K-centered Patch Sampling for Efficient Video Recognition

Seong Hyeon Park[1], Jihoon Tack[1], Byeongho Heo[2], Jung-Woo Ha[2], and Jinwoo Shin[1]

[1] Korea Advanced Institute of Science and Technology (KAIST)
{seonghyp,jihoontack,jinwoos}@kaist.ac.kr
[2] NAVER AI LAB {bh.heo,jungwoo.ha}@navercorp.com

Abstract. For decades, it has been a common practice to choose a subset of video frames for reducing the computational burden of a video understanding model. In this paper, we argue that this popular heuristic might be sub-optimal under recent transformer-based models. Specifically, inspired by that transformers are built upon patches of video frames, we propose to sample patches rather than frames using the greedy K-center search, *i.e.*, the farthest patch to what has been chosen so far is sampled iteratively. We then show that a transformer trained with the selected video patches can outperform its baseline trained with the video frames sampled in the traditional way. Furthermore, by adding a certain spatiotemporal structuredness condition, the proposed K-centered patch sampling can be even applied to the recent sophisticated video transformers, boosting their performance further. We demonstrate the superiority of our method on Something–Something and Kinetics datasets.

Keywords: Patch sampling, video transformers, efficient video recognition, K-center search, farthest point sampling

1 Introduction

Video recognition, *i.e.*, recognizing events in a sequence of image frames, in real-world scenarios is an important yet challenging problem in computer vision [21]. Typically, the challenges originate from the dimensional complexity due to spatiotemporal characteristics of video data, *i.e.*, one has to design a model to handle both spatial information and temporal extent simultaneously. In this respect, transformer-based architectures [45] have recently shown remarkable performance for video recognition tasks [1, 35], following their success in both spatial [3] and temporal [2, 37] domains; they map each video frame to non-overlapping image patches and model the temporal sequence of frames as a sequence of patches.

Video transformers are, however, notorious for their high computing demand and CO_2 footprint [39], mainly due to the quadratic computational complexity (with respect to the number of patches) of the self-attention mechanism [45]. To address the issue, video transformers simply choose a subset of video frames to

Table 1. Effects of sampling methods. We experiment with a simple extension of a DeiT-base (an image transformer) [43] to videos. We report Top-1 and Top-5 classification accuracies (%) on Something-Something v2 dataset. The bold denotes the best results. Values in parenthesis are relative improvements compared to the Random Frame sampling scheme.

Sampling methods	Top-1	Top-5
Random Frame	58.95 (-)	85.03 (-)
Random Patch	61.64 (△4.56%)	86.32 (△1.52%)
K-centered (ours)	**64.31**(△9.09%)	**90.38**(△6.29%)

(a) K-centered sampling **(b)** Frame-based sampling

Fig. 1. Our proposed K-centered sampling and the frame-based sampling methods demonstrated in a Something-Something v2 video clip. The highlighted region denotes the sampled parts. As depicted, our method finds diverse and important patches from all frames, while the frame-based sampling ignores most of the frames. Note that the number of sampled patches are equal in (a) and (b).

process and reduce the range of the self-attention to secure feasible computation costs, *e.g.*, dividing the spatiotemporal self-attention [3] or using local temporal self-attention [4]. Our motivation here is that the common heuristics of partially sampling frames from a long video [14, 51], which we refer to as the frame-based sampling methods, can be improved or replaced.

Before the video transformers emerge, the frame-based sampling has been an inevitable design choice for prior convolutional architectures that require a regularly-structured input (*e.g.*, complete frames). However, it may not be an optimal choice with respect to representing the entire video information succinctly; the frame-based sampling might ignore an entire important frame and even contain redundant features, *e.g.*, backgrounds, with high probability [46, 52]. In particular, the rationality of using this sampling strategy in video transformers is more questionable since transformers handle videos as patches (not frames); hence, they do not require such regular structures of the frame to be sampled.

Contribution. In this paper, we indeed argue that the popular frame-based sampling can be a sub-optimal choice for transformer-based models under several video recognition benchmarks. For example, we found that even randomly sampling patches outperforms the frame-based sampling under a simple extension of a vision transformer [43] for video (see Table 1 for the details).

Motivated by this finding, we propose a simple yet effective patch-based sampling strategy suitable for recent transformer-based video recognition models. To be specific, our scheme utilizes the greedy K-center search [16, 19], where patches farthest from each other in the geometric distance[3] are selected iteratively. Intuitively, such a strategy forces models to sample diverse yet discriminative patches from the video. Hence, models are expected to learn more informative spatiotemporal features. As in Fig. 1, our sampling scheme selects patches that contain objects and humans, while frame-based sampling contains many redundant background parts. By utilizing our scheme in a vision transformer [43], we show that it outperforms both the frame-based sampling and the uniform random sampling, as reported in Table 1.

We further show that the proposed K-centered patch sampling strategy is even applicable to sophisticated transformers variants [3, 4, 35] on video domain that assumes a regular spatiotemporal structure in input patches. To this end, we extend our scheme to control the level of structuredness in the sampling by enforcing a constraint on how the sample patches are distributed in spatiotemporal regions of a given video. As a result, our method is compatible not only with a plain vision transformer architecture but the sophisticated variants of video transformers; one can achieve performance improvements purely due to the replacement of their frame-based sampling method with our patch-based sampling scheme.

We demonstrate the efficacy of our method through evaluations on video recognition datasets, including Kinetics 200 & 400 [22, 49] and Something-Something v2 (SSv2) [18] datasets. Overall, our scheme consistently improves the action classification accuracy compared to the frame-based sampling across various transformer architectures ranging from a simple vision transformer to recent video transformers, *e.g.*, it improves the top-1 classification accuracy by (relatively) up to 4.80% (71.42% $\rightarrow$ 74.85%) and 2.13% (62.50% $\rightarrow$ 63.83%) for Kinetics-400 and SSv2, respectively.

The importance of how to manipulate video data for building efficient video understanding models has been largely dismissed in the literature. Along with recent developments of video transformer architectures, we show that the conventional frame sampling method can underperform a new simple patch-based sampling scheme. We believe that our work would inspire many new future intriguing directions for the important problem. Our code is available at https://github.com/kami93/kcenter_video.

2 Related Works

Convolution-based video recognition. Recent works approach video recognition tasks as a temporal extension of image classification. To this end, most of the prior works suggested a temporal extension of convolutional neural networks (CNNs) based on the remarkable success of CNNs in modeling spatial

[3] Denotes a vector distance between patches; not a distance between patch's coordinates in the video.

distributions, *e.g.*, 2D CNNs with an additional network for temporal dimension [10, 40] and 3D CNNs for joint spatiotemporal modeling [5, 15, 44]. However, despite the great success of CNNs in video recognition, recent works found vision transformer-based models outperform their CNN counterparts [1]. Upon this success of transformer-based video models, we develop a new patch-based sampling method for an efficient video transformer.

Transformer-based video recognition. The transformer-based architecture has recently gained interest for modeling both spatial [25, 43] and temporal distributions [9, 50]. Following this line of research, recent works considered the temporal extension of vision transformer (ViT) [11] for an effective spatiotemporal understanding of videos [12, 23, 26, 34, 35]. However, due to the computational burden of the self-attention mechanism [45] of the transformer, only recently, some proposed efficient self-attention methods to overcome this issue [3, 4]. In this paper, we tackle this problem in a different yet orthogonal direction by suggesting a new efficient patch sampling strategy from the video.

Efficient video recognition. Due to the high dimensional nature of video datasets, various works have focused on developing efficient video recognition frameworks, *e.g.*, architectural design [13, 31], frame quantization [41], and resolution control [30]. Among various approaches, frame-based sampling, *e.g.*, randomly sampling frames from a long video, is one the most commonly used scheme for efficient learning [5, 21, 44]. In this regard, more advanced frame-based sampling methods have been proposed, *e.g.*, Wu *et al.* [48] sample relevant frames with reinforcement learning, and Gowda *et al.* [17] consider the relationship between the selected frames when sampling. Nevertheless, these sampling methods were mainly built upon convolution-based architectures. In this paper, we propose a sampling strategy that is specialized for video transformers.

Efficient token pooling. Recent works found that vision transformers often rely on a small portion of patches, *i.e.*, also referred to as tokens, when classifying the objects [33,38]. Motivated by this, there were several attempts to drop or aggregate redundant tokens at the internal feature space for efficient training [6, 24, 28, 29]. Concurrent to our work, Wang *et al.* [47] proposed a token selection scheme for video transformers by pooling informative tokens in both spatial and temporal dimensions. While it is maybe seemingly similar to our work, we remark that our work does not access any internal representation of transformers. Therefore, it is orthogonal to the prior token pooling works.

3 *K*-centered Patch Sampling

In this section, we compare the frame-based and the patch-based sampling approaches and explain how the latter can be applied to video data (Sec. 3.1). Then, we describe our proposed K-center search for more enhancing patch-based

sampling of a video (Sec. 3.2). Finally, we introduce an extension of our method for controlling the amount of the structural characteristics in a set of patch-based samples of a video, which is required and exploited by sophisticated video transformer architectures (Sec. 3.3).

As for the notations, we use plain lower-case letters to denote vectors (*e.g.*, $x \in \mathbb{R}^D$), boldface lower-case letters for multi-dimensional matrices (*e.g.*, $\boldsymbol{x} \in \mathbb{R}^{N\times D}$ and $\boldsymbol{y} \in \mathbb{R}^{N\times M\times D}$), and other letters are scalars unless otherwise defined (*e.g.*, an upper-case letter $X \in \mathbb{R}$). Letters with subscripts are used to denote the ith element given an ordered finite set (*e.g.*, $x_i \in \mathbb{R}^D$ given $\{x_1, ..., x_N\} \equiv \boldsymbol{x} \in \mathbb{R}^{N\times D}$).

3.1 Patch-based Sampling

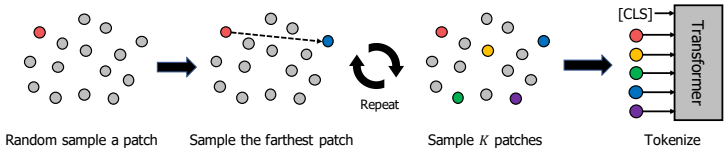
Consider a video clip $\boldsymbol{v} \in \mathbb{R}^{T\times H\times W\times C}$, where T is the frame length, $H \times W$ is the spatial resolution, and C is the number of channels (*e.g.*, for RGB, $C = 3$) for the video. The frame-based sampling has been a common standard in video recognition models, which maps $\boldsymbol{v}$ to a matrix of sample frames $\boldsymbol{f} \in \mathbb{R}^{F\times H\times W\times C}$, by indexing the frame indices (*i.e.*, the time dimension) so that the frame length to be used for recognition is reduced to $F < T$. The frame-based sampling has served as a core component to curtail the computational burden in video recognition tasks, specifically well mingling with video convolutional neural networks (CNNs).

Meanwhile, the emergence of transformer models in image recognition tasks [11,20,25,43] has also brought a paradigm shift in video models [1,3,4,35], where a video should be viewed as a set of patch vectors instead of the long-established $H \times W$ grid view. These patch vectors, each of which serves an elementary input entity in vision transformers, are derived by rearranging an image (or equivalently a video frame) $\boldsymbol{i} \in \mathbb{R}^{H\times W\times C}$ into a patch matrix $\boldsymbol{p} \in \mathbb{R}^{\left(H\cdot W/P^2\right)\times D}$, where $D = P^2 \cdot C$ and P is a patch's edge length as initially proposed by ViT [11]. It is straightforward to prolong this concept for videos by simply prepending an additional time dimension and considering a rearrange mapping $\boldsymbol{v} \in \mathbb{R}^{T\times H\times W\times C} \mapsto \boldsymbol{p} \in \mathbb{R}^{\left(T\cdot H\cdot W/P^2\right)\times D}$. In this way, the video is presented as a matrix with its rows composed of patch vectors (without forcing explicit multi-dimensional structure present in the indices; *i.e.*, the indices are flattened). Note that the rearrange operation does not materialize a new data matrix nor require any extra FLOPs, as it just produces an alternate view of the same data.

Nevertheless, the previous art for video transformers is mostly built upon the traditional frame-based scheme for sampling[4]: $\mathbb{R}^{T\times\left(H\cdot W/P^2\right)} \mapsto \mathbb{R}^{F\times\left(H\cdot W/P^2\right)} \mapsto \mathbb{R}^{\left(F\cdot H\cdot W/P^2\right)}$. Instead, we argue that, for the transformers, directly sampling on the patches—$\mathbb{R}^{\left(T\cdot H\cdot W/P^2\right)} \mapsto \mathbb{R}^K$—can be more effective, and we refer to this alternate sampling scheme as the patch-based sampling. Likewise the frame-based sampling, there can also be many instances of patch-based sampling, including the grid-sampling, random sampling, and our proposed K-centered sampling that we describe in the following subsection.

[4] We omit the last dimension D for simplicity.

(a) Pre-process a video clip into patches



(b) Select K-centered patches with greedy farthest point sampling

Fig. 2. Concept figure of our K-centered patch sampling for video transformers. Our method first (a) pre-processes a video clip into a set of patches and maps each patch into a vector with a fixed position encoding. Then (b) sample K patches by running the greedy farthest point sampling algorithm. Note that we utilize a learnable positional embedding when training the video transformers.

3.2 Greedy K-center Search

Given a video represented by a set of patch vectors $\boldsymbol{p} \equiv \{p_1, ..., p_N\} \in \mathbb{R}^{N \times D}$, we propose to employ the greedy K-center searching [16, 19], also known as the farthest point sampling (FPS) [36] algorithm, for sampling K patches $\boldsymbol{p'} \equiv \{p'_1, p'_2, ..., p'_K\} \in \mathbb{R}^{K \times D}$. It basically chooses a subset of patches iteratively, in a way that p'_k is the farthest vector in geometric distance (*e.g.*, Euclidean norm $\|\cdot\|$ of the difference) with respect to previously sampled patches, as described by (1):

$$p'_k = \underset{p_i \in \boldsymbol{p} \setminus \{p'_1, ..., p'_{k-1}\}}{\operatorname{argmax}} \min_{p'_j \in \{p'_1, ..., p'_{k-1}\}} \left\| p_i - p'_j \right\|. \tag{1}$$

The complexity of the greedy K-center searching is $\mathcal{O}(KN)$, and it has long been considered one of the valid options for sampling from a set, known to have better coverage over the original set compared to random sampling methods [16, 19, 36].

The motivation for choosing greedy K-center search, however, is not limited to its computational efficiency or the excellence in the coverage but includes its generality and flexibility in designing sampling attributes. For example, while sampling algorithms such as random sampling or grid sampling determine the sampling indices solely based on the coordinate of patches in videos, we might want to consider the other attributes, such as the RGB feature of patches.

Since the greedy K-center search is built upon comparing mutual distances between vectors, adding extra attributes can simply be done by concatenating any attributes to the vectors. In other words, we can build K-center considering

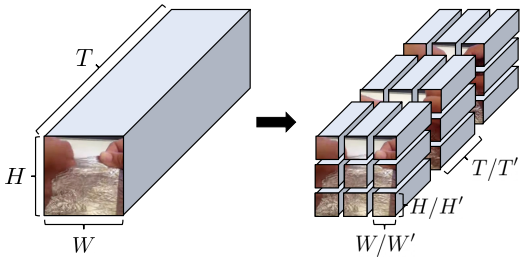


Fig. 3. Dividing a video into sub-video regions. The H, W, and T, denote the height, weight, and time dimension of the video clip, respectively. The H', W', and T', denote the division parameters for each height, weight, and time dimension, respectively.

spatial and temporal distance by concatenating spatiotemporal coordinates to the RGB color vectors.

Concerning the scale, without loss of generality, we normalize the range of the patch's values (*e.g.*, the RGB values) to $\left[0, D^{-\frac{1}{4}}\right]$. Then, we encode the spatiotemporal coordinate of a patch v_i, $T_i \in [1, T]$, $H_i \in [1, H]$ and $W_i \in [1, W]$ to a normalized positional encoding $\left(\frac{T_i-1}{T-1}, \frac{H_i-1}{\sqrt{2}(H-1)}, \frac{W_i-1}{\sqrt{2}(W-1)}\right)$ to concatenate it with the patch vector prior to performing the K-center search based on color shape, spatial and temporal distance. In addition, we introduce the coefficients ω_s and ω_t, which give priority between colors, temporal distance, and spatial distance upon searching:

$$\tilde{p}_i = \left(p_i; \left(\frac{\omega_t\,(T_i-1)}{T-1}, \frac{\omega_s\,(H_i-1)}{\sqrt{2}(H-1)}, \frac{\omega_s\,(W_i-1)}{\sqrt{2}(W-1)}\right)\right). \tag{2}$$

By adjusting ω_s and ω_t in (2), one can customize how the samples are chosen given a vector. Some examples are depicted in Fig. 6.

3.3 Structure-aware K-Center Search

Although the greedy K-center search is directly applicable to general vision transformer models, we may further consider an extension of our method toward video transformers.

In general, video transformers are composed of sophisticated attention mechanisms that leverage the structured shape in frame-based inputs (*i.e.*, inputs with $T \times H \times W$ shape). Since patch-based sampling methods inherently break this structure by rearranging a video into a 1-dimensional sequence of vectors, it is often non-trivial for them to sample patches without losing a certain structure demanded by video transformers.

To preserve the structural information for video transformers, we design an extension of our method coined *structure-aware* K-centered sampling, which

controls the amount of structuredness by defining a video with a set of spatiotemporal chunks as shown in Fig. 3, which enforces the same numbers of patches to be sampled for every chunk.

To be specific, we introduce the division parameters T', H' and W' to our method and consider the set of sub-tensor regions of an input video $\boldsymbol{v} \equiv \{\bar{\boldsymbol{v}}_1, ..., \bar{\boldsymbol{v}}_S\}$ where $\bar{\boldsymbol{v}}_s \in \mathbb{R}^{\frac{T}{T'} \times \frac{H}{H'} \times \frac{W}{W'} \times C}$. Then, throughout the iterations for K-centered sampling, we track the number of patches sampled at each region and constrain them to be equal in all regions.

As a consequence, the resultant sample patches from the structure-aware K-centered sampling can have a certain amount of spatiotemporal structuredness controlled by the division parameters T', H' and W'. Intuitively, the higher we set the values for the division parameters, the more structuredness in the sample patches would be preserved by the sampler. We employ this extension to enable our method to be compatible with recent video transformer architectures.

Finally, we also consider another way to bring out extra structuredness in inputs by combining patch sampling with frame sampling, which we refer to as hybrid sampling. Specifically, we grid-sample some frames from a video, and then we conduct patch sampling from the remaining frames. It is just the same as patch sampling on a video with fewer frames. We find empirically that hybrid sampling is sometimes useful to further boost the model performance in a practical scenario.

4 Experiments

We verify the effectiveness of our technique on video action classification datasets. First, the performance gain achieved by our K-centered patch-based sampling compared to the frame-based sampling is investigated. To do so, we compare the classification accuracies of various transformers, varying from a naïve extension of ViT [11] (an image transformer) tweaked for video processing to recent sophisticated video transformers [3, 4, 35] trained with the two sampling methods in different datasets. Our results exhibit that incorporating our K-centered sampling into these transformer models generally provides performance gains without architectural modification. Finally, we perform various analyses to understand the effect of our patch-based sampling in video recognition.

4.1 Experimental Setups

Datasets. We evaluate the performance of our proposed models on two sets of video classification benchmarks:

– **Kinetics 400 & 200 [22, 49].** Kinetics-400 consists of 10-second videos sampled at 25 frame-per-second from YouTube, categorized into 400 action classes. For the ablation study, we evaluate on Kinetics-200 dataset, which is a subset with 200 classes randomly sampled from the full Kinetics-400 dataset. As Kinetics is a dynamic dataset (videos may be removed from YouTube),

we note our training and validation dataset sizes are approximately 240,000 and 20,000, respectively, for Kinetics-400, and 80,000 and 5,000, respectively, for Kinetics-200.
– **Something–Something v2 (SSv2) [18].** SSv2 contains 220,847 videos, with 168,913 in the training set, 24,777 in the validation set, and 27,157 in the test set. The dataset consists of 174 labels and the video duration range from 2 to 6 seconds. In contrast to the other video datasets, the objects and backgrounds are consistent across different action classes. Hence, the model requires understanding temporal motion to generalize on SSv2 dataset.

Network architectures. We compare the video classification performance of the patch-based sampled transformer (our method) with frame-based baselines comprising various recent video transformer models [3, 4, 11, 35]. For each model, we consider several different choices of the number of the input frames considered by models (or, equivalently, the number of patches for our model). For fair comparisons, we use the ImageNet-1k [8] pre-trained ViT-Base model[5] when initializing all video transformers. The details of each architecture are as:

– **ViT** [11]. ViT learns the spatial relationship of the patches with a learnable position encoding which is added during the tokenization step. To extend ViT to video recognition, we additionally consider temporal positional embedding when tokenizing the patches.
– **TimeSformer** [3]. TimeSformer adapts the ViT to video by proposing a divided space-time attention mechanism to learn the temporal feature in an efficient manner. We use the base TimeSformer architecture, which requires 8 frames as an input, *i.e.*, equivalent to 1,568 patches.
– **Motionformer** [35]. Motionformer extends the ViT with a newly proposed trajectory attention which implicitly aggregates the motion path of the object. We utilize the base Motionformer model, which processes 1,568 tokens.
– **XViT** [4]. XViT proposes local temporal attention and a space-time mixing scheme to reduce the computational burden of the full self-attention mechanism. We use the XViT model with 16 frames input for SSv2 dataset and 8 frames input for Kinetics dataset.

Training details. For all experiments, we follow most training details of Patrick *et al.* [35]—especially dataset augmentation methods—for training all baselines and our method. Concretely, we use the AdamW optimizer [27] with a weight decay of 0.05, label smoothing [42] with a smoothing constant of 0.2, and mixed precision training [32]. For the data augmentation, we utilize random frame selection, random size jittering, and random crop for all datasets. For SSv2, we additionally utilize RandAugment [7] with a severity magnitude of 20 and one augmentation operation per video clip; note that we apply the same augmentation per video clip. Unless otherwise specified, we set both the spatial and temporal search coefficients of K-centered patch sampling, *i.e.*, ω_s, ω_t, to 1.0.

[5] We utilize the original ViT-B weights to ensure the fair comparison, unlike Table 1 that utilizes DeiT-base.

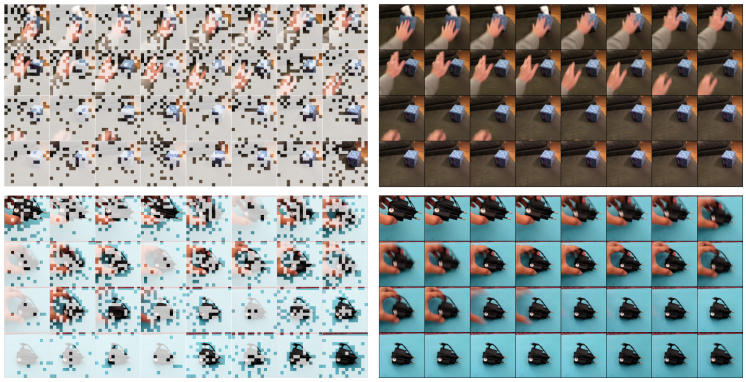
Table 2. Comparison of our K-centered sampling and frame-based sampling methods in Kinetics-400 (K400) and Something-Something v2 (SSv2) datasets. We report Top-1 and Top-5 action classification accuracies (%) on validation sets of the datasets. Computational budgets for forwarding a video sample are equal between the same model. For a fair comparison, we unify all models to use the same ImageNet-1k pre-training and evaluation protocols. The bold denotes the best result.

Model	Sampling	K400		SSv2	
		Top-1	Top-5	Top-1	Top-5
ViT [11]	Frame-based	74.80	91.65	62.50	88.06
	K-centered (ours)	**75.65**	**92.04**	**63.83**	**88.89**
TimeSformer [3]	Frame-based	77.95	**93.29**	63.76	88.53
	K-centered (ours)	**77.98**	93.09	**63.81**	**88.59**
Motionformer [35]	Frame-based	71.42	88.62	61.79	86.82
	K-centered (ours)	**74.85**	**91.27**	**62.15**	**87.19**
XViT [4]	Frame-based	72.73	90.25	62.40	87.82
	K-centered (ours)	**73.05**	**90.48**	**62.78**	**88.27**

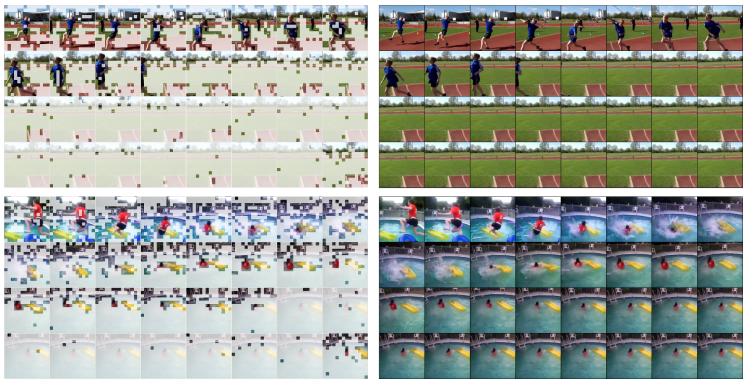
Inference details. For inference, we follow the previous benchmark standard of 3×1 ensemble testing [3]. To be specific, we sample a fixed-length clip from the video and then crop 3 different spatial views (top-left, center, bottom-right) from the clip. The final prediction is made by averaging the scores for all crops. We fix the same inference video pre-processing for all baselines and our method for a fair comparison.

4.2 Main Results

Comparison with frame-based sampling. We consider comparing our K-centered sampling with conventional frame-based sampling strategy. To this end, we adapt our scheme to various video transformer models. As shown in Table 2, our K-centered sampling consistently outperforms the frame-based sampling on Kinetics-400 and SSv2 datasets. For instance, for Motionformer on Kinetics-400 dataset, our sampling improves the Top-1 action classification accuracy by 4.8% relatively (71.42% $\rightarrow$ 74.85%). Moreover, our sampling uniformly improves the accuracy for all models in the SSv2 dataset (*e.g.*, relative 2.86% Top-1 accuracy gain on ViT), where it especially requires a high level of temporal understanding. This indicates that our sampling scheme contains more temporal dynamics information than conventional frame-based sampling. In video models that are highly optimized under the frame-based inputs (e.g., TimeSformer [3] assumes a fixed number of patches in space and time axes), the gain from our method could diminish. We believe developing new architectures optimized for K-centered patch-based sampling would be interesting future work.



(a) Something-Something v2



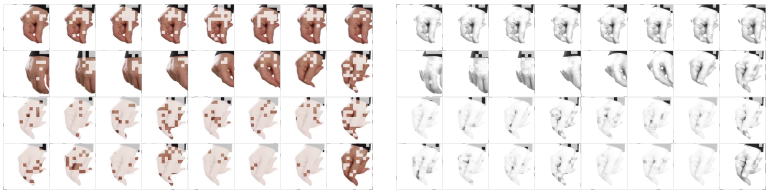
(b) Kinetics-400

Fig. 4. Visualization examples of our K-centered patch sampling on Something-Something v2 and Kinetics-400 dataset. The highlighted region in the left video denotes the sampled patches, and the right video indicates the original video.

Visualization of sampled patches. Fig. 4 visualizes the patches sampled by our strategy. One can observe that our sampling tends to focus on the target objects while less frequently sampling the redundant patches, *e.g.*, backgrounds. For instance, most of the sampled patches in the first example of Kinetics include the moving human rather than the repeating background patches with similar colors. This supports the result that our methods show effectiveness for datasets with complex temporal dynamics.

4.3 Ablation Study

Since our study is the first introduction of patch sampling in video transformers, we intend to provide various analyses to facilitate future video transformer research using patch sampling. We demonstrate the effects of using color channels



(a) RGB-patch sampling **(b)** Grayscale-patch sampling

Fig. 5. Visualization of sampled patches by K-centered sampling on (a) RGB patches and (b) Grayscale patches in a Kinetics video clip. The highlighted region denotes the sampled parts.

Table 3. Comparison of our K-centered sampling with RGB patches and Grayscale patches in Kinetics-200 dataset. We report Top-1 and Top-5 classification accuracies (%). Note that Grayscale patches are used only for determining patch indices, and the actual inputs are normal RGB patches.

Model	Channel	Top-1	Top-5
ViT [11]	RGB	82.90	95.42
	Grayscale	81.94	95.58
TimeSformer [3]	RGB	84.24	96.10
	Grayscale	83.60	96.00

for K-centered sampling and of sampling hyperparameters—hybrid sampling and distance coefficients.

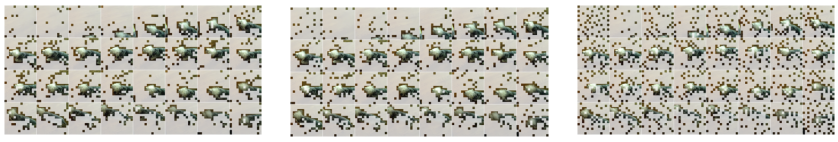
Effect of color channels in sampling distance. We notice that the color information is important for the patch sampling as it directly changes the distance of the patches. To investigate the effect of the color information, we consider sampling the patch index with a Grayscale video when measuring distances between patches[6]. As shown in Table 3, the patch selection with RGB videos shows better performance compare to the patch sampling with Grayscale videos. Also, in Fig. 5, we visualize the sampled patches from each configuration.

Effect of the hybrid sampling. Depending on the choice of models and datasets, sometimes, the full patch-based sampling leads to sub-optimal results. For the Kinetics dataset, specifically, the full patch-based sampling underperforms the frame-based sampling. In this respect, we consider taking advantage of both patch and frame sampling; we introduce the hybrid sampling-X, where X denotes the number of complete frames involved in the sampled patches. Table 4 depicts the number of hybrid sampling and their corresponding performances measured in

[6] Grayscale is only used for sampling, *i.e.*, we use RGB patches for network input.

Table 4. Effect of hybrid sampling, *i.e.*, sampling both frames and patches, on ViT under Kinetics-200 dataset. Note that the total sampled areas are equal. We report Top-1 and Top-5 classification accuracies (%). The bold indicates the best result.

Metric	Hybrid Sampling							
	0	1	2	3	4	5	6	7
Top-1	81.56	81.90	81.94	**82.90**	82.38	82.58	82.62	82.00
Top-5	95.40	95.26	95.40	95.42	95.46	**96.48**	95.54	95.48



$(\omega_s, \omega_t) = (1, 1)$ $(\omega_s, \omega_t) = (2, 2)$ $(\omega_s, \omega_t) = (10, 10)$

Fig. 6. Visualization of our K-centered patch sampling under different distance coefficients ω_s and ω_t upon search. The highlighted region in the video denotes the sampled patches, and the numbers of sampled patches are equal. Higher coefficients force models to sample patches with large spatiotemporal diversity.

Kinetics-200 dataset. We find that for ViT, hybrid sampling leads to meaningful performance improvements, *i.e.*, 81.56% to 82.90% improvement in Top-1 accuracy when 3 frames were used in the hybrid sampling. Intriguingly, patch-based sampling only was enough for Motionformer, *i.e.*, the best performance is observed when no hybrid frames are used.

Effect of spatiotemporal distance coefficients. We also investigate the effect of spatiotemporal distance coefficients ω_s, ω_t for the K-centered patch sampling. As shown in Fig. 6, samples under higher search coefficients tend to be more diversely distributed in spatiotemporal regions of video. We empirically test some selections for w_s and w_t for our experiments as done similarly to Table 5 and report the best performance among them.

Effect of the test-time sampling. We also demonstrate the effect of test-time sampling by dropping some proportion of patches or dropping patches and swapping the sampling methods used in test time (*e.g.*, training with K-centered sampling, then testing with frame-based sampling). As shown in Table 6, models trained with our K-centered sampling strategy generally outperform models trained with the frame-based sampling, up to dropping 75% of the input patches, when tested with the identical sampling method to the training time. Interestingly, in cases where dropping an extremely high portion of patches, applying the frame-based sampling at *test time* can be useful to avoid sudden performance drop

Table 5. Effect of distance coefficients ω_s, ω_t in the patch sampling. We report Top-1 and Top-5 classification accuracies (%) on ViT trained with Kinetics-200 dataset. The bold indicates the best result.

Metric	Sampling Coefficients				
	$\omega_s = 1$ $\omega_t = 1$	$\omega_s = 1$ $\omega_t = 2$	$\omega_s = 2$ $\omega_t = 1$	$\omega_s = 2$ $\omega_t = 2$	$\omega_s = 10$ $\omega_t = 10$
Top-1	**82.90**	82.00	82.38	81.72	81.56
Top-5	95.42	**95.74**	95.43	95.26	95.16

Table 6. Effect of test-time sampling on ViT trained with Kinetics-200 dataset. We report the Top-1 classification accuracy (%) by (a) controlling the sampling method and (b) reducing the number of frames (or patches) at test time. The bold indicates the best result. For K-centered samples, we drop those patches sampled in the latest search iterations. For frame-based samples, we drop *frames* (if possible), *i.e.*, dropping $12.5 \times n\%$ indicates dropping $n = 1, 2, \ldots, 7$ frames given the original 8-frame input (for the extreme cases of 93.8 % and 96.9 % drops, we sample the half, and the quarter portion of a frame, respectively).

Train Sampling	Test Sampling	Proportion of droped patches at inference (%)								
		12.5	25.0	37.5	50.0	62.5	75.0	87.5	93.8	96.9
Frame-based	Frame-based	80.20	78.82	77.44	75.64	73.47	69.97	**63.45**	35.85	14.44
	K-centered (ours)	76.38	75.08	73.33	70.41	66.81	59.09	40.39	21.16	9.3
K-centered (ours)	Frame-based	77.56	76.32	75.20	73.99	71.83	67.97	62.79	**37.89**	**15.46**
	K-centered (ours)	**81.32**	**80.92**	**80.16**	**79.28**	**77.16**	**72.89**	57.43	32.65	12.88

observed around dropping 87.5% of input patches. This is because watching a complete frame, albeit static, can give more useful information for recognizing a video when the number of input patches is scarce.

5 Conclusion

In this paper, we address a fundamental issue in video recognition models by arguing that the conventional frame-based sampling approach is sub-optimal for recent transformer-based models processing a sequence of video patches. We propose a new patch-based sampling scheme, coined the greedy K-center search, outperforming the conventional one. We believe our work would guide new future directions for building efficient video understanding models.

Acknowledgement. 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 supported by KAIST-NAVER Hypercreative AI Center.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: IEEE International Conference on Computer Vision (2021) 1, 4, 5
2. Baevski, A., Zhou, H., Mohamed, A., Auli, M.: wav2vec 2.0: A framework for self-supervised learning of speech representations. In: Advances in Neural Information Processing Systems (2020) 1
3. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: International Conference on Machine Learning (2021) 1, 2, 3, 4, 5, 8, 9, 10, 12
4. Bulat, A., Perez-Rua, J.M., Sudhakaran, S., Martinez, B., Tzimiropoulos, G.: Space-time mixing attention for video transformer. In: Advances in Neural Information Processing Systems (2021) 2, 3, 4, 5, 8, 9, 10
5. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 4
6. Chen, B., Li, P., Li, B., Li, C., Bai, L., Lin, C., Sun, M., Yan, J., Ouyang, W.: Psvit: Better vision transformer via token pooling and attention sharing. arXiv preprint arXiv:2108.03428 (2021) 4
7. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Advances in Neural Information Processing Systems (2020) 9
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (2009) 9
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Annual Conference of the North American Chapter of the Association for Computational Linguistics (2019) 4
10. Donahue, J., Hendricks, L.A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: IEEE Conference on Computer Vision and Pattern Recognition (2015) 4
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021) 4, 5, 8, 9, 10, 12
12. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: IEEE International Conference on Computer Vision (2021) 4
13. Feichtenhofer, C.: X3d: Expanding architectures for efficient video recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2020) 4
14. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: IEEE International Conference on Computer Vision (2019) 2
15. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spatiotemporal residual networks for video action recognition. In: Advances in Neural Information Processing Systems (2016) 4
16. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theoretical computer science **38**, 293–306 (1985) 3, 6

17. Gowda, S.N., Rohrbach, M., Sevilla-Lara, L.: Smart frame selection for action recognition. In: AAAI Conference on Artificial Intelligence (2021) 4
18. 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., Memisevic, R.: The "something something" video database for learning and evaluating visual common sense. In: IEEE International Conference on Computer Vision (2017) 3, 9
19. Har-Peled, S.: Geometric approximation algorithms. No. 173, American Mathematical Soc. (2011) 3, 6
20. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: IEEE International Conference on Computer Vision (2021) 5
21. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2014) 1, 4
22. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset (2017) 3, 8
23. Li, K., Wang, Y., Gao, P., Song, G., Liu, Y., Li, H., Qiao, Y.: Uniformer: Unified transformer for efficient spatiotemporal representation learning. arXiv preprint arXiv:2201.04676 (2022) 4
24. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: Expediting vision transformers via token reorganizations. In: International Conference on Learning Representations (2022) 4
25. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE International Conference on Computer Vision (2021) 4, 5
26. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. arXiv preprint arXiv:2106.13230 (2021) 4
27. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019) 9
28. Marin, D., Chang, J.H.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers. arXiv preprint arXiv:2110.03860 (2021) 4
29. Meng, L., Li, H., Chen, B.C., Lan, S., Wu, Z., Jiang, Y.G., Lim, S.N.: Adavit: Adaptive vision transformers for efficient image recognition. arXiv preprint arXiv:2111.15668 (2021) 4
30. Meng, Y., Lin, C.C., Panda, R., Sattigeri, P., Karlinsky, L., Oliva, A., Saenko, K., Feris, R.: Ar-net: Adaptive frame resolution for efficient action recognition. In: European Conference on Computer Vision (2020) 4
31. Meng, Y., Panda, R., Lin, C.C., Sattigeri, P., Karlinsky, L., Saenko, K., Oliva, A., Feris, R.: Adafuse: Adaptive temporal fusion network for efficient action recognition. In: International Conference on Learning Representations (2021) 4
32. Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., Wu, H.: Mixed precision training. In: International Conference on Learning Representations (2018) 9
33. Naseer, M., Ranasinghe, K., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Intriguing properties of vision transformers. In: Advances in Neural Information Processing Systems (2021) 4
34. Neimark, D., Bar, O., Zohar, M., Asselmann, D.: Video transformer network. arXiv preprint arXiv:2102.00719 (2021) 4

35. Patrick, M., Campbell, D., Asano, Y.M., Metze, I.M.F., Feichtenhofer, C., Vedaldi, A., Henriques, J.F.: Keeping your eye on the ball: Trajectory attention in video transformers. In: Advances in Neural Information Processing Systems (2021) 1, 3, 4, 5, 8, 9, 10
36. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems (2017) 6
37. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018) 1
38. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. In: Advances in Neural Information Processing Systems (2021) 4
39. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green ai. arXiv preprint arXiv:1907.10597 (2019) 1
40. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems (2014) 4
41. Sun, X., Panda, R., Chen, C.F., Oliva, A., Feris, R., Saenko, K.: Dynamic network quantization for efficient video inference. In: IEEE International Conference on Computer Vision (2021) 4
42. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 9
43. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning (2021) 2, 3, 4, 5
44. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: IEEE International Conference on Computer Vision (2015) 4
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017) 1, 4
46. Wang, J., Gao, Y., Li, K., Lin, Y., Ma, A.J., Cheng, H., Peng, P., Huang, F., Ji, R., Sun, X.: Removing the background by adding the background: Towards background robust self-supervised video representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11804–11813 (2021) 2
47. Wang, J., Yang, X., Li, H., Wu, Z., Jiang, Y.G.: Efficient video transformers with spatial-temporal token selection. arXiv preprint arXiv:2111.11591 (2021) 4
48. Wu, Z., Xiong, C., Ma, C.Y., Socher, R., Davis, L.S.: Adaframe: Adaptive frame selection for fast video recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 4
49. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp. 305–321 (2018) 3, 8
50. Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: Videogpt: Video generation using vq-vae and transformers. arXiv preprint arXiv:2104.10157 (2021) 4
51. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 2

52. Zhi, Y., Tong, Z., Wang, L., Wu, G.: Mgsampler: An explainable sampling strategy for video action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1513–1522 (2021) 2