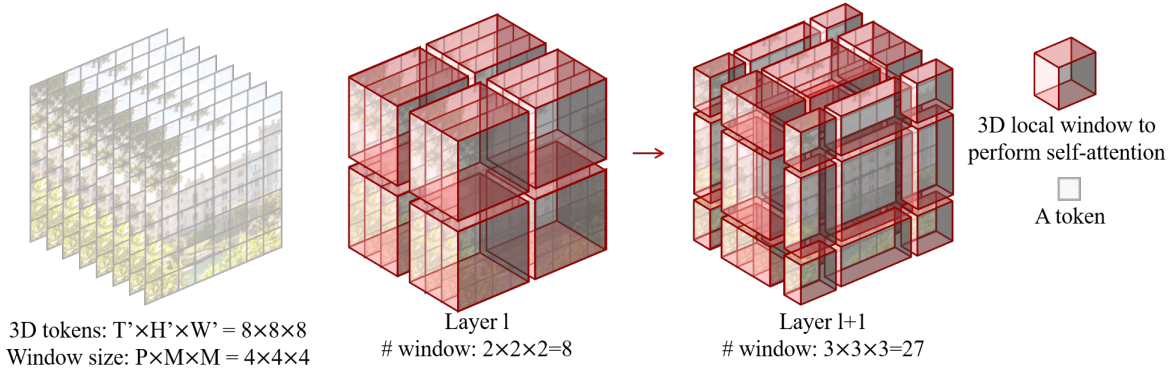# window: 3×3×3=27

3D local window to perform self-attention

A token

Figure 3. An illustrated example of 3D shifted windows. The input size $T' \times H' \times W'$ is 8×8×8, and the 3D window size $P \times M \times M$ is 4×4×4. As layer l adopts regular window partitioning, the number of windows in layer l is 2×2×2=8. For layer $l+1$, as the windows are shifted by $(\frac{P}{2}, \frac{M}{2}, \frac{M}{2})$=(2, 2, 2) tokens, the number of windows becomes 3×3×3=27. Though the number of windows is increased, the efficient batch computation in [28] for the shifted configuration can be followed, such that the final number of windows for computation is still 8.

Similar to image recognition [28], this 3D shifted window design introduces connections between neighboring non-overlapping 3D windows in the previous layer. This will later be shown to be effective for several video recognition tasks, such as action recognition on Kinetics 400/600 and temporal modeling on SSv2.

3D Relative Position Bias Numerous previous works [2, 16, 17, 31] have shown that it can be advantageous to include a relative position bias to each head in self-attention computation. Thus, we follow [28] by introducing 3D relative position bias $B \in \mathbb{R}^{P^2 \times M^2 \times M^2}$ for each head as

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \quad (2)$$

where $Q, K, V \in \mathbb{R}^{PM^2 \times d}$ are the *query*, *key* and *value* matrices; d is the dimension of *query* and *key* features, and PM^2 is the number of tokens in a 3D window. Since the relative position along each axis lies in the range of $[-P+1, P-1]$ (temporal) or $[-M+1, M-1]$ (height or width), we parameterize a smaller-sized bias matrix $\hat{B} \in$ $\mathbb{R}^{(2P-1) \times (2M-1) \times (2M-1)}$, and values in B are taken from $\hat{B}$.

3.3. Architecture Variants

Following [28], we introduce four different versions of Video Swin Transformer. The architecture hyperparameters of these model variants are:

- Swin-T: $C = 96$, layer numbers = $\{2, 2, 6, 2\}$
- Swin-S: $C = 96$, layer numbers = $\{2, 2, 18, 2\}$
- Swin-B: $C = 128$, layer numbers = $\{2, 2, 18, 2\}$
- Swin-L: $C = 192$, layer numbers = $\{2, 2, 18, 2\}$

where C denotes the channel number of the hidden layers in the first stage. These four versions are about $0.25\times$, $0.5\times$, $1\times$ and $2\times$ the base model size and computational complexity, respectively. The window size is set to $P = 8$ and $M = 7$ by default. The query dimension of each head is $d = 32$, and the expansion layer of each MLP is set to $\alpha = 4$.

3.4. Initialization from Pre-trained Model

As our architecture is adapted from Swin Transformer [28], our model can be initialized by its strong pre-trained model on a large-scale dataset. Compared to the original Swin Transformer, only two building blocks in Video Swin Transformers have different shapes, the linear embedding layer in the first stage and the relative position biases in the Video Swin Transformer block.

For our model, the input token is inflated to a temporal dimension of 2, thus the shape of the linear embedding layer becomes $96 \times C$ from $48 \times C$ in the original Swin. Here, we directly duplicate the weights in the pre-trained model twice and then multiply the whole matrix by 0.5 to keep the mean and variance of the output unchanged. The shape of the relative position bias matrix is $(2P - 1, 2M - 1, 2M - 1)$, compared to $(2M - 1, 2M - 1)$ in the original Swin. To make the relative position bias the same within each frame, we duplicate the matrix in the pre-trained model $2P - 1$ times to obtain a shape of $(2P - 1, 2M - 1, 2M - 1)$ for initialization.

4. Experiments

4.1. Setup

Datasets For human action recognition, we adopt two versions of the widely-used Kinetics [20] dataset, Kinetics-400 and Kinetics-600. Kinetics-400 (K400) consists of $\sim$240k training videos and 20k validation videos in 400 human action categories. Kinetics-600 (K600) is an extension of K400 that contains $\sim$370k training videos and 28.3k validation videos from 600 human action categories. For temporal modeling, we utilize the popular Something-Something V2 (SSv2) [13] dataset, which consists of 168.9K training videos and 24.7K validation videos over 174 classes. For all methods, we follow prior art by reporting top-1 and top-5 recognition accuracy.

Implementation Details For K400 and K600, we employ an AdamW [21] optimizer for 30 epochs using a cosine decay learning rate scheduler and 2.5 epochs of linear warm-up. A batch size of 64 is used. As the backbone is initialized from the pre-trained model but the head is randomly initialized, we find that multiplying the backbone learning rate by 0.1 improves performance (shown in Tab. 7). Specifically, the initial learning rates for the ImageNet pre-trained backbone and randomly initialized head are set to 3e-5 and 3e-4, respectively. Unless otherwise mentioned, for all model variants, we sample a clip of 32 frames from each full length video using a temporal stride of 2 and spatial size of 224×224, resulting in $16 \times 56 \times 56$ input 3D tokens. Following [28], an increasing degree of stochastic depth [19] and weight decay is employed for larger models, i.e. $0.1, 0.1, 0.3$ stochastic depth rate and $0.02, 0.02, 0.05$

weight decay for Swin-T, Swin-S, and Swin-B, respectively. For inference, we follow [1] by using 4×3 views, where a video is uniformly sampled in the temporal dimension as 4 clips, and for each clip, the shorter spatial side is scaled to 224 pixels and we take 3 crops of size 224×224 that cover the longer spatial axis. The final score is computed as the average score over all the views. For the experiments of training from scratch, we adopt the code base of [9] and follow its setting and hyperparameters, and train our Swin tiny with temporal dimension 8. We also report the accuracy with test views 5×1.

For SSv2, we employ an AdamW [21] optimizer for longer training of 60 epochs with 2.5 epochs of linear warm-up. The batch size, learning rate and weight decay are the same as that for Kinetics. We follow [9] by employing a stronger augmentation, including label smoothing, RandAugment [7], and random erasing [44]. We also employ stochastic depth [19] with ratio of 0.4. As also done in [9], we use the model pre-trained on Kinetics-400 as initialization and a window size in temporal dimension of 16 is used. For inference, the final score is computed as the average score of 1×3 views.

4.2. Comparison to state-of-the-art

Kinetics-400 Table 1 presents comparisons to the state-of-the-art backbones, including both convolution-based and Transformer-based on Kinetics-400. Compared to the state-of-the-art vision Transformers without large-scale pre-training, Swin-S with ImageNet-1K pre-training achieves slightly better performance than MViT-B (32×3) [9] which is trained from scratch with similar computation costs. Compared to the state-of-the-art ConvNet X3D-XXL [11], Swin-S also outperforms it with similar computation costs and fewer views for inference. For Swin-B, the ImageNet-21K pre-training brings a 2.1% gain over training on ImageNet-1K from scratch. With ImageNet-21K pre-training, our Swin-L (384↑) outperforms ViViT-L (320) by 3.6% on top-1 accuracy with similar computation costs. Pre-training on a significantly smaller dataset (ImageNet-21K) than ViViT-H (JFT-300M), our Swin-L (384↑) achieves the state-of-the-art performance of 84.9% on K400.

For a fair comparison with MViT [9], we train our Swin-T from scratch, with no change in hyperparameters. Our approach achieves competitive results with MViT [9], but we note that training from scratch needs 200 epochs with two repetitions of repeated augmentation [15], which takes $\sim$13.3$\times$ more training time than our fine-tuning setting.

Kinetics-600 Results on K600 are shown in Table 2. The observations on K600 is similar to those for K400. Compared with the state-of-the-art with ImageNet-21K pre-training, our Swin-L (384↑) outperforms ViViT-L (320)

Table 1. Comparison to state-of-the-art on Kinetics-400. "384↑" signifies that the model uses a larger spatial resolution of 384×384. "Views" indicates # temporal clip × # spatial crop. The magnitudes are Giga (10^9) and Mega (10^6) for FLOPs and Param respectively.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
R(2+1)D [37]	-	72.0	90.0	10 × 1	75	61.8
I3D [6]	ImageNet-1K	72.1	90.3	-	108	25.0
NL I3D-101 [40]	ImageNet-1K	77.7	93.3	10 × 3	359	61.8
ip-CSN-152 [36]	-	77.8	92.8	10 × 3	109	32.8
TPN-R101 [42]	ImageNet-1K	78.9	93.9	-	374	-
CorrNet-101 [39]	-	79.2	-	10 × 3	224	-
SlowFast R101+NL [12]	-	79.8	93.9	10 × 3	234	59.9
X3D-XXL [11]	-	80.4	94.6	10 × 3	144	20.3
MViT-S [9]	-	76.0	92.1	5 × 1	32.9	26.1
MViT-B, 32×3 [9]	-	80.2	94.4	5 × 1	170	36.6
MViT-B, 64×3 [9]	-	81.2	95.1	3 × 3	455	36.6
TimeSformer-L [3]	ImageNet-21K	80.7	94.7	1 × 3	2380	121.4
ViT-B-VTN [29]	ImageNet-21K	78.6	93.7	1 × 1	4218	11.04
ViViT-L/16x2 [1]	ImageNet-21K	80.6	94.7	4 × 3	1446	310.8
ViViT-L/16x2 320 [1]	ImageNet-21K	81.3	94.7	4 × 3	3992	310.8
ip-CSN-152 [36]	IG-65M	82.5	95.3	10 × 3	109	32.8
ViViT-L/16x2 [1]	JFT-300M	82.8	95.5	4 × 3	1446	310.8
ViViT-L/16x2 320 [1]	JFT-300M	83.5	95.5	4 × 3	3992	310.8
ViViT-H/16x2 [1]	JFT-300M	84.8	95.8	4 × 3	≈ 2800	647.5
Swin-T (T'=8,P=8)	-	76.6	92.5	5 × 1	44	28.2
Swin-T	ImageNet-1K	78.8	93.6	4 × 3	88	28.2
Swin-S	ImageNet-1K	80.6	94.5	4 × 3	166	49.8
Swin-B	ImageNet-1K	80.6	94.6	4 × 3	282	88.1
Swin-B	ImageNet-21K	82.7	95.5	4 × 3	282	88.1
Swin-L	ImageNet-21K	83.1	95.9	4 × 3	604	197.0
Swin-L (384↑)	ImageNet-21K	**84.9**	**96.6**	10 × 5	2107	200.0

Table 2. Comparison to state-of-the-art on Kinetics-600.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
SlowFast R101+NL [12]	-	81.8	95.1	10 × 3	234	59.9
X3D-XL [11]	-	81.9	95.5	10 × 3	48	11.0
MViT-B-24, 32×3 [9]	-	83.8	96.3	5 × 1	236	52.9
TimeSformer-HR [3]	ImageNet-21K	82.4	96	1 × 3	1703	121.4
ViViT-L/16x2 320 [1]	ImageNet-21K	83.0	95.7	4 × 3	3992	310.8
ViViT-H/16x2 [9]	JFT-300M	85.8	96.5	4 × 3	8316	647.5
Swin-B	ImageNet-21K	83.8	96.4	4 × 3	282	88.1
Swin-L (384↑)	ImageNet-21K	**85.9**	**97.1**	4 × 3	2107	200.0

by 2.9% on top-1 accuracy with similar computation costs. With pre-training on a significantly smaller dataset (ImageNet-21K) than ViViT-H (JFT-300M), our Swin-L (384↑) obtains state-of-the-art accuracy of 85.9% on K600.

Something-Something v2 Table 3 compares our approach with the state-of-the-art on SSv2. We follow MViT [9] by using the K400 pre-trained model as initialization. With pre-trained models on K400, Swin-B at-

tains 69.6% top-1 accuracy, surpassing the previous best approach MViT-B-24 with K600 pre-training by 0.9%. Our approach could be further improved via using larger model (e.g. Swin-L), larger resolution of input (e.g. 384^2) and better pre-trained model (e.g. K600). We leave these attempts as future work.

4.3. Ablation Study

Different designs for spatiotemporal attention We ablate three major designs for spatiotemporal attention: joint,

Table 3. Comparison to state-of-the-art on Something-Something v2.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
TimeSformer-HR [3]	ImageNet-21K	62.5	-	1×3	1703	121.4
SlowFast R101, 8×8 [12]	K400	63.1	87.6	1×3	106	53.3
TSM-RGB [27]	K400	63.3	88.2	2×3	62	42.9
MSNet [23]	ImageNet-21K	64.7	89.4	1×1	67	24.6
TEA [26]	ImageNet-21K	65.1	89.9	10×3	70	-
blVNet [10]	SSv2	65.2	90.3	1×1	129	40.2
ViViT-L/16x2 [1]	-	65.4	89.8	-	903	352.1
MViT-B, 64×3 [9]	K400	67.7	90.9	1×3	455	36.6
MViT-B-24, 32×3 [9]	K600	68.7	91.5	1×3	236	53.2
Swin-B	K400	**69.6**	**92.7**	1×3	321	88.8

split and factorized variants. The joint version jointly computes spatiotemporal attention in each 3D window-based MSA layer, which is our default setting. The split version adds two temporal transformer layers on top of the spatial-only Swin Transformer, which is shown to be effective in ViViT [1] and VTN [29]. The factorized version adds a temporal-only MSA layer after each spatial-only MSA layer in Swin Transformer, which is found to be effective in TimeSformer [3]. For the factorized version, to reduce the bad effects of adding randomly initialized layers into the backbone with pre-trained weights, we add a weighting parameter at the end of each temporal-only MSA layer which is initialized as zero.

Table 4. Ablation study on different designs for spatiotemporal attention with Swin-T on K400.

	Top-1	Top-5	FLOPs	Param
joint	78.8	93.6	88	28.2
split	76.4	92.1	83	42.0
factorized	78.5	93.5	95	36.5

Results are shown in Table 4. We can observe that the joint version achieves the best speed-accuracy tradeoff. This is mainly because locality in the spatial domain reduces computation for the joint version while maintaining effectiveness. In contrast, a joint version based on ViT/DeiT would be too computationally expensive. The split version does not work well in our scenarios. Though this version could naturally benefit from the pre-trained model, the temporal modeling of this version is not as efficient. The factorized version yields relatively high top-1 accuracy but requires many more parameters than the joint version. This is due the factorized version having a temporal-only attention layer after each spatial-only attention layer, while the joint version performs spatial and temporal attention in the same attention layer.

Temporal dimension of 3D tokens We perform an ablation study on the temporal dimension of 3D tokens in a

Table 5. Ablation study on the temporal dimension of 3D tokens and temporal window size with Swin-T on K400 and SSv2.

T'	Window size	K400 Top-1	K400 Top-5	SSv2 Top-1	SSv2 Top-5	FLOPs	Param
16	$16 \times 7 \times 7$	79.1	93.8	66.2	90.8	106	28.5
8	$8 \times 7 \times 7$	78.5	93.2	64.5	90.1	44	28.2
4	$4 \times 7 \times 7$	76.7	92.5	61.1	87.7	20	28.0
16	$16 \times 7 \times 7$	79.1	93.8	66.2	90.8	106	28.5
16	$8 \times 7 \times 7$	78.8	93.6	64.7	89.8	88	28.2
16	$4 \times 7 \times 7$	78.6	93.4	64.2	89.6	79	28.0

temporally global fashion, where the temporal dimension of 3D tokens is equal to the temporal window size. Results with Swin-T on K400 and SSv2 are shown in Table 5. In general, a larger temporal dimension leads to a higher top-1 accuracy but with a greater computation cost and slower inference. SSv2 benefits more from an increase in temporal dimension, which we attribute to the higher temporal dependency in SSv2.

Temporal Window Size Fixing the temporal dimension of 3D tokens to 16, we perform an ablation study over temporal window sizes of 4/8/16. Results with Swin-T on K400 are shown in Table 5. We observe that Swin-T with a temporal window size of 8 incurs only a small performance drop of 0.3 compared to a temporal window size of 16 (temporally global), but with a 17% relative decrease in computation (88 vs. 106). So we adopt this as the default setting for K400. However, SSv2 tends to need more temporal tokens and larger (global) temporal windows, so we adopt 16 temporal tokens with a global window as the default setting for the SSv2 dataset.

3D shifted windows Ablations of the *3D shifted windowing* approach on Swin-T are reported for K400 and SSv2 in Table 6. 3D shifted windows bring +0.7%/+3.5% in top-1 accuracy, and temporally shifted windows yield

Table 6. Ablation study on the 3D shifted window approach with Swin-T on K400 and SSv2.

	K400		SSv2	
	Top-1	Top-5	Top-1	Top-5
w. 3D shifting	78.8	93.6	64.7	89.8
w/o temporal shifting	78.5	93.5	62.1	88.5
w/o 3D shifting	78.1	93.3	61.2	87.7

+0.3%/+2.6% in K400/SSv2 respectively. The results indicate the effectiveness of the 3D shifted windowing scheme to build connections among non-overlapping windows.

Ratio of backbone/head learning rate An interesting finding on the ratio of backbone and head learning rates is shown in Table 7. With a model pre-trained on ImageNet-1K/ImageNet-21K, we observe that a lower learning rate of the backbone architecture (e.g. $0.1\times$) relative to that of the head, which is randomly initialized, brings gains in top-1 accuracy for K400. Also, using the model pre-trained on ImageNet-21K benefits more from this technique, due to the model pre-trained on ImageNet-21K being stronger. As a result, the backbone forgets the pre-trained parameters and data slowly while fitting the new video input, leading to better generalization. This observation suggests a direction for further study on how to better utilize pre-trained weights.

Table 7. Ablation study on the ratio of backbone lr and head lr with Swin-B on K400.

ratio	Pretrain	Top-1	Top-5
$0.1\times$	ImageNet-1K	80.6	94.6
$1.0\times$	ImageNet-1K	80.2	94.2
$0.1\times$	ImageNet-21K	82.6	95.7
$1.0\times$	ImageNet-21K	82.0	95.3

Initialization on linear embedding layer and 3D Relative Position Bias matrix In ViViT [1], center initialization of the linear embedding layer outperforms inflate initialization by a large margin. This motivates us to conduct an ablation study on these two initialization methods for Video Swin Transformer. As shown in Table 8, we surprisingly find that Swin-T with center initialization obtains the same performance as Swin-T with inflate initialization, of 78.8% top-1 accuracy using the ImageNet-1K pre-trained model[1] On K400. In this paper, we adopt the conventional inflate initialization on the linear embedding layer by default.

For the 3D relative position bias matrix, we also have two different initialization choices, duplicate or center initialization. Unlike the center initialization method for linear

Table 8. Ablation study on the two initialization methods of linear embedding layer with Swin-T on K400.

Initialization	Top 1	Top 5
Inflate	78.8	93.6
Center	78.8	93.7

Table 9. Ablation study on the two initialization methods of 3D relative position bias matrix with Swin-T on K400.

Initialization	Top 1	Top 5
Duplicate	78.8	93.6
Center	78.8	93.6

embedding layer, we initialize the 3D relative position bias matrix by masking the relative position bias across different frames with a small negative value (e.g. -4.6), so that each token only focuses inside the same frame from the very beginning. As shown in Table 9, we find that both initialization methods achieve the same top-1 accuracy of 78.8% with Swin-T on K400. We adopt duplicate initialization on the 3D Relative Position Bias matrix by default.

Shuffling frame inference We also evaluate our video model with random shuffling of frames at test time [9], and find that the effects on K400 and SSv2 differ considerably. As shown in Table 10, frame shuffling incurs a 3.3% top-1 accuracy drop on K400 and 49.9% on SSv2. The significant drop on SSv2 indicates that the SSv2 dataset requires much more effort in temporal modeling, and our video Swin Transformer can achieve a better trade-off on modeling spatial and temporal information for different datasets.

Table 10. Ablation study on shuffling frame inference with Swin-T on K400 and SSv2.

DataSet	shuffling	Top 1	Top 5
K400		78.8	93.6
	✓	75.5	91.5
SSv2		66.2	90.8
	✓	16.3	38.4

5. Conclusion

We present a pure-transformer architecture for video recognition that is based on spatiotemporal locality inductive bias. This model is adapted from the Swin Transformer for image recognition, and thus it could leverage the power of the strong pre-trained image models. The proposed approach achieves state-of-the-art performance on three widely-used benchmarks, Kinetics-400, Kinetics-600 and Something-Something v2. We hope that Video Swin Transformer will serve as a simple but solid baseline for future study.

---

[1]As this observation is inconsistent with that in [1], we will analyze the difference once the code of ViViT is released.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 1, 2, 5, 6, 7, 8

[2] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR, 2020. 4

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 1, 2, 6, 7

[4] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 2

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 5

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2

[9] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv:2104.11227*, 2021. 2, 5, 6, 7, 8

[10] Quanfu Fan, Chun-Fu Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. *arXiv preprint arXiv:1912.00869*, 2019. 7

[11] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 2, 5, 6

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 2, 6, 7

[13] 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. 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[15] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5

[16] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 4

[17] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3464–3473, October 2019. 4

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1

[19] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 5

[20] 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. 5

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[23] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *European Conference on Computer Vision*, pages 345–362. Springer, 2020. 7

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[25] Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. *arXiv preprint arXiv:2104.11746*, 2021. 2

[26] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 909–918, 2020. 7

[27] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 7

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1, 2, 3, 4, 5

[29] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021. 2, 6, 7

[30] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatiotemporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017. 1, 2

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 4

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, May 2015. 1

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1, 2

[35] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1, 2

[36] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 6

[37] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2, 6

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 1

[39] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 352–361, 2020. 6

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2, 6

[41] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 1, 2

[42] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020. 6

[43] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, 2020. 2

[44] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020. 5