Token Shift Transformer for Video Classification

Hao Zhang
City University of Hong Kong
Hong Kong SAR, China
zhanghaoinf@gmail.com

Yanbin Hao∗
University of Science and Technology of China
Hefei, China
haoyanbin@hotmail.com

Chong-Wah Ngo
Singapore Management University
Singapore
cwngo@smu.edu.sg

ABSTRACT
Transformer achieves remarkable successes in understanding 1- and 2-dimensional signals (e.g., NLP and Image Content Understanding). As a potential alternative to convolutional neural networks, it shares merits of strong interpretability, high discriminative power on hyper-scale data, and flexibility in processing varying length inputs. However, its encoders naturally contain computational intensive operations such as pair-wise self-attention, incurring heavy computational burden when being applied on the complex 3-dimensional video signals.

This paper presents Token Shift Module (i.e., TokShift), a novel, zero-parameter, zero-FLOPs operator, for modeling temporal relations within each transformer encoder. Specifically, the TokShift barely temporally shifts partial [Class] token features back-and-forth across adjacent frames. Then, we densely plug the module into each encoder of a plain 2D vision transformer for learning 3D video representation. It is worth noticing that our TokShift transformer is a pure convolutional-free video transformer pilot with computational efficiency for video understanding. Experiments on standard benchmarks verify its robustness, effectiveness, and efficiency. Particularly, with input clips of 8/12 frames, the TokShift transformer achieves SOTA precision: 79.83%/80.40% on the Kinetics-400, 66.56% on EGTEA-Gaze+, and 96.80% on UCF-101 datasets, comparable or better than existing SOTA convolutional counterparts. Our code is open-sourced in: https://github.com/VideoNetworks/TokShift-Transformer.

CCS CONCEPTS
• Computing methodologies → Activity recognition and understanding.

KEYWORDS
Video classification; Transformer; Shift; Self-attention

ACM Reference Format:
Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. 2021. Token Shift Transformer for Video Classification. In Proceedings of the 29th ACM International Conference on Multimedia (MM ’21), October 20–24, 2021, Virtual Event, China. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475272

1 INTRODUCTION
“If a picture is worth a thousand words, is a video worth a million?” [40]. This quote basically predicts that image and video can be potentially interpreted as linguistical sentences, except that videos contain richer information than images. The recent progress of extending linguistic-style, convolutional-free transformers [9] on visual content understanding successfully verifies the quote’s prior half, whereas the latter half for video remains an open hypothesis. This paper studies the efficient and effective way of applying similar transformers for video understanding and answer the quote’s prophecy for videos.

As a potential alternative for convolutional neural networks, the transformer achieves remarkable progress in both NLP [1, 2, 8, 17, 31] and vision tasks [3, 9, 28, 32, 35]. Particularly, language and vision transformers show several merits over their CNN counterparts, such as good interpretability (e.g., attention to highlight
core parts), unsaturated discriminatory power scalable with hyperscale data (e.g., million-scale language corpus[8,31] or billion-scale images[9]), and flexibility in processing varying length inputs [1, 17].

However, unlike signal understanding on 1 and 2-dimensional data (i.e., NLP & static vision), the application of transformers on 3-dimensional video signals is challenging. Specifically, each encoder of a transformer naturally contains heavy computations such as pair-wise self-attention; meanwhile, a video has a longer sequential representation \( \mathbf{z}_n \in \mathbb{R}^{T \times (N+1) \times D} \) than an image \( \mathbf{z}_f \in \mathbb{R}^{(N+1) \times D} \) due to an extra temporal axis. Consequently, directly applying general transformers on flattened spatio-temporal video sequences will introduce an exponential explosion of computations (e.g., \( T^2 \cdot \frac{(N+1)^2}{2} \) pair-wise distance calculations\(^1\)) in the training and inference phases.

To tackle this, we propose **Token Shift Module** (i.e., TokShift), a novel zero-parameter, zero-FLOPs operator, for modeling temporal relations within each video encoder. Specifically, the TokShift module barely temporally shifts partial \([\text{Class}]\) token features back-and-forth (Figure (1d)) across frames. Then, we densely plug the TokShift module into each encoder of a plain 2D vision transformer for learning 3D video representation.

Our TokShift is partially inspired by the success of spatial [6, 37, 41] and temporal [10, 20] shift operators for efficiency optimization on CNNs, but bears its own uniqueness. Specifically, on CNNs, spatial/temporal shift is uniformly applied across all spatial receptive fields on a feature-map. Follow this imitation, a copycat of the Temporal Shift Module for transformers is shown in Figure (1b), where the shift is imposed on all patches and \([\text{Class}]\) token features. However, we experimentally verify that, rather than temporally shifting features of all reception fields, just shift \([\text{Class}]\) token feature is sufficient. As in Figure (1b-1d), our TokShift introduces minimum modifications on original feature \( \mathbf{z}_n \), and gets the most improvement among all shift variants. More importantly, equipped with the TokShift, a video transformer only needs to handle \( T \cdot \frac{(N+1)^2}{2} \) pair-wise calculations. Finally, we verify that the shift operator is generalizable on transformers as it does on CNNs.

We try to intuitively explain the efficacy of the TokShift by drawing an analogy between textual description and visual feature vector of each frame. Specifically, the text describing a frame usually follow a similar linguistic form: “verb” + “noun” (e.g., “walking the dog” in Figure (4)), where “verb” is dynamically correlated with neighboring frames and “nouns” are static. Correspondingly, the TokShift exchanges partial visual features of a frame (in the form of a token vector) temporally back-and-forth for motion capturing while keeping the rest for static semantics modeling.

We conduct extensive experiments of the TokShift-xfmr on standard benchmarks, such as Kinetics-400 [4], EGTEA-Gaze+ [19] and UCF-101 [26] datasets. Experiments demonstrate that with the TokShift, video transformers could perform comparable or better than the best 3D-CNNs, reaching SOTA performance. Our contributions are summarized below.

- **Transformer for video classification.** Our TokShift-xfmr is an efficient pure convolutional-free transformer pilot applied for video classification. We repeat the successes of transformers on 3D videos as on 1/2D texts/images, promoting applications of transformers on different domains.

- **Efficient TokShift for transformer encoder.** We specifically design a novel zero-parameter, zero-FLOPs TokShift operator for the encoder. By merely manipulating global frame representation (i.e., \([\text{Class}]\) token vector) via shift operations, we can reduce computations of pair-wise attention in dealing with sequential flattening video representations within transformers.

- **State-of-the-art performances.** We test the TokShift-xfmr on several standard benchmarks. Experiments show that the TokShift-xfmr achieves SOTA top-1 accuracy 79.83%/80.40% (8/12f) on Kinetics-400, 66.56% (8f) on EGTEA-Gaze+, and 96.80% (8f) on UCF-101 datasets, comparable or better than the best 3D-CNNs available (TSM, SlowFast-R101-NL, X3D-XXL).

2 RELATED WORKS

The TokShift-xfmr is closely relevant to research areas, including 3D-CNNs for video classification, shift strategies for convolutional optimization, and 1/2D transformers for NLP/vision understanding; we separately relate the related literature for each area as below.

3D-CNNs for video classifications. Convolutional neural networks stepped into a prosperous age in the past few years [5, 16, 18, 24, 25, 27, 36, 38, 42–45]. Consequently, the 3D-CNN [4, 11, 12, 14, 15, 21–23, 30, 39] has become a de-facto standard for video content understanding. Specifically, C3D [30] firstly inflates the convolutional kernel from 2D to 3D, to facilitate temporal modeling. For motion capturing, the two-stream network [23] fuses two CNNs trained on optical-flow with RGB modalities. Furthermore, 3D [4] deepens 3D-CNNs by inflating 3D kernels on inception-net, accompanied with a large-scale Kinetics dataset. To reduce computational overhead, Pseudo-3D [21] decomposes a 3D Spatio-temporal convolution into Spatial-2D + Temporal-1D form. This decomposition greatly balances computations, parameters, and performances, hence gain its popularity in later 3D-CNNs. S3D-G [39] scales channel elements with attention yielded by global feature. Non-local networks [33] introduces self-attention on top of CNNs; furthermore, CBA-QSA CNN [15] extends self-attention with compact bi-linear mapping for fine-grained action classification. SlowFast [12] and X3D [11] are currently best two 3D-CNNs. The former designs dual-paths CNNs, where each path receives input clip of slow/fast sampling rate; the latter presents an efficient strategy to search for optimal hyperparameters (e.g., spatial/temporal resolutions, channels, depth, etc.) on a template network (i.e., X2D).

Spatial/temporal shift for efficient CNNs. Mobile computing demands more efficiency than cloud computing. Thereby, the Shift [6, 10, 20, 37, 41], a zero-parameter, zero-FLOPs operator for local feature aggregation, is proposed to reduce the complexities of CNNs. Specifically, Shift + 1D/2D convolution can lossily approximate a 2D/3D convolution. For example, in [37], Wu replaces spatial convolution of \( 3 \times 3 \) kernel with Shift + \( 1 \times 1 \) kernel for vision

\(^1T, N, D\) denote time-stamps, spatial-patches, \([\text{Class}]\) token and feature-dim
An overview of the Token Shift transformer (i.e., TokShift-xfmr) or deeper transformer). For object detection and segmentation tasks, including image classification [9, 28], object detection [3] and segmentations [35]. Specifically, Dosovitskiy [9] proposes Shift + Add, eliminating all multiplications. For videos, Lin [20] constructs TSM with Temporal Shift + spatial convolutions; RubiksNet [10] further removes spatial convolutions and introduces a learnable spatio-temporal shift operator.

Language & vision transformer. Firstly proposed in [31], the transformer relies on residual attention & feedforward (FFN) in feature learning and outperforms CNNs to become the de-facto standard in NLP tasks [2, 8]. To tackle inputs of different/long length, Reformer [17] and Longformer [1] introduce optimizations on attention calculations. Recently, researchers repeat this success in vision tasks, including image classification [9, 28], object detection [3], and segmentations [35]. Specifically, Dosovitskiy [9] proposes the first pure convolutional-free vision transformer (i.e., ViT) for image classification. To tackle the generalization problem caused by insufficient data, they equip the pre-training phase with a hyper-scale internal dataset (JFT-300M). In [28], DeiT bypasses the necessity of pre-training transformer on hyper-scale data through distillation. They introduce the token distillation for transformers to teach “student” transformer from various “teachers” (i.e., convnet or deeper transformer). For object detection and segmentations, as the transformer contains positional encodings under the encoder/decoder mechanism, it can be naturally extended to regress coordinates [3] and masks [35]. Overall, transformers are strong alternatives to CNNs and share merits like good interpretability, unsaturated discriminative power and flexibility on input length.

3 METHOD

An overview of the Token Shift transformer (i.e., TokShift-xfmr) is presented in Figure (2). Our TokShift-xfmr follows a general pipeline of transformers, except for working on video inputs and containing extra temporal processing modules (i.e., the TokShift).

The TokShift-xfmr (Figure (2)) contains L replicated, identical encoders. An encoder consists of the TokShift module, Layer-Norm (LN), Multi-Head Self-attention (MSA), and Feed-Forward Network (FFN). The workflow for connecting them is shown by Figure (2) or Equations (2-4 & 8-10). We will describe each module separately as below.

The TokShift module (Equation (2) & (8)) serves to in-placed manipulate global token $\mathbf{z}/\hat{\mathbf{z}}_{l-1}$ along temporal axis, introducing temporal interactions before feeding video embedding $\mathbf{z}/\hat{\mathbf{z}}_{l-1}$ through MSA/FFN. Hereby, the token $\mathbf{c}/\hat{\mathbf{c}}_{l-1}$ is the special [Class] word of $\mathbf{z}/\hat{\mathbf{z}}_{l-1}$. We will present details of the TokShift and its variants in section 3.1.

$$
\mathbf{z}_0 = \left[ \mathbf{c}_0; \mathbf{x}_0^0 \mathbf{E}; \ldots; \mathbf{x}_0^N \mathbf{E} \right] + \mathbf{E}_{pos} \tag{1}
$$

$$
\mathbf{x}_i^t \in \mathbb{R}^{T \times D}, \mathbf{E} \in \mathbb{R}^{d \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}
$$

$$
\mathbf{z}_0 \in \mathbb{R}^{T \times (N+1) \times D}
$$

The MSA (same for LN & FFN) acts in the same way as in 2D ViT [9]. Specifically, its functions independently process each frame. Pick $\mathbf{z}^{(i)}_{l-1} \in \mathbb{R}^{(N+1) \times D}$ at time-stamp $i$ of $\mathbf{z}_{l-1}$ for example, functions of MSA are performed within each frame (Equations (5)-(7)).
We propose and compare several shift variants, including the TokShift module, customized for the video transformer paradigm. The TokShift can be plugged in multiple positions of a vision encoder: "prior residual" (TokShift), "prior layer-norm" (TokShift-B), "prior MSA/FFN" (TokShift-C), and "post MSA/FFN" (TokShift-C). Notably, a frontier position will affect the subsequent modules inside residual blocks more.

Given a token tensor $c_l \in \mathbb{R}^{T \times D}$ representing the global sequence of a clip, the TokShift works as follows: $c_l$ is firstly split into 3 groups along channel dimension (Equation (12)).

$$c_l = [s_a, s_b, s_c]$$

Then, channels of splits $s_a/s_b/s_c$ are temporally shifted with prior/post time-stamps. For the split $s_a$, its content remains unchanged (Equations (13)-(15)). The percentages of shifted channels are determined by $\frac{a}{D}$ & $\frac{b}{D}$.

Non Shift module can be obtained by relacing "TokShift" with "Identity" function in Equation (2) & (8). This modification removes all temporal interactions and processes each frame independently. We name this simple 2D vision transformer for videos as ViT (video) for simplicity (i.e., Figure (1a) & (3a)).

**Temporal Shift** extension for the encoder is obtained by imitating spatial/temporal shifts [20, 37, 41] in CNN. Recall that a CNN feature-map of image/video is a tensor collecting features from all the receptive fields (spatial or spatio-temporal locations). A shift operator then exchanges partial channels from neighboring receptive fields for each location on the feature-map. Similarly, in Figure (1b), temporal shift for encoder exchanges partial channels from prior/post time-stamps for all "N patches" and "1 [Class] token".

**Patch Shift** is a variant of the temporal shift. Since the [Class] token is a global representation, we exempt it from shift operation and only process patch embeddings (Figure (1c)).
We compare the four shift variants on a standard benchmark and observe that the TokShift performs the best among them, showing the potential of capitalizing [Class] tokens for temporal modeling in video transformers. We claim that TokShift introduces the minimum modifications on $z_t$ but grants the maximum benefits.

3.2 Integration into ViT Encoder
There are multiple candidate positions in an encoder to implant the TokShift module. The position is essential since it determines the degree of motions involved in representation learning. Specifically, from TokShift to TokShift-A/B/C (Figure (3b)-(3e)), the influence of motions decreases as fewer modules work upon shifted features.

A general encoder contains two residual blocks: one for self-attention, another for feature embedding (Figure (3a)). In each block, the TokShift can be placed at “prior residual”, “prior layer-norm”, “prior MSA/FFN” and “post MSA/FFN”. We experimentally verify that placing TokShift at a front position (i.e., “prior residual”) brings the best effect. This finding is different from TSM on CNN, where the optimal position is “post residual”.

4 EXPERIMENTS
We conduct extensive experiments on three standard benchmarks for video classification and adopt Top-1/5 accuracy (%) as evaluation metrics. We also report the model parameters and GFLOPs to quantify computations.

4.1 Datasets
We adopt one large-scale (Kinetics-400) and two small-scale (EGTEA-Gaze+ & UCF-101) datasets as evaluation benchmarks. Below presents their brief descriptions.

**Kinetics-400** [4] serves as a standard large-scale benchmark for video classification. Its clips are truncated into 10 seconds duration with humans’ annotations. The training/validation set separately contains ~246k/20k clips covering 400 video categories.

**EGTEA Gaze+** [19] is a First-Person-View video dataset covering 106 fine-grained daily action categories, such as “human-object” interactions, sports and etc. We also select split-1 for performance reporting. This split contains 9,537/3,783 training/validation clips with average duration of 3.1 seconds.

**UCF-101** [26] contains 101 action categories, such as “human-object” interactions, sports and etc. We also select split-1 for performance reporting. This split contains 8,299/2,022 training/validation clips, with an average clip length of 3.1 seconds.

4.2 Implementations
We implement a paradigm of shift-transformers, including the ViT (video), TemporalShift/PatchShift/TokShift-xfmr, on top of 2D ViT [9] with PyTorch. We also support backbones of various types or depths.

All experiments share the same settings unless particularly specified. For ablations with higher resolutions, more frames, or heavier backbones, we follow the “linear scaling rule” [13] to adjust lr according to batch-size.

**Training.** Each clip, randomly cropped into 224×224 (256/384 for high resolutions), contains 8/16 frames with a temporal step of 32. We apply training augmentations before cropping to reduce overfitting, including random resize (with the short side in [244, 330] and maintain aspect ratio), random brightness, saturation, gamma, hue, and horizontal-flip. We train the model for 18 epochs with batch-size 21 per GPU. Base lr is set to 0.1, is decayed by 0.1 at epoch [10, 15] during training. The **Base-16 ViT** [9] (contains 12 encoders) is adopted as the backbone and the shifted proportion is set to $\frac{3}{7}, \frac{6}{7} = \frac{3}{7}$. All experiments are run on 2/8× V100 GPUs.

**Inference** adopts the same testing strategies as [12]. Specifically, we uniformly sample 10 clips from a testing video. Each clip are resized to short-side=224 (256/384), then cropped into three 224×224 sub-clips (from “left”, “center”, “right” parts). A video prediction is obtained by averaging scores of 30 sub-clips.

4.3 Ablation Study
Our ablation studies the impacts of the shift-types, integrations, hyperparameters of video transformers on the Kinetics-400 dataset.

**Non Shift vs TokShift.** We compare the Non Shift and TokShift under various sampling steps. Specifically, we fix clip size (7) to 8 frames, but change sampling steps (S) to cover variable temporal duration.

| Model       | T × S | 8 × 8 | 8 × 16 | 8 × 32 | 8 × 64 | Acc1 (%) | Acc5 (%) |
|-------------|-------|-------|--------|--------|--------|----------|----------|
| ViT (video) [9] | None | 76.17 | 76.32  | 76.62  | 76.81  | 76.90 (0.73%) | 76.81 (0.49%) |
| TokShift-A | Token | 134.7 × 30 | 76.85 | 76.90 (0.49%) | 76.90 (0.73%) |
| TokShift-B | Token | 134.7 × 30 | 77.21 | 92.81 |
| TokShift-C | Token | 134.7 × 30 | 77.00 | 92.92 |
| TokShift | Token | 134.7 × 30 | 77.28 (1.26%) | 76.73 (0.87%) |

**Table 1**: NonShift vs TokShift under different sampling strategies (‘T/S’ refers to frames/sampling-step; Accuracy-1).

As in Table 1, the TokShift-xfmr outperforms ViT (video) under all settings. We observe that the performance of TokShift varies with different temporal steps. A smaller step makes TokShift not effective in learning temporal evolution. On the other hand, a large step of 64 frames results in a highly discontinuous frame sequence, affecting learning effectiveness. A temporal step of 32 frames appears to be a suitable choice.

**Integration.** We study the impacts of implanting the TokShift at different positions of an encoder. As mentioned in section 3.2, a front position will affect the later modules inside residual blocks more.

| Model       | Words Shifted | GFLOPs × ViT wins | Acc1 (%) | Acc5 (%) |
|-------------|---------------|--------------------|----------|----------|
| ViT (video) [9] | None | 134.7 × 30 | 76.81 | 92.92 |
| TokShift-A | Token | 134.7 × 30 | 76.85 | 92.81 |
| TokShift-B | Token | 134.7 × 30 | 77.21 | 92.92 |
| TokShift-C | Token | 134.7 × 30 | 77.00 | 92.92 |
| TokShift | Token | 134.7 × 30 | 77.28 (1.26%) | 76.73 (0.87%) |

**Table 2**: Comparisons of the TokShift residing in different positions (GFLOPs, Accuracy-1/5).

Table 2 lists comparisons of the TokShift residing in different positions. We observe that: (1) All models share the same GFLOPs, which is consistent with our zero-GFLOPs claims on the TokShift; (2) Regardless of positions, the TokShift stably improves over the ViT (video) baseline; (3) Placing the TokShift at “prior residual”, a front position that affects all later modules, presents the best top-1 accuracy.

**Words Shifted.** We assess contributions of the token-specific shift by comparing all shift variants in section 3.1.
In Table 3, the TokShift exhibits the best top-1 accuracy. Additionally, consideration of patches in shift (TemporalShift & PatchShift) reduces overall performance. The reason lies in that patches of the same spatial grid across times incurs visual misalignment when rigidly dividing a moving object. We verify this hypothesis by measuring the mean cosine similarity between the patch and its temporal neighbor features on kinetics-400 val set with Temporal/Patch Shift transformer and observe distances scores of 0.577/0.570. As for [Class] token part, since it reflects global frame-level contents without alignment concerns, the distance between neighboring temporal tokens is 0.95. Consequently, for the TokShift or CNN-Shift, the drawback is bypassed by global alignment or sliding window.

Proportion of shifted channels. This hyperparameter controls the ratio of dynamic/static information in a video feature. We evaluate \( \frac{\alpha}{\alpha + \beta} \) in range \([1/4, 1/8, 1/16]\) and present results in Table 4. We experimentally find that 1/4 is an optimal value, indicating that half channels are shifted (1/4 back + 1/4 forth) while the rest half remain unchanged.

### Table 3: Comparisons of shifting different visual “Words” (GFLOPs, Accuracy-1/5).

| Shift Type          | Words Shifted | GFLOPs x Views | Acc1 (%) | Acc5 (%) |
|---------------------|---------------|----------------|----------|----------|
| ViT (video) [9]     | None          | 134.7 x 30     | 76.02    | 92.32    |
| TemporalShift       | Tokens + Patches | 134.7 x 30   | 72.55    | 91.24    |
| PatchShift          | Patches       | 134.7 x 30     | 73.08    | 91.17    |
| TokShift            | Token         | 134.7 x 30     | 77.28    | 92.91    |

Table 4: Comparisons of the TokShift regarding to proportions of shifted channels (Accuracy-1/5).

| Model       | Channels Shifted   | Acc1 (%) | Acc5 (%) |
|-------------|--------------------|----------|----------|
| TokShift    | 1/4 + 1/4         | 77.28    | 92.91    |
| TokShift    | 1/8 + 1/8         | 77.18    | 92.95    |
| TokShift    | 1/16 + 1/16       | 76.93    | 92.82    |

Word count is an essential factor in transformer learning since more words indicate more details. As for videos, the number of visual words is positively correlated with two factors: spatial resolutions and temporal frames. We study the impacts of word count on the TokShift-xfmr. Table 5 lists the detailed performances under different word counts. Here, MR/HR represents middle/high resolutions.

### Table 5: Comparisons of the TokShift according to visual word counts (Accuracy-1).

| Model       | Res (H x W) | Words/Frame | #Frames | # Words | T \cdot (N + 1) | Acc1 (%) |
|-------------|-------------|-------------|---------|---------|----------------|----------|
| TokShift    | 224 x 224   | 14^2 + 1    | 8       | 1,102   | 76.72          |
| TokShift    | 224 x 224   | 14^2 + 1    | 8       | 1,576   | 77.28          |
| TokShift    | 224 x 224   | 14^2 + 1    | 10      | 19,70   | 77.56          |
| TokShift    | 224 x 224   | 14^2 + 1    | 16      | 3,152   | 78.18          |
| TokShift (MR)| 256 x 256   | 16^2 + 1    | 8       | 2,096   | 77.68          |
| TokShift (HR)| 384 x 384   | 24^2 + 1    | 8       | 4,616   | 78.14          |

### Table 6: Comparisons of the TokShift on different transformer backbones (Accuracy-1).

- **On various backbones.** We test the TokShift on transformers of various depths and types. Specifically, we further evaluate the TokShift-xfmr on Large-16 (24 encoders) and Hybrid-16 (12 encoders) ViTs. Compared to the Base-16, the Large-16 doubles the number of encoders. Hybrid-16 shares the same number of encoders but replaces the linear projection layer (\( \mathbf{E} \) in Equation (1)) with three residual blocks of ResNet50. Notably, for the TokShift-Large (HR), we set \( T = 12 \) due to the memory limit.

- **Comparison with the State-of-the-Art**
  - We compare our TokShift-xfmr with current SOTAs on Kinetics-400 datasets in Table 7. Since there is no prior work that purely utilizes transformer, we mainly select 3D-CNNs, such as i3D, Non-Local, TSM, SlowFast, X3D as SOTAs.
  - Compared to 3D-CNNs, transformers exhibit prominent performances. Particularly, the spatial-only transformer, i.e., ViT (video), achieves comparable or better performance (76.02) than strong 3D-CNNs, such as i3D (71.1), 3D-G (74.7) and TSM (76.3). Moreover, a comparison of our most slimmed TokShift-xfmr (77.28) with TSM (76.3) indicates that shift gains extra benefits on transformers than CNNs. Thirdly, the TokShift-xfmr (MR) achieves comparable performance (77.68) with Non-Local R101 networks (77.7) under the same spatial resolutions (255) but using fewer frames (8 vs 128). This is reasonable since both contain the attentions. Finally, we compare TokShift-xfmr with the current best 3D-CNNs: SlowFast and X3D. The two 3D-CNNs are particularly optimized by a dual-paths or efficient network parameters searching mechanism. Since both SlowFast & X3D introduce large structural changes over corresponding 2D nets, their authors prefer to train on videos with sufficient epochs (256), rather than initialize from ImageNet pretrained weights. We show that our TokShift-Large-xfmr (80.40) performs better than SlowFast (79.80) and is comparable with X3D-XXL (80.4), where models are under their best settings. Notably, we only use 12 frames, whereas SlowFast/X3D-XXL uses 32/16+ frames.
  - In all, a pure transformer can perform comparable or better than 3D-CNNs for video classification. We additionally verify that temporal modeling can be imposed just on global [Class] tokens rather than whole video embeddings. Besides, the common computational limitation of the transformer (appeared in NLP/vision task) also
exists for videos. Though the TokShift introduces zero parameters & FLOPs, the 2D transformer alone consumes more computations than 3D-CNNs. The reasons lie in: (1). Pair-wise distance calculation in attention is computationally expensive; (2). 3D-CNNs have been computationally optimized (TSM, P3D, etc.). Nevertheless, computation optimization for the transformer is already very hot for NLP [1, 7] and image understanding [29]. As expected, computation optimization for long-length videos also requires future exploration.

4.5 Attention Visualization

An advantage of transformer over CNN is interpretability. Specifically, the self-attention emphasizes vital visual clues with high attentions. Hence, we can visualize the attention map to infer what the transformer values.

We adopt a public script\(^1\) to visualize the attention map from the last encoder. Maps from multiple heads are averaged to a heat map. We sample six clips, including daily and sports activities, from the Kinetics-400 and show their maps in Figure (4). We keep half (4 of 8) frames for space-saving. Notably, most action categories follow a form of “verb + noun” (e.g., “flying kites”), meeting our intuitive explanations. Our TokShift-xfmr localizes video features of 60 categories from EGTEA Gaze+ with large improvements compared to strong 3D-CNNs (I3D, P3D & TSM), and especially, our TokShift-xfmr alleviates intensive pair-wise distance calculations of the Spatio-temporal attention, maintaining the same modality (Video) with EGTEA Gaze+, while ImageNet-21k is still efficient for fine-tuning from Kinetics-400 weights, whereas harmful for ImageNet-21k weights. Because the Kinetics-400 shares the same modality (Video) with EGTEA Gaze+, while ImageNet-21k is different (Image). We verify the efficacy of two-stream TokShift-xfmr by adding the optical-flow (64.82→65.08). Finally, we compare the TokShift with strong 3D-CNNs (I3D, P3D & TSM), and verify its efficacy on small-scale datasets.

4.6 Fine-tune on small-scale datasets

We study the impacts of pre-trained weights on small-scale datasets. Specifically, we fine-tune models on EGTEA Gaze+ and UCF-101 datasets with pre-weights from “None”, “ImageNet-21k” and “Kinetics-400”. The sampling step is reduced to 8, as both datasets have a shorter duration (3/5s) than Kinetics-400 (10s). Training schemes are also optimized (EGTEA: 18 epochs, decay at [10,15]) and UCF: 25 epochs, decay at [10, 20]).

Table 7: Comparison to state-of-the-arts on Kinetics-400 Val.

Table 8: Impacts of various pretrained weights on EGTEA-Gaze++ Split-1 dataset (** means freeze layer-norm).

Table (8) & (9) list their performances. Firstly, pre-training with Kinetics-400 outperforms the rest (i.e., 64.82 vs 62.85 vs 28.90 on EGTEA Gaze++, 95.35 vs 91.65 vs 91.60 on UCF-101). We project features of 60 categories from EGTEA Gaze+ with large improvements by t-SNE and find that the larger an upstream dataset is, the more scattered a downstream feature space are (Figure 5). Also, we study the impacts of freezing norm. Freezing “layer-norms” is beneficial for fine-tuning from Kinetics-400 weights, whereas harmful for ImageNet-21k weights. Because the Kinetics-400 shares the same modality (Video) with EGTEA Gaze+, while ImageNet-21k is different (Image). We verify the efficacy of two-stream TokShift-xfmr by adding the optical-flow (64.82→65.08). Finally, we compare the TokShift with strong 3D-CNNs (I3D, P3D & TSM), and verify its efficacy on small-scale datasets.

5 CONCLUSION

We propose a zero-parameter, zero-FLOPs TokShift operator for constructing a pure convolutional-free video transformer. Specifically, our TokShift-xfmr alleviates intensive pair-wise distance calculations of the Spatio-temporal attention, maintains the same computations as a common 2D ViT, while achieving better or comparable performance (80.40%) as 3D-CNN SOTAs. More importantly, TokShift firstly conducts temporal modeling on the global token of video transformers. Since global [Class] token is aggregated by

---

\(^1\)https://github.com/jeanworld/ViT-pytorch/blob/main/visualize_attention_map.ipynb
weighted-sum all spatial patches’ embeddings and reflects global visual content. Hence, modeling temporal interactions across frames can be done through the tokens. Besides, we still face an inherent computational burden of attention, within the 2D ViT, especially in processing long-length video. We leave computational optimization of video transformer as future works. Finally, with the help of our TokShift-xfmr, we can partially answer the prophecy of the quote on video: “A 10 seconds video is worthy of 3,152 visual words”.

| Model                | Pretrain        | Res (H × W) | # Frames T | UCF101 Acc1 (%) |
|----------------------|-----------------|-------------|------------|-----------------|
| I3D [4]              | Kinetics-400    | 224 × 224   | 250        | 84.60           |
| P3D [21]             | Kinetics-400    | 224 × 224   | 16         | 84.20           |
| Two-Stream 3D [4]    | Kinetics-400    | 224 × 224   | 500        | 93.40           |
| TSM [20]             | Kinetics-400    | 256 × 256   | 8          | 95.80           |
| ViT (Video) [9]      | ImageNet-21k    | 256 × 256   | 8          | 91.46           |
| TokShift             | None            | 256 × 256   | 8          | 91.69           |
| TokShift*            | ImageNet-21k    | 256 × 256   | 8          | 91.65           |
| TokShift**           | Kinetics-400    | 256 × 256   | 8          | 95.85           |
| TokShift** (HR)      | Kinetics-400    | 384 × 384   | 8          | 96.14           |
| TokShift-Large* (HR) | Kinetics-400    | 384 × 384   | 8          | 96.80           |
