MGSampler: An Explainable Sampling Strategy for Video Action Recognition

Yuan Zhi Zhan Tong Limin Wang✉ Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University, China
{yuanzhi,tongzhan}@smail.nju.edu.cn, {lmwang,gswu}@nju.edu.cn

Abstract

Frame sampling is a fundamental problem in video action recognition due to the essential redundancy in time and limited computation resources. The existing sampling strategy often employs a fixed frame selection and lacks the flexibility to deal with complex variations in videos. In this paper, we present a simple, sparse, and explainable frame sampler, termed as Motion-Guided Sampler *(MGSampler). Our basic motivation is that motion is an important and universal signal that can drive us to adaptively select frames from videos. Accordingly, we propose two important properties in our MGSampler design: motion sensitive and motion uniform. First, we present two different motion representations to enable us to efficiently distinguish the motion-salient frames from the background. Then, we devise a motion-uniform sampling strategy based on the cumulative motion distribution to ensure the sampled frames evenly cover all the important segments with high motion salience. Our MGSampler yields a new principled and holistic sampling scheme, that could be incorporated into any existing video architecture. Experiments on five benchmarks demonstrate the effectiveness of our MGSampler over the previous fixed sampling strategies, and its generalization power across different backbones, video models, and datasets. The code is available at* https://github.com/MCG-NJU/MGSampler.

1. Introduction

Video understanding is becoming more and more important in computer vision research as huge numbers of videos are captured and uploaded online. Human action recognition [31, 37, 40, 24] has witnessed great progress in the past few years by designing various deep convolutional networks in videos. The core effort has been devoted to obtaining compact yet effective video representations for efficient and robust recognition. Compared with static images, the extra time dimension requires us to devise a sophisticated temporal module equipped with high capacity and figure out an efficient inference strategy for fast processing. However, in addition to these modeling and computational issues, a more fundamental problem in video understanding is *sampling*. Due to the essential redundancy in time and as well limited computational budget in practice, it is unnecessary and also infeasible to feed the whole video for subsequent processing. How to sample a small subset of frames is very important for developing a practical video recognition system, but it still remains to be an unsolved problem.

Currently, deep convolutional networks (CNNs) typically employ a fixed hand-crafted sampling strategy for training and testing in videos [31, 37, 40]. In the training phase, CNN is trained on frames/clips which are randomly sampled either evenly or successively with a fixed stride from the original video. In the test phase, in order to cover the full temporal duration of video, clips are densely sampled from video and the final result is averaged from these dense prediction scores. There are multiple problems with these fixed sampling strategies. First, the action instance varies with different videos and sampling should not be fixed across videos. Second, not all the frames are of equal importance for classification and sampling should pay more attention to discriminative frames rather than irrelevant background frames.

Recently, some works [47, 45, 5] focus on frame selection in untrimmed videos, and try to improve the inference efficiency with an adaptive sampling module. These methods typically employ a learnable module to automatically select more discriminative frames for subsequent processing. However, these methods heavily rely on the training data with complicated learning strategies, and can not easily transfer to unseen action classes in practice. In addition, they typically deal with untrimmed video recognition by selecting foreground frames and removing background information. But it is unclear how to adapt them to trimmed video sampling due to the inherent difference between trimmed and untrimmed videos.

Based on the analysis above, how to devise a principled and adaptive sampling strategy for trimmed videos still needs further consideration in research. In this paper, we

✉: Corresponding author.

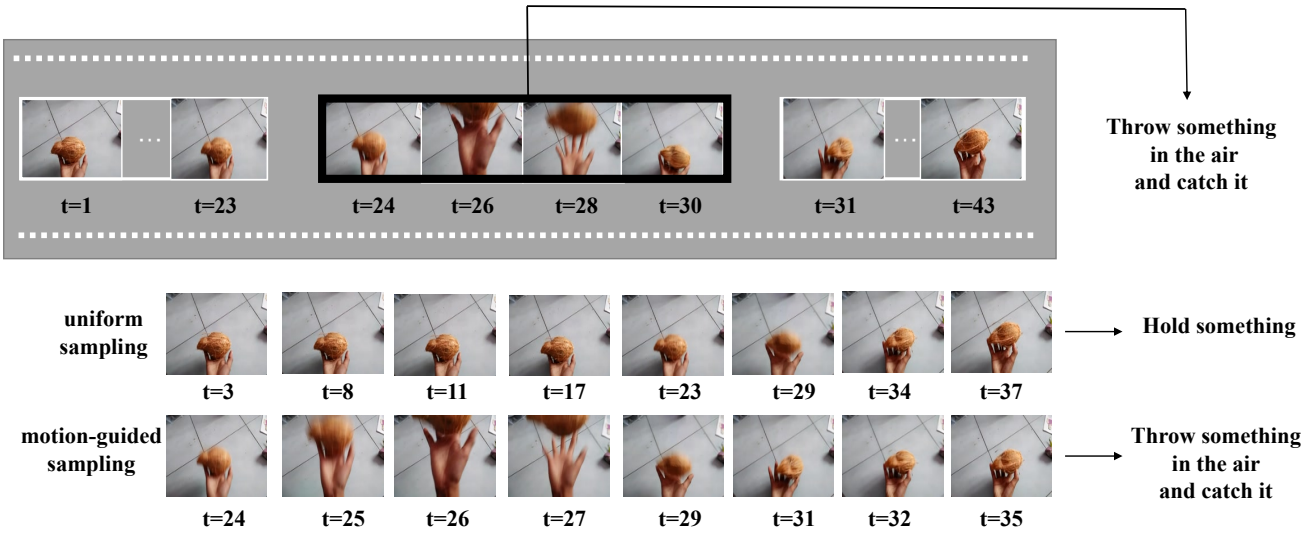


Figure 1. Sample eight frames from a video of throwing something in the air and catching it. Due to the quick moment in action, uniform sampling may miss the key information while our sampling strategy can identify and select frames with large motion magnitude.

aim to present a simple, sparse and explainable sampling strategy for trimmed video action recognition, which is independent of the training data for good generalization ability and also capable of dealing with various video content adaptively. Our basic observation is that motion is a universal and transferable signal that can guide us to sample discriminative frames, in the sense that action related frames should be of high motion salience to convey most information about human movement while background frames typically contain no or limited irrelevant motion information. According to this motion prior, we can roughly analyze the frame importance and group frames into several segments according to their temporal variations. Consequently, these temporal segments enable us to perform a holistic and adaptive sampling to capture most motion information while suppressing the irrelevant background distraction, yielding a general frame sampler (MGSampler).

Specifically, to implement our motion-guided sampling, two critical components are proposed to handle motion estimation and temporal sampling, respectively. For motion representation, we use temporal difference at different levels to approximate human movement information for efficiency. In practice, temporal difference is highly correlated with motion information, and the absolute value of the difference is able to reflect the motion magnitude to some extent. For temporal sampling, based on the motion distribution along time, we present a uniform grouping strategy, where each segment should convey the same amount of motion salience. Then, according to this uniform grouping, we can perform adaptive sampling over the entire video by randomly picking a representative frame from each segment. Figure 1 exhibits a vivid example of sampling frames from a video of the class "throwing something in the air and catching it". The motion relevant content only contains a small portion of the whole video (e.g., from the 24th frame to the 30th frame). If we use traditional uniform sampling, the important information between the 24th frame and 30th frame will be missed and as a result, the video is classified into holding a ball by mistake. In contrast, our motion-guided sampler selects more frames between the 24th frame and the 30th frame and makes a correct prediction.

We conduct extensive experiments on five different trimmed video datasets: Something-Something V1 & V2 [10], UCF101 [32], Jester [27], HMDB51 [18], and Diving48 [23]. Significant improvement is obtained on these datasets by adopting our motion-guided sampling strategy. It is worth noting that using the motion-guided sampling strategy will not increase the burden of computation and running time greatly. In addition, the method is agnostic to the network architecture, and can be used in both training and test phases, demonstrating its strong applicability.

2. Related Work

Action Recognition. Action recognition is a task to identify various human actions in a video. The last decade has witnessed a growing research interest in video action recognition with the availability of large-scale datasets and the rapid progress in deep learning. Methods can be generally categorized into four types: (1) Two-Stream Networks or variants: One stream takes RGB images as input to model appearance and another stream takes optical flow as input to model motion information. In the prediction stage, scores from two streams were averaged in a late fusion way [31]. Based on this architecture, several works were proposed for a better fusion of two streams [8, 42]. (2) 3D CNNs: 3D CNN for action recognition aims to learn features along both spatial and temporal dimensions [15, 36, 7]. However, 3D CNN suffers from more computational cost than their

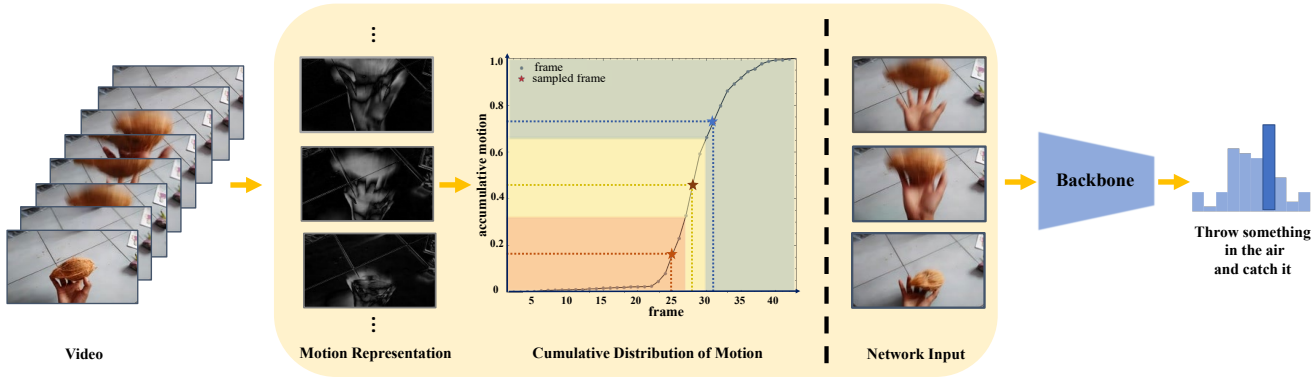


Figure 2. **Motion-guided Sampler (MGSampler).** Our MGSampler aims to on-the-fly select frames containing rich motion information to help the classifier see the whole process of action. Our proposed MGSampler is a general and flexible sampling scheme, that could be easily deployed for any existing video models for action recognition.

2D competitors due to the temporal dimension. In order to reduce computational costs, these works [37, 29] decomposed 3D convolution into a 2D convolution and a 1D temporal convolution or integrated 2D CNN into 3D CNN [53]. (3) Mixed spatiotemporal network models: ECO [54] and TSM [24] designed the lightweight models to fuse spatiotemporal features. MFNet [20], TEINet [25], TEA [22], MSNet [19] and others [16, 33, 43, 44, 39] explored better temporal modeling architecture for motion representation. (4) Long-term network models: short-term clip-based networks are unable to capture long-range temporal information. Several methods were proposed to overcome this limitation by stacking more frames with RNN [48] or long temporal convolution [38], or using a sparse sampling and aggregation strategy [40, 52, 50]. Unlike them, our goal is not designing a better model but devising effective frame sampling for a more fundamental issue in video analysis.

Frame Sampling. For some 3D CNN based methods [36, 2, 7], the video clip is obtained by choosing a random frame as the starting point. Then the next 64 consecutive frames in the video are subsampled uniformly to a fixed number of frames. TSN [40] performed a simple and effective sampling strategy where frames are uniformly sampled along the whole temporal dimension. The above two sampling strategies are commonly used by different models. However, they treated every frame equally and ignored the redundancy between frames, so selecting salient frames or clips conditioned on inputs is a key issue for efficient action recognition. Recently, several works proposed reinforcement learning (RL) to train agents with policy gradient methods to choose frames. FastForward [5] utilized RL for both frame skipping planning and early stop decision making to reduce the computation burden for untrimmed video action recognition. Adaframe [47] proposed a LSTM augmented with a global memory to search which frames to use over time, which was trained by policy gradient methods. Multi-agent [45] uses N agents in the framework and each agent was responsible for selecting one informative frame/clip from an untrimmed video. DSN [51] presented a dynamic version of TSN with RL-based sampling. In order to avoid complex RL policy gradients, LiteEval [46] proposed a coarse-to-fine and differentiable framework that contains a coarse LSTM and a fine LSTM organized hierarchically, as well as a gating module for selecting either coarse or fine features. AR-Net [28] addressed both the selection of optimal frame resolutions and skipping in a unified framework and learned the whole framework in a fully differentiable manner. Audio has also been used as an efficient way to select salient frames for action recognition. SCSampler [17] used a lightweight CNN as the selector to sample clips at test time using salience scores. In order to train the selector effectively, they leveraged audio as an extra modality. Listen to Look [9] used audio as a preview mechanism to eliminate both short-term and long-term visual redundancies for fast video-level recognition. Though these approaches bring improvement in action recognition, their target is long and untrimmed videos rather than short and trimmed videos. What is more, the design of sampling module is usually complex and the training process requires large number of training samples with long training time. Instead, our goal is to present a simple, general, explainable frame sampling module without any learning strategy.

3. Method

In this section, we give detailed descriptions of our motion-guided sampling strategy. First, we give an overview of our motion-guided sampling. Then, we introduce the details of representing motion information of each frame. Finally, we elaborate on the concepts of using the cumulative distribution of motion magnitude to guide the sampling (MGSampler).

3.1. Overview

Videos are composed of a sequence of densely-captured frames. Due to temporal redundancy and limited computa-

tional resource, it is usual to sample a subset of frames to develop an efficient yet accurate action recognition method. Our proposed motion-guided sampling is a general and flexible module to compress the whole video into a fixed number of frames, which could be used for subsequent recognition with any kind of video recognition network (e.g., TSM, TEA, etc.).

Motion prior is the core of our proposed sampling module and we assume this prior knowledge is general and transferable across videos and helpful to devise a universal sampler. Based on this assumption, we devise an adaptive sampling strategy with two important properties: *motion sensitive* and *motion uniform*. Concerning the requirement of motion sensitive, we hope our sampler is able to identify the motion salience along the temporal dimension and distinguish the action-relevant frames from the background. For the property of motion uniform, we expect our sampler can automatically select frames evenly according to the motion information distribution. In this sense, our sampled frames need to distribute uniformly over all of the temporal motion segments to cover the important details of action instances. To accomplish the above requirements of motion-guided sampling, we design two critical components: *motion representation* and *motion-guided sampling*. For motion representation, to balance between accuracy and efficiency, we use the temporal difference to approximately capture the human movement. For motion-guided sampling, we devise a uniform sampling strategy based on the cumulative motion distribution to ensure cover all the important motion segments in the entire videos. Next, we will give detailed descriptions of these two components.

3.2. Motion Representation

As RGB images usually represent static appearance at a specific time point, we need to consider the temporal variations of adjacent frames to leverage temporal context for motion estimation.

Optical flow [12] is a common choice for motion representation, but the high computational cost makes it infeasible for efficient video recognition. Many works have been proposed to estimate optical flow with CNN [4, 13, 6, 30] or explore alternatives of optical flow such as RGBdiff [40], optical guided feature [34], dynamic image [1] and fixed motion filter [21]. Our target is to obtain an efficient yet relatively accurate motion representation to guide the subsequent sampling. We propose two motion representations based on different levels at little computation cost for selecting frames.

Image-level Difference. RGB difference between two consecutive frames describes the appearance change and has the correlation with the estimation of optical flow. Therefore, we adopt image-level difference between adjacent RGB frames as an alternative lightweight motion representation for the proposed sampling strategy. As shown in Figure 3, the image-level difference between frames usually reserves only motion-specific features and suppresses the static background.

Formally, given a frame $I(x, y, t)$ from the video $\mathbf{V} \in \mathbb{R}^{T\times H\times W}$ where T, H, W is the length, height, and width of the video, to formulate its motion magnitude, we first subtract each pixel value of the previous frame $I(x, y, t-1)$ from the current frame $I(x, y, t)$, then accumulate the absolute value of difference values over the spatial domain for each frame:

$$S_t = \sum_{y=1}^{H}\sum_{x=1}^{W} |I(x, y, t) - I(x, y, t-1)|, t \in \{2, 3, \ldots, T\} \quad (1)$$

where S_t describes the motion signal of frame $I(x, y, t)$ and $S_1 = 0$. We further normalize S_t with ℓ_1-norm to obtain motion salience distribution M_t (i.e., $\sum_t^T M_t = 1$).

Feature-level Difference. Although difference between original images can reflect motion information to some extent, more precise patterns such as motion boundaries and textures are hard to capture only by image-level difference.

It is a consensus that convolution has the ability to extract feature and filters in low convolutional layers usually describe boundaries and textures, whereas filters in high convolutional layers are more likely to represent abstract parts. We reemphasize that the main idea in designing motion representation is to achieve a balance between computation and efficiency, so we perform shallow-layer convolution operation on original images. Then, in order to focus on small motion displacements and motion boundaries, we extend the subtraction operation to the feature space by replacing the original image $I(x, y, t)$ by its corresponding feature maps $F(x, y, t)$. The feature-level difference is defined as follows:

$$Diff_i(x, y, t) = F_i(x, y, t) - F_i(x, y, t-1) \quad (2)$$

where the subscript $i \in \{1, 2, \ldots, C\}$ represents the i-th feature map of the original image and C is the number of channel. In experiments, we use one convolutional layer which consists of eight 7×7 convolutions with stride=1 and padding=3, following the design of PA module [49]. The padding operation avoids the reduction in spatial resolution.

Because $Diff(x, y, t) \in \mathbb{R}^{H\times W\times C}$ is three-dimensional, to formulate the feature-level difference, C channels are accumulated to 1 channel by the square sum operation, leading $Diff(x, y, t) \in \mathbb{R}^{H\times W}$. Then all pixel values are added into one value. The mapping $\mathbb{R}^{H\times W\times C} \to \mathbb{R}$ makes $Diff(x, y, t)$ represent the motion magnitude of each frame.

$$S_t = \sum_{y=1}^{H}\sum_{x=1}^{W} \sqrt{\sum_{i=1}^{C} (Diff_i(x, y, t))^2}, t \in \{2, 3, \ldots, T\} \quad (3)$$

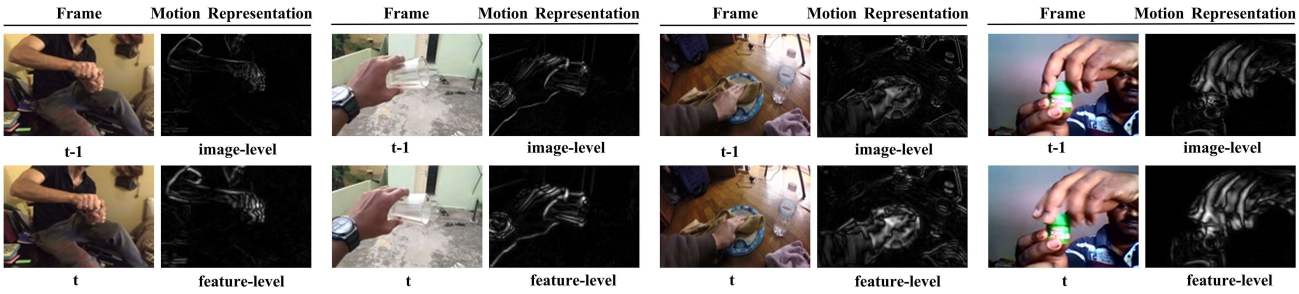


Figure 3. Examples of original frames and its corresponding motion representation.The RGB frame contains rich appearance information and motion representation retains salient motion cues. Compared with image-level difference, feature-level difference captures more detailed and core motion displacement.

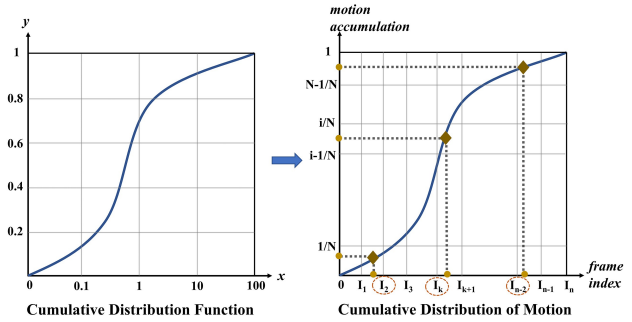


Figure 4. Inspired by the cumulative distribution function.

Like image-level difference, we further normalize S_t with ℓ_1-norm to obtain motion salience distribution M_t, that is $\sum_t^T M_t = 1$.

3.3. Motion-Guided Sampling (MGSampler)

After obtaining the motion salience distribution along time M_t, we are ready to describe how to use it to perform motion-guided sampling. Similar to segment-based sampling in TSN [40], our sampling is a holistic and duration-invariant strategy, in the sense that we sample over the entire video and compress the whole video into a subset of frames. In contrast to TSN which is a fixed sampling strategy, our motion-guided sampling adaptively selects frames according to motion uniform property and hopes the sampled frames could cover the important motion segments. In order to perform sampling adaptively according to motion distribution, we present a temporal segmentation scheme based on the cumulative motion distribution and then randomly sample a representative from each segment.

Specifically, the cumulative distribution function of a purely discrete variable X, having n values $x_1, x_2, \ldots x_n$ with probability $p_i = p(x_i)$ is the defined by the following function:

$$F_X(x) = \mathrm{P}(X \leq x) = \sum_{x_i \leq x} \mathrm{P}(X = x_i) = \sum_{x_i \leq x} p(x_i), \quad (4)$$

where F_X is the accumulation of the probability from x_1 to x_n and ranges from 0 to 1. Furthermore, the cumulative distribution function is non-decreasing and right-continuous.

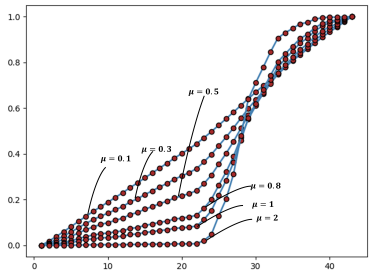


Figure 5. the cumulative motion distribution under different values of μ.

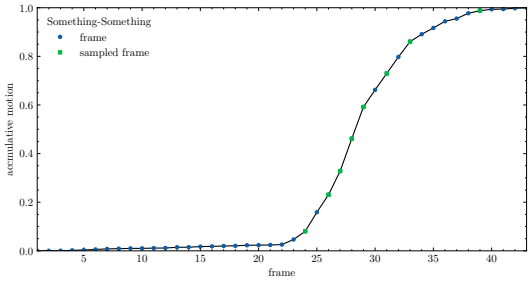
$$F_X(x_0) = 0, \quad F_X(x_n) = 1. \quad (5)$$

Based on this definition of cumulative distribution function, we construct motion cumulative curve along the temporal dimension as shown in Figure 4, where x-axis represents the frame index and y-axis represents the motion information accumulation up to current frame. To further control the smoothness of motion-guided sampling, we introduce a hyper-parameter μ to adjust the original motion distribution M_t:

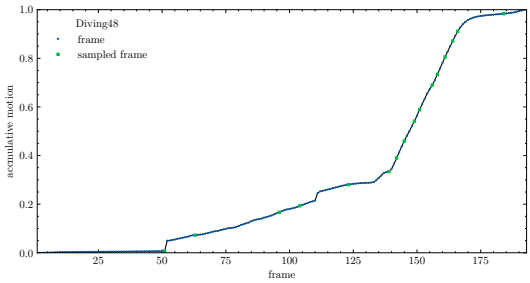
$$\hat{M}_t = \frac{(M_t)^\mu}{\sum_{t=1}^T (M_t)^\mu}. \quad (6)$$

As shown in Figure 5, a lower value for μ produces a more uniform probability distribution of motion magnitude.

According to the obtained motion cumulative distribution curve, we now can perform our motion-guided sampling strategy. In order to sample N frames from the original video, the interval of y-axis is divided into N parts evenly: $\left([0, \frac{1}{N}], \left(\frac{1}{N}, \frac{2}{N}\right], \left(\frac{2}{N}, \frac{3}{N}\right], \ldots \left(\frac{N-1}{N}, \frac{N}{N}\right]\right)$. From each interval $\left(\frac{i-1}{N}, \frac{i}{N}\right]$, one value will be chosen randomly in its segment and its corresponding frame index on the x-axis will be picked out based on the curve. Considering that the x-axis value on the curve might not be an integer, we choose the integer closest to that value. Our sampling strategy is able to sample more frames during motion salient segments while sampling very small frames on static ones,



(a) a typical distribution of motion magnitude of Sth-Sth V1.



(b) a typical distribution of motion magnitude of Diving48.

Figure 6. Different datasets has different video time and action categories, yet motion-guided sampling method can guide the sampling with the cumulative distribution of motion magnitude.

thus allowing the subsequent video recognition models to focus on discriminative motion information learning. The sampled frames constitute a frame volume and will be fed into video CNNs to perform action recognition. In practice, we experiment with multiple network architectures and datasets Figure 6 to verify the effectiveness of our motion-guided sampler.

3.4. Discussion

We have noticed that there are several sampling methods proposed recently, such as SCSampler [17], DSN [51], Adaframe [47], Listen to Look [9] and AR-Net [28]. However, their focus is completely different from ours. Firstly, they aim at sampling a reduced set of clips from namely long and frequently sparse videos with a typical length of a minute or more, while our target is to choose a more effective input with a fixed length in trimmed videos. Secondly, some of the methods need extra input to train the sampler. SCSampler [17] and Listen to Look [9] use audio as an extra modality for exploiting the inherent semantic correlation between audio and the visual image. Thirdly, when training the sampler, Reinforcement Learning is commonly used where one agent or multiple agents are trained with policy gradient methods to select relevant video frames(Adaframe [47], DSN [51]). AR-net [28] contains a policy network with a lightweight feature extractor and an LSTM module. Both of the above training processes are complex and bring the network much extra computation.

In contrast to previous work, our proposed sampling strategy differs in three aspects. (1) The frame selection aims at selecting more effective frames with a fixed length in trimmed videos. (2) The sampling process doesn't need any extra input, making the input the same as the original. (3) MGSampler avoids complex training and is flexible enough to be inserted into other models.

4. Experiments

4.1. Datasets and Implementation Details

Datasets. We evaluate the motion-guided sampling strategy on five video datasets. These datasets can be grouped into two categories. (1) **Motion-related datasets**: Something-Something V1&V2 [10], Diving 48 [23], and Jester [27]. For these datasets, motion information rather than static appearance is the key to action understanding. In Something-Something V1&V2 [10], the same action is performed with different objects ("something") so that models are forced to understand the basic actions instead of recognizing the objects. It includes about 100K videos covering 174 classes. Jester [27] is a collection of labeled video clips that show humans performing hand gestures in front of a laptop camera or webcam, containing 148k videos and 27 classes. Diving48 [23] is designed to reduce the bias of scene and object context in action recognition. It has a fine-grained taxonomy covering 48 different types of diving with 18K videos in total. (2) **Scene-related datasets**: UCF101 [32] and HMDB51 [18]. Action recognition in these datasets can be greatly influenced by the scene context. UCF101 [32] consists of 13,320 manually labeled videos from 101 action categories. HMDB51 [18] is collected from various sources, e.g., web videos and movies, which proves to be realistic and challenging. It consists of 6,766 manually labeled clips from 51 categories.

Implementation details. In experiments, we use different models and backbones to verify the robustness of the motion-guided sampling strategy. Experiments are conducted on MMAction2 [3]. For fair comparison, all the settings are kept the same during training and testing. Taking Sth-Sth V1 dataset and TSM model as an example, we utilize 2D ResNet pre-trained on ImageNet dataset as the backbone. During training, random scaling and corner cropping are utilized for data augmentation, and the cropped region is resized to 224×224 for each frame. The batch size, initial learning rate, weight decay, and dropout rate are set to 64, 0.01, 5e-4, and 0.5 respectively. The networks are trained for 50 epochs using stochastic gradient descent (SGD), and the learning rate is decreased by a factor of 10 at 20 and 40 epochs. During testing, 1 clip with T frames is sampled from the video. Each frame is resized to 256×256 , and a central region of size 224×224 is cropped for action prediction. The implementation on other backbones and datasets is similar to this setting.

	μ	0	0.1	0.3	0.5	0.8	1	2
Image Diff	Sth V1	45.6	46.2	46.6	**47.1**	46.5	45.7	42.8
	Sth V2	57.9	58.2	59.7	**59.8**	59.4	58.0	56.2
Feature Diff	Sth V1	46.0	46.5	46.8	**47.3**	46.7	46.2	43.9
	Sth V2	58.2	58.5	60.0	**60.1**	59.8	58.3	56.4

Table 1. The effect of different values of μ on the results of Something-Something V1&V2. We use TSM model for the abalation study. Both train and test phase sample one clip of eight frames.

Dataset	Original	Image Diff	Feature Diff
Sth-V1	45.6	47.1(+1.5)	47.3(**+1.7**)
Sth-V2	57.9	59.8(+1.9)	60.1(**+2.2**)
Diving-48	35.2	36.9(+1.7)	37.4(**+2.2**)
UCF-101	94.5	94.9(+0.4)	95.2((**+0.7**)
HMDB-51	72.6	73.3(+0.7)	73.8(**+1.2**)
Jester	96.5	96.9(+0.4)	97.5(**+1.0**)

Table 2. **Performance of different motion representations.** The *original* means using TSN method to sample frames, which is the original sampling strategy in TSM. Noting that both UCF101 and HMDB51 have 3 splits, we report the average result on all splits.

4.2. Ablation Studies

Study on the smoothing hyperparameter. As shown in Figure 5, the smoothing hyperparameter μ controls the smoothness degree in our motion-guided sampling. When μ equals 1, the motion magnitude maintains the original one. If μ is greater than 1, it will increase the difference of motion magnitude between frames. On the contrary, when μ is set to less than 1, the influence of motion is decreased and particularly if μ is 0, the sampling process is equivalent to TSN [40] method. We perform the ablation study on hyperparameter μ and the results are reported in Table 1. We observe that $\mu = 0.5$ achieves the best result because it balances the relationship between the overall temporal structure and motion difference. We also observe that our motion-guided sampling is better than the baseline of TSN sampling (i.e., $\mu = 0$) by around 1.5% and 2% on Sth V1 and V2.

Study on different motion representations. We design two motion representations based on different levels. The image-level difference is a quite convenient way to capture motion replacement, but it ignores some important features and motion boundaries. On the other hand, feature-level difference can represent more precise motion cue while it needs a little bit more computation. Considering that our goal is to find an efficient way to represent motion, we only add one shallow convolutional layer to the original input yet it brings significant improvement. PAN [49] indicates that when the convolutional layer goes deeper, the performance based on feature-level difference degrades because high-level features have been highly abstracted and fail to reflect small motion replacement and boundaries.

To compare the performance based on the two motion representations, we conduct experiments on five different datasets, using TSM as the base model and inputting 8 frames. The result shows that regardless of the dataset, feature-level difference performs better than image-level difference mainly because differences in low-level features can capture small motion variations at boundaries.

Strategy	**Sth-V1**	**Sth-V2**
Segment based sampling	45.6	57.9
Fixed stride sampling	43.7	53.4
Motion magnitude sampling	41.5	52.8
Motion-guided sampling(Ours)	**47.3**	**60.1**

Table 3. Performance of different sampling strategies on the Something-Something V1 & V2 dataset.

Model	Backbone	Frames	TSN	MG Sampler
TSM [24]	ResNet50	8	45.6	47.1(**+1.5**)
TSM [24]	ResNet50	16	47.2	48.6(**+1.4**)
TSM [24]	ResNet101	8	46.9	47.8(**+0.9**)
TSM [24]	ResNet101	16	47.9	49.0(**+1.1**)
TEA [22]	ResNet50	8	48.9	50.2(**+1.3**)
TEA [22]	ResNet50	16	51.9	52.9(**+1.0**)
TEA [22]	ResNet101	8	49.4	50.6(**+1.2**)
TEA [22]	ResNet101	16	52.0	53.2(**+1.2**)
GSM [33]	BNInception	8	47.2	48.2(**+1.0**)
GSM [33]	BNInception	16	49.6	50.8(**+1.2**)
GSM [33]	InceptionV3	8	49.0	50.1(**+1.1**)
GSM [33]	InceptionV3	16	50.6	51.9(**+1.3**)

Table 4. Motion-guided sampling improves the accuracy for all different backbones and models, proving to be quite robust. In this ablation experiment, we use image-level difference as motion representation.

Comparison with different sampling strategies. To better illustrate the effectiveness of our proposed motion-guided sampling, we compare it with three other sampling methods. First, we compare with two fixed sampling baselines: (1) segment based sampling [40] where 8 frames are sampled uniformly along the temporal dimension and (2) fixed stride sampling [2] where an 8-frame clip with a fixed stride (s=4) is randomly picked from the video. We see that our adaptive sampling module is better than those hand-crafted sampling schemes. Then, we compare with another adaptive sampling method based on motion magnitude (motion magnitude sampling), where 8 frames selected merely based on motion magnitude regardless of motion uniform assumption. We see that this alternative motion-guided sampling strategy yields much worse performance, which confirms the effectiveness of our strategy based cumulative motion distribution.

Method	Backbone	Frames	Sth-Sth V1 Top-1 (%)	Sth-Sth V2 Top-1 (%)
I3D [2]	3D ResNet50	32×3×2	41.6	-
NL I3D [41]	3D ResNet50		44.4	-
ECO [54]	BNIncep+R18	8×1×1	39.6	-
ECO_{En} [54]		92×1×1	46.4	-
TSN [40]	BNInception	8×1×1	19.5	-
TSN [40]	ResNet50		19.7	27.8
TSM [24]	ResNet50	8×1×1	45.6	57.9
TSM [24]		16×1×1	47.2	59.9
GST [26]	ResNet50	8×1×1	47.0	61.6
GST [26]		16×1×1	48.6	62.6
TEINet [25]	ResNet50	8×1×1	47.4	61.3
TEINet [25]		16×1×1	49.9	62.1
GSM [33]	InceptionV3	8×1×1	49.0	-
GSM [33]		16×1×1	50.6	-
TDRL [43]	ResNet50	8×1×1	49.8	62.6
TDRL [43]		16×1×1	50.9	63.8
MVFNet [44]	ResNet50	8×1×1	48.8	60.8
MVFNet [44]		16×1×1	51.0	62.9
TEA [22]	ResNet50	8×1×1	48.9	60.9
TEA [22]		16×1×1	51.9	62.2
MG-TEA(Ours)	ResNet50	8×1×1	50.4	62.5
MG-TEA(Ours)		16×1×1	53.2	63.8
MG-TEA(Ours)	ResNet101	8×1×1	50.8	63.7
MG-TEA(Ours)		16×1×1	**53.3**	**64.8**

Table 5. **Comparison with other state-of-the-art methods on Something-Something V1&V2.** We use TEA model with our motion-guided sampling strategy(MG-TEA) for the comparison. We mainly compare with other methods with similar backbones under the 1-clip and center crop setting. "-" indicates the numbers are not available for us.

Varying backbones and models. We further demonstrate the robustness of our sampling strategy by varying the backbones and models. We choose ResNet 50 [11], ResNet 101 [11], BNInception [14], Inception V3 [35] for backbones and TSM [24], TEA [22], GSM [16] for models. Results on Table 4 indicate that motion-guided sampling is able to bring consistent performance improvement across different methods.

Efficiency and latency analysis. During training phase, we process the whole training set in advance by computing the difference. The 5^{th} row of Table 7 reports the total computing time of processing training data. For testing, we first report the inference time of the standard sampling strategy(TSN) for each video in 6^{th} row. Our MGSampler can slightly increase the inference time due to extra computation (7^{th} row), which is acceptable.

4.3. Comparison with the state of the art

We further report the performance of our motion-guided sampling on other datasets, including Diving48, UCF101, HMDB51, and Jester, and compare with the previous state-of-the-art methods. All the results are tested by one clip sampled from original video and reported in Table 2, Table 5 and Table 6. We see that our motion-guided sampling strategy is independent of the datasets and able to generalize well across datasets by bringing consistent performance improvement for different kinds of datasets with similar backbones under the single-clip and center-crop testing scheme.

Model	Frames	Top-1
TSN [40]	16	16.8
C3D [36]	64	27.6
R(2+1)D [37]	64	28.9
P3D-ResNet50 [29]	16	32.4
GST-ResNet50 [26]	16	38.8
TEA-ResNet50 [22]	16	36.0
GSM-InceptionV3[33]	16	39.0
MG-TEA-ResNet50(Ours)	16	**39.5**

Table 6. Performance on the Diving-48 dataset compared with the state-of-the-art methods. For fair comparison, all the models are tested by one clip.

	UCF101	HMBD51	Jester	Diving48	Sth-V2
Training Set	9537	3750	118562	15943	168913
Testing Set	3783	1530	14743	2096	24777
Average Frame	187.3	96.6	36.0	159.6	45.8
Training Time (all videos)	72.4s	23.2s	264.5s	388.7s	451.9s
TSN Sampling (each video)	6.5ms	4.7ms	3.2ms	6.8ms	4.4ms
MGSampler (each video)	6.9ms	5.0ms	3.5ms	7.4ms	5.0ms

Table 7. Running time and latency of MGSampler.

5. Conclusion

In this paper, we have presented a sparse, explainable, and adaptive sampling module for video action recognition, termed as MGSampler. Our new sampling module generally follows the assumption that motion is a universal and transferable prior information that enables us to design an effective frame selection scheme. Our motion-guided sampling shares two important ingredients: motion sensitive and motion uniform, where the former can help us identify the most salient segments against the background frames, and the latter enables our sampling to cover all these important frames with high motion salience. Experiments on five benchmarks verify the effectiveness of our adaptive sampling over these fixed sampling strategies, and also the generalization power of motion-guided sampling across different backbones, video models, and datasets.

Acknowledgements. This work is supported by National Natural Science Foundation of China (No. 62076119, No. 61921006), Program for Innovative Talents and Entrepreneur in Jiangsu Province, and Collaborative Innovation Center of Novel Software Technology and Industrialization. The first author would like to thank Ziteng Gao and Liwei Jin for their valuable suggestions.

References

[1] Hakan Bilen, Basura Fernando, Efstratios Gavves, and Andrea Vedaldi. Action recognition with dynamic image networks. *PAMI*, 40(12):2799–2813, 2017.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[3] MMAction Contributors. Openmmlab's next generation video understanding toolbox and benchmark, 2020.

[4] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015.

[5] Hehe Fan, Zhongwen Xu, Linchao Zhu, Chenggang Yan, Jianjun Ge, and Yi Yang. Watching a small portion could be as good as watching all: Towards efficient video classification. In *IJCAI*, 2018.

[6] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *CVPR*, pages 6016–6025, 2018.

[7] Christoph Feichtenhofer. X3D: Expanding Architectures for Efficient Video Recognition. In *CVPR*, pages 203–213, 2020.

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[9] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020.

[10] 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 *ICCV*, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[12] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[13] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, pages 2462–2470, 2017.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[15] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 2012.

[16] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. STM: SpatioTemporal and Motion Encoding for Action Recognition. In *ICCV*, 2019.

[17] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *ICCV*, 2019.

[18] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A Large Video Database for Human Motion Recognition. In *ICCV*, 2011.

[19] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *ECCV*, 2020.

[20] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion Feature Network: Fixed Motion Filter for Action Recognition. In *ECCV*, 2018.

[21] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion feature network: Fixed motion filter for action recognition. In *ECCV*, pages 387–403, 2018.

[22] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, 2020.

[23] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, pages 513–528, 2018.

[24] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. In *ICCV*, 2019.

[25] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. TEINet: Towards an Efficient Architecture for Video Recognition. In *AAAI*, 2020.

[26] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *ICCV*, 2019.

[27] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *ICCVW)*, pages 2874–2882, 2019.

[28] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. In *ECCV*, 2020.

[29] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

[30] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, pages 4161–4170, 2017.

[31] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *NeurIPS*, 27, 2014.

[32] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv preprint arXiv:1212.0402*, 2012.

[33] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *CVPR*, pages 1102–1111, 2020.

[34] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *CVPR*, pages 1390–1399, 2018.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[36] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[37] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[38] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term Temporal Convolutions for Action Recognition. *PAMI*, 2018.

[39] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. TDN: Temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021.

[40] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[41] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[42] Yifan Wang, Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Two-stream sr-cnns for action recognition in videos. In *BMVC*, 2016.

[43] Junwu Weng, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, Xudong Jiang, and Junsong Yuan. Temporal distinct representation learning for action recognition. In *ECCV*, pages 363–378, 2020.

[44] Wenhao Wu, Dongliang He, Tianwei Lin, Fu Li, Chuang Gan, and Errui Ding. Mvfnet: Multi-view fusion network for efficient video recognition. In *AAAI*, 2021.

[45] Wenhao Wu, Dongliang He, Xiao Tan, Shifeng Chen, and Shilei Wen. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *ICCV*, 2019.

[46] Zuxuan Wu, Caiming Xiong, Yu-Gang Jiang, and Larry S Davis. Liteeval: A coarse-to-fine framework for resource efficient video recognition. In *NeurIPS*, 2019.

[47] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *CVPR*, 2019.

[48] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond Short Snippets: Deep Networks for Video Classification. In *CVPR*, 2015.

[49] Can Zhang, Yuexian Zou, Guang Chen, and Lei Gan. Pan: Persistent appearance network with an efficient motion cue for fast action recognition. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 500–509, 2019.

[50] Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R. Scott, and Limin Wang. V4D:4D Convolutional Neural Networks for Video-level Representation Learning. In *ICLR*, 2020.

[51] Yin-Dong Zheng, Zhaoyang Liu, Tong Lu, and Limin Wang. Dynamic sampling networks for efficient action recognition in videos. *IEEE Transactions on Image Processing*, 29:7970–7983, 2020.

[52] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal Relational Reasoning in Videos. In *ECCV*, 2018.

[53] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *CVPR*, 2018.

[54] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: Efficient Convolutional Network for Online Video Understanding. In *ECCV*, 2018.