Efficient Action Recognition via Dynamic Knowledge Propagation

Hanul Kim[1,2], Mihir Jain[1], Jun-Tae Lee[1], Sungrack Yun[1], Fatih Porikli[1]
[1]Qualcomm AI Research*
[2]Seoul National University of Science and Technology
hukim@seoultech.ac.kr, {mijain, juntlee, sungrack, fporikli}@qti.qualcomm.com

Abstract

Efficient action recognition has become crucial to extend the success of action recognition to many real-world applications. Contrary to most existing methods, which mainly focus on selecting salient frames to reduce the computation cost, we focus more on making the most of the selected frames. To this end, we employ two networks of different capabilities that operate in tandem to efficiently recognize actions. Given a video, the lighter network processes more frames while the heavier one only processes a few. In order to enable the effective interaction between the two, we propose dynamic knowledge propagation based on a cross-attention mechanism. This is the main component of our framework that is essentially a student-teacher architecture, but as the teacher model continues to interact with the student model during inference, we call it a dynamic student-teacher framework. Through extensive experiments, we demonstrate the effectiveness of each component of our framework. Our method outperforms competing state-of-the-art methods on two video datasets: ActivityNet-v1.3 and Mini-Kinetics.

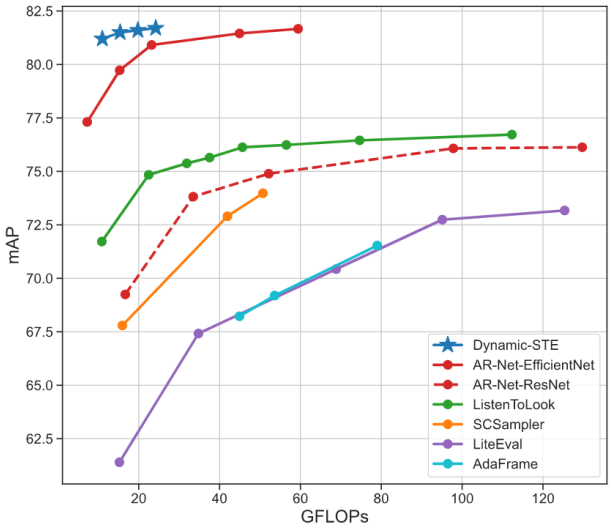


Figure 1: mAP vs. GFLOPs curves on ActivityNet-v1.3 [5]: The proposed dynamic student-teacher framework performs similar to or better than the recent state-of-the-art methods [11, 22, 26, 44, 45] at a much lower computational cost. More experimental results are available in Section 4.

1. Introduction

Video action recognition is one of the fundamental problems for video understanding. Consequently, several methods have been proposed and significant progress has been made over the last decade thanks to advances in deep learning. Recent state-of-the-art methods mostly use 3D-CNN that takes as input a video clip of several frames [9, 13]. Many of these methods densely sample clips from each video and aggregate the activations to achieve excellent results. However, they require high computational costs, making it challenging to apply to practical applications. In this paper, we are interested in the problem of efficient video recognition, to achieve better performance and computational cost trade-off as shown in Figure 1.

Multiple approaches have been proposed for efficient action recognition in recent years with focus on two aspects: (a) more efficient CNNs and (b) salient frame/clip selection. On the first aspect, methods have been proposed to design efficient versions of 3D-CNN [21, 29, 35, 36, 46], such as the temporally separable convolution that reduces the computational cost per clip. But the more successful way has been to simply switch to the lighter 2D-CNNs instead [8, 24], often in conjunction with RNN/LSTM models [3, 11, 44]. Even with more efficient networks, computation would be high for longer videos if all the frames are processed. So, the second aspect of frame selection based on saliency complements the first one and has led to most success too [3, 11, 22, 44]. These methods rely on learning a policy that decides how a particular frame should be pro-

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

cessed/skipped [3, 11] and at what resolution [26, 30]. Such policy functions certainly boost efficiency, however these methods rely on the policy not to miss frames that are crucial for action recognition.

In this paper, we focus on making the most of the selected frames by minimizing the performance loss due to switching to a lighter network while keeping the efficiency high. To this end, we propose a knowledge propagation framework with two networks, one light and the other relatively heavy, that operate in tandem to efficiently recognize actions. The interaction between the two networks is not just limited to the training phase, but it is used for inference also, dynamically adjusting to each test video, hence we call this method *Dynamic Knowledge Propagation*. We take inspiration from the self-attention technique [38] that describes a mapping between a query and key-value pairs. Ours is a cross-attention mechanism instead, where the query comes from the light network and key-value pairs come from the heavy network. These two networks are used as student and teacher models in our framework, which we call *Dynamic Student-Teacher* architecture, as the student and teacher interact dynamically for each test video. The proposed framework lets the student process most of the sampled frames, while the teacher processes only a few of them. Then, with the dynamic knowledge propagation the student features are enhanced by the better quality features from the teacher model.

Our main contribution is the dynamic knowledge propagation mechanism. This enables interaction between the student and teacher model both during training and inference and is the key component of the proposed framework. Our second contribution is the dynamic student-teacher framework for efficient video action recognition. The model combines light and heavy networks to reduce computational costs without significant performance degradation using dynamic knowledge propagation. Finally, through extensive experiments we demonstrate the effectiveness of each component of the proposed approach. We evaluate our method on two popular benchmarks, ActivityNet-v1.3 [5] and Mini-Kinetics [4], and improve state-of-the-art on both of them with $1.4\times$ and $2.5\times$ less GFLOPs, respectively.

2. Related Work

2.1. Efficient Video Recognition

There are two streamlines of deep learning approaches for efficient action recognition. The first focuses on designing network architecture to efficiently obtain the spatio-temporal video representation. For example, [29, 36, 46] factorized the 3D convolution filters to 2D spatial and 1D temporal ones, and [21, 35] modified efficient 2D modules to their 3D counterparts. [24] simply shifted part of features temporally, and [7] extended it to adaptive spatio-temporal shifting. SlowFast [10] are related to ours in that both methods use two-branch architectures. However, their objectives and structural details are quite different. SlowFast intends to capture different semantics, while our goal is efficiency. Furthermore, SlowFast fuses two branches with several lateral structures. Contrarily, the proposed dynamic knowledge propagation module makes two branches interact dynamically, attaining both efficiency and accuracy.

The second streamline attempts to select salient frames (or video clips) to alleviate the computational cost. The advantage of this approach is that it can be applied model-agnostically. Reinforcement learning is utilized in [6,43,45, 47, 48] to train agents or policies to decide the next frame to sample. Campos *et al*. [3] proposed Skip RNN model that learns to skip state update to decrease the number of sequential operations of RNN. Korbar *et al*. [22] designed a ranking loss to learn a video clip sampler to mimic an oracle sampler. To adaptively extract coarse or fine features, Wu *et al*. [44] used a binary gate, and Meng *et al*. [26] developed a policy network which does even skipping. Quader *et al*. [30] ensembled multiple networks with different spatio-temporal granularity of inputs, invoking the finer network when the previous coarse network failed. The closest to ours is the method in [11] that trains a light Image+Audio student with a 3D-CNN teacher. They also use a separate attention-based LSTM network as a sampler with some additional computational cost. The proposed dynamic knowledge propagation also exploits student-teacher framework, but unlike [11] it is a dynamic architecture with teacher active during inference also.

2.2. Cross-Attention

Attention mechanisms have yielded significant improvement in many tasks including image classification [17], object detection [41], image captioning [18], and machine translation [25, 38]. Recently, to leverage relationship between two heterogeneous representation, the diverse cross-attention schemes are devised. Several works exploited the cross correlation between the heterogeneous representation as the attention weights for image and sentence matching in visual question answering (VQA) [19, 23], and for query and prototype matching in prototypical few-shot learning [16]. Inspired by the transformer [38] which conducts self-attention using the scaled dot-product of key, query, and value, [40] applied the scaled dot-product operation for the concatenation of image and text features for VQA. Also, [37] attended query from one modality using key and value of another modality for multi-modal sentiment analysis. In this work, we exploit the scaled dot-product operation to propagate the knowledge of a heavy network to a light one.

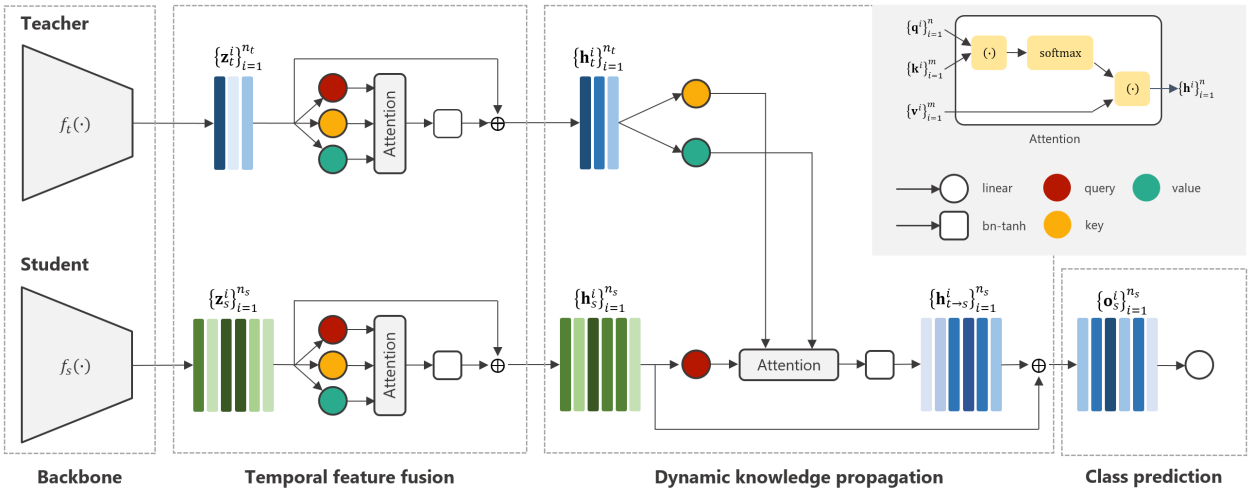


Figure 2: The dynamic student-teacher model for efficient video recognition. Our framework has three stages. In the first, feature sequences from the backbones are temporally enhanced by self-attention. Then, these enhanced feature sequences go through the proposed dynamic knowledge propagation mechanism. Finally, the knowledge propagated features are combined with student self-attended features via skip connection and used for action class prediction.

2.3. Knowledge Distillation

Recent studies have shown that the knowledge learned from a teacher network can be used to improve the performance of a student network [2, 15, 27]. In the literature, the teacher network often refers to a heavy and cumbersome model, whereas the student network a simple and lightweight model. Both the teacher and student networks address the same task. In the action recognition tasks, the knowledge distillation framework is employed for student networks to leverage the high-level knowledge of teacher networks, such as depth [12] and temporal information [32]. [42] employed multiple different teacher networks. To reduce computational cost, [11, 28] used the teacher with high-resolution inputs to leverage the student using low-resolution inputs, and [39] exploited the teacher observing the entire videos and student only seeing partial videos. Also, distilling knowledge from the teacher taking high-quality skeleton as inputs, [1] learned the student with low-quality skeleton inputs. Contrary to the existing works, the proposed method propagates the teacher's knowledge to the lightweight student also during the inference. With this, we can effectively propagate the rich teacher's knowledge to the student using only a few sampled features.

3. Proposed Method

In this section, we first describe our dynamic knowledge propagation. It is a cross-attention mechanism, between two sequences, which relates different positions of one sequence to the other in order to propagate knowledge dynamically. Next, we present our dynamic student-teacher framework, that employs the dynamic knowledge propagation in a student-teacher architecture for the efficient video action classification. The dynamic student-teacher model consists of light and heavy networks. The light network (*student*) produces features for all the sampled frames very efficiently, and the heavy network (*teacher*) extracts high-quality features for a much smaller set of sampled frames. Then, the dynamic knowledge propagation enriches the entire student features using a handful of the teacher features.

3.1. Dynamic Knowledge Propagation

We consider two networks f_s and f_t, both trained for the same task but have different capabilities (f_t is deeper than f_s). Then, we sample two subsets of frames from a video, one with n_s frames and the other containing n_t frames. The first subset is processed by the network f_s to extract feature sequence, $\{\mathbf{h}_s^i\}_{i=1}^{n_s}$, while the second one is processed by the network f_t to obtain feature sequence, $\{\mathbf{h}_t^i\}_{i=1}^{n_t}$. Here, $\mathbf{h}_s^i \in \mathbb{R}^{d_s}$ and $\mathbf{h}_t^i \in \mathbb{R}^{d_t}$.

Now, using $\{\mathbf{h}_s^i\}_{i=1}^{n_s}$ and $\{\mathbf{h}_t^i\}_{i=1}^{n_t}$, we propagate the richer knowledge of f_t to f_s. To this end, inspired by the self-attention mechanism [38], we develop the cross-attention mechanism between the feature sequences of the two models. Specifically, the feature sequences $\{\mathbf{h}_s^i\}_{i=1}^{n_s}$ and $\{\mathbf{h}_t^i\}_{i=1}^{n_t}$ are first projected to queries $\{\mathbf{q}_s^i\}_{i=1}^{n_s}$ and key-value pairs $\{(\mathbf{k}_t^i, \mathbf{v}_t^i)\}_{i=1}^{n_t}$, respectively. Here, $\mathbf{q}_s^i \in \mathbb{R}^{d_k}$, $\mathbf{k}_t^i \in \mathbb{R}^{d_k}$ and $\mathbf{v}_t^i \in \mathbb{R}^{d_s}$. Note that, unlike the self-attention, the proposed cross-attention takes queries and key-value pairs from different networks for the knowledge propagation between them. Then, the cross-attended feature $\mathbf{h}_{t\to s}^i$

for $\mathbf{h}_s^i$ is generated by

$$\mathbf{h}_{t\to s}^i = \sum_u \frac{\exp(\mathbf{q}_s^i \cdot \mathbf{k}_t^u/\tau)}{\sum_r \exp(\mathbf{q}_s^i \cdot \mathbf{k}_t^r/\tau)} \cdot \mathbf{v}_t^u \tag{1}$$

where the temperature [15] τ is set to the square root of key's dimensionality in order to scale the dot-product of the query and the key. Thus, the low-quality feature is replaced with the weighted sum of the high-quality ones, where the attention weights are determined by scaled dot-product similarities between queries and keys. Also, note that the longer feature sequence of f_s are dynamically improved using a small number of features of f_t.

3.2. Dynamic Student-Teacher Framework

Figure 2 shows the overall architecture of the proposed dynamic student-teacher model. To apply the dynamic knowledge propagation to the student-teacher model, we set backbone networks f_s and f_t as light student and heavy teacher networks, respectively. These backbone networks extract the frame-level features. The proposed dynamic knowledge propagation utilizes both teacher and student networks during testing phase. For efficiency, we set n_t much less than n_s, so the deeper teacher network extract better quality features from a much smaller number of frames. Then, for each backbone, the frame-level features, $\{\mathbf{z}_s^i\}_{i=1}^{n_s}$ and $\{\mathbf{z}_t^i\}_{i=1}^{n_t}$, are fed into the temporal feature fusion module. On top of the temporal feature fusion modules, the dynamic knowledge propagation module transfers the knowledge of the teacher network to the student. Then, the class prediction module outputs video-level action class. Our framework is different from usual student-teacher network as during inference student and teacher act as an ensemble and interact dynamically for each test video. Therefore, to differentiate, we refer to it as *Dynamic Student-Teacher Ensemble* (Dynamic-STE). Next, we discuss each of these modules in detail.

Sampling scheme: As shown in Figure 3, a video can be divided to T short clips, V^t where $t = 1, \ldots, T$. Since a clip includes visually similar (almost identical) frames, we summarize each clip with a representative frame which is simply the first frame in a clip. To this end, in testing phase, we uniformly sample frames with different sampling intervals r_s and r_t for student and teacher, respectively. To reduce the computation cost in the heavy teacher network, we set $r_t > r_s$. Then, we sample $n_s = \lfloor T/r_s \rfloor$ frames for the student and $n_t = \lfloor T/r_t \rfloor$ frames for the teacher. Also, as illustrated in Figure 3, to avoid redundant sampling for the student and the teacher, we skip the sampled student frame, if it is in the same clip of a sampled teacher frame. However, during training, to exploit the relationship between matched features of student and teacher networks, we set r_t equal to r_s allowing redundant sampling.

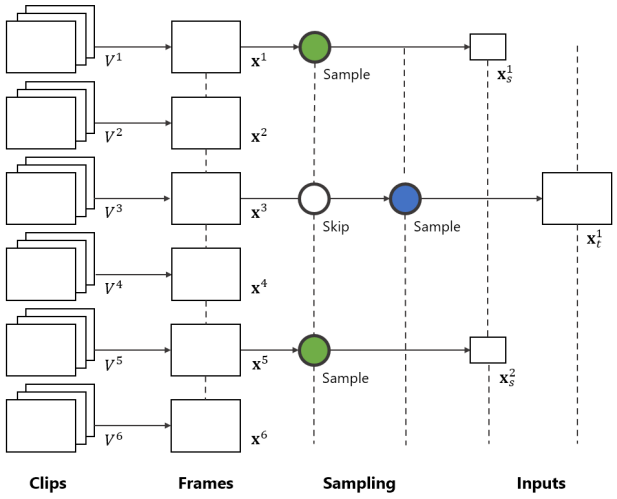


Figure 3: Sampling scheme at inference time (example for a 6-clip video). From each clip V^t, the first frame $\mathbf{x}^t$ is selected as a representative. The student (green) and teacher (blue) uniformly sample the frames with different sampling intervals, avoiding overlapped sampling. Then, for student, the spatial resolution of the frame is reduced to a fourth.

Backbone network: The backbone network extracts features from input clips (a frame per clip). According to the role, the student employs a light backbone (*e.g.* EfficientNet-B0 [33]) taking a low spatial resolution (112×112) frames as inputs to quickly process all n_s frames. On the other hand, the teacher uses a relatively heavier backbone (*e.g.* EfficientNet-B3 [33]) taking high spatial resolution (224×224) inputs for more accurate prediction at an additional computation cost.

Temporal feature fusion: As an action instance is captured by a temporal sequence of the several frames, it is essential to aggregate neighboring frames over time for accurate video action recognition. Therefore, for the frame-level features of each backbone, we conduct the temporal feature fusion using the self-attention mechanism [38]. First, three linear projection layers generate query, key, and value for each frame-level feature. Then, the scaled dot-product operation of the queries, keys, and values produces the temporally fused feature $\{\mathbf{h}^i\}_{i=1}^{n}$ where n is n_s and n_t for the student and teacher, respectively.

Dynamic knowledge propagation: As described in Section 3.1, the dynamic knowledge propagation transfers the rich knowledge for actions from the teacher feature sequence to the student feature sequence. In the dynamic knowledge propagation, the student features $\{\mathbf{h}_s^i\}_{i=1}^{n_s}$ are transformed to queries $\{\mathbf{q}_s^i\}_{i=1}^{n_s}$ using a linear layer. Sim-

ilarly, the teacher features $\{\mathbf{h}_t^i\}_{i=1}^{n_t}$ are linearly projected to keys $\{\mathbf{k}_t^i\}_{i=1}^{n_t}$ and values $\{\mathbf{v}_t^i\}_{i=1}^{n_t}$. Then, the knowledge propagated student features, $\mathbf{h}_{t\to s}^i$, are obtained using the cross-attention in Equation (1). Here, the student knowledge over larger number of frames is fused with the higher quality teacher features. Then, through a residual connection the knowledge of the student is combined with $\mathbf{h}_{t\to s}^i$ as follows:

$$\mathbf{o}_s^i = \mathbf{h}_s^i + \mathbf{h}_{t\to s}^i. \tag{2}$$

The residual connection has the effect of combining the knowledge of the student and teacher networks. Note that the first term $\mathbf{h}_s^i$ is generated by the student network only, and the second term $\mathbf{h}_{t\to s}^i$ is the dynamically modified student feature using the teacher's knowledge.

Class prediction: The class prediction is performed by a simple linear classifier. Given $\{\mathbf{o}_s^i\}_{i=1}^{n_s}$, the network computes the score s_j^i indicating the confidence that the ith frame belongs to the jth action class. We then select the k frames with the highest maximum confidence scores, where k is proportional to the number of sampled student frames n_s as

$$k = \max(1, \lfloor \frac{n_s}{\gamma} \rfloor) \tag{3}$$

where γ is a hyperparameter. Then, the dynamic student-teacher model averages their scores for each class, and outputs the video action class with the maximum average score.

3.3. Loss Functions

Here, we describe the loss functions to train the dynamic student-teacher model. We learn the student and teacher backbones independently. Hence, we first optimize the video classification loss $\mathcal{L}_{\text{vid}}$ to train the teacher network. Then, we train the student network by minimizing the sum of the three losses: video classification loss $\mathcal{L}_{\text{vid}}$, frame classification loss $\mathcal{L}_{\text{frm}}$, and cosine similarity loss $\mathcal{L}_{\text{cos}}$. Formally, the loss functions of the teacher and student, $\mathcal{L}_t$ and $\mathcal{L}_s$, are represented by

$$\mathcal{L}_t = \mathcal{L}_{\text{vid}} \tag{4}$$

$$\mathcal{L}_s = \mathcal{L}_{\text{vid}} + \lambda_{\text{frm}}\mathcal{L}_{\text{frm}} + \lambda_{\text{cos}}\mathcal{L}_{\text{cos}} \tag{5}$$

where λ_{frm} and λ_{cos} are hyperparameters to control the contribution of frame classification and cosine similiarity losses. Let us describe each loss subsequently.

Video classification loss: Video classification loss penalizes the prediction errors of the student (or teacher) network, which estimates the softmax probabilities of action classes. Given an input video $\mathcal{V}$ and the ground-truth one-hot vector $\mathbf{y}$, the video classification loss is defined by

$$\mathcal{L}_{\text{vid}}(\mathcal{V}; s) = \text{CE}(\hat{\mathbf{y}}, \mathbf{y}) \tag{6}$$

where CE denotes the cross entropy loss function, and $\hat{\mathbf{y}}$ means the softmax probabilities obtained from the confidences scores of the student (or teacher) network.

Frame classification loss: Since we address the weakly supervised action recognition, the frame-level ground-truth labels are not available. Instead, we use the prediction of the teacher network as the pseudo label for the frame-level prediction of the student network. Consequently, we encourage the student's prediction to be similar with the teacher's prediction which is more accurate. Specifically, we define the frame classification loss by

$$\mathcal{L}_{\text{frm}}(\mathcal{V}; s, t) = \frac{1}{n_s}\sum_{i=1}^{n_s} \text{CE}(\hat{\mathbf{y}}_s^i, \hat{\mathbf{y}}_t^i) \tag{7}$$

where, for the ith frame, $\hat{\mathbf{y}}_s^i$ and $\hat{\mathbf{y}}_t^i$ are the softmax probabilities computed by the student and teacher networks, respectively.

Cosine similarity loss: The attention technique [38] replaces queries with values based on the scaled dot-product similarities between queries and keys. In our cross-attention, queries and keys come from student and teacher networks, respectively. Therefore, for each video frame, it is beneficial to make student and teacher networks produce similar features for stable knowledge propagation. To this end, in this loss, we maximize cosine similarities between the query and the key, by

$$\mathcal{L}_{\text{cos}}(\mathcal{V}; s, t) = -\frac{1}{n_s}\sum_{i=1}^{n_s} \frac{\mathbf{q}_s^i \cdot \mathbf{k}_t^i}{\|\mathbf{q}_s^i\|\|\mathbf{k}_t^i\|} \tag{8}$$

where $\mathbf{q}_i$ and $\mathbf{k}_i$ denote the query and key corresponding to the ith frame.

4. Experiments

In this section, we provide experimental analysis and comparative evaluation to show the effectiveness of the proposed method.

4.1. Experimental Setup

Datasets: We experiment on two large-scale datasets: ActivityNet 1.3 [5] and Mini-Kinetics [4]. ActivityNet 1.3 provides samples from 200 action classes with 10,024 videos for training and 4,926 videos for validation. The average action instance per video is 1.41, and the average duration of the videos is 167 seconds. Mini-Kinetics includes 121,215 training and 9,867 testing videos, each containing one of 200 action classes. The average length of videos is 10 seconds.

Table 1: Comparison with state-of-the-arts in efficient action recognition on ActivityNet 1.3 [5].

Method	Backbone	mAP (%)	GFLOPs
AdaFrame [45]	ResNet	71.6	79.0
LiteEval [44]	ResNet	72.7	95.1
ListenToLook [11]	ResNet	75.6	37.5
SCSampler [22]	ResNet	72.9	41.9
AR-Net [26]	ResNet	73.8	33.5
Ours	ResNet	75.9	30.5
AR-Net [26]	EfficientNet	79.7	15.3
Ours	EfficientNet	81.2	11.0

Table 2: Comparison with state-of-the-arts in efficient action recognition on Mini-kinectics [4].

Method	Backbone	Top1 (%)	GFLOPs
LiteEval [44]	ResNet	61.0	99.0
SCSampler [22]	ResNet	70.8	41.95
AR-Net [26]	ResNet	71.7	32.0
Ours	ResNet	72.7	18.3
AR-Net [26]	EfficientNet	74.8	16.32
Ours	EfficientNet	75.2	6.6

Evaluation metrics: Following the literature [26], we measure the mean average precision (mAP) and top-1 accuracy to evaluate the performance on ActivityNet 1.3 and Mini-Kinetics, respectiviely. We measure the mean average precision (mAP) to evaluate the performance for action recognition accuracy. We evaluate the computation cost as giga floating point operations (GFLOPs). The computational cost highly depends on backbone networks, thus the contribution of the other parts can be ignored for efficiency evaluation. More specifically, the total computational cost of the proposed method is given by $n_s \times b_s + n_t \times b_t$, where b_s and b_t are the computational costs for student and teacher's backbone networks.

As a reference, the proposed student and teacher networks have 0.1 and 1.8 GFLOPs respectively for a single input frame, when they employ EfficientNet backbones. In case of ResNet-based backbones, they have 0.5 and 4.1 GFLOPs per frame. Note that the student uses low-resolution inputs 112×112 to decrease computational cost, while the teacher takes high-resolution inputs 224×224 as its inputs. Different baseline methods have different sampling intervals r_s and r_t for action recognition, so we report average GFLOPs per video for all the experiments.

Implementation details: We use ResNet-18 [14] or EfficientNet-B0 [33] for the student's backbone network. And we employ ResNet-50 [14] or EfficientNet-B3 [33] for the teacher's backbone network. All backbones are pre-trained on ImageNet [31]. Here, we remove the last classification layers from the backbone networks. We set d_k to 256 for the temporal feature fusion and the dynamic knowledge propagation networks. In Equation (3), we set γ to 4. The hyperparameters λ_f and λ_c are fixed to 0.5.

For experiments, we set the frame rates of input video to 16 FPS and split it into clips. Each of which has 16 frames. As we described in Section 3.2, we regard each clip as a single frame. Here, we simply pick the first frame for each sampled clip. We adopt the fixed number of sampled frames to train and to test the proposed method. Specifically, we adaptively adjust sampling intervals r_s and r_t depending on the length of input video, to satisfy n_s and n_t for a desired GFLOPs of computation.

The proposed student-teacher model is trained in two stages: First, we train the teacher network by minimizing Equation (4). We then train the student via Equation (5). Note that the parameters of teacher is fixed in the second stage. The training is iterated for 40 epochs in both stages. The Adam optimizer [20] is employed with an initial learning rate of 1.0×10^{-4}. We decay the learning rate by a factor of 0.1 after 15, and 30 epochs.

4.2. Comparison on AcitivtyNet 1.3

We compare the proposed method with recent state-of-the-art methods on the ActivityNet 1.3 dataset: AdaFrame [45], LiteEval [44], SCSampler [22], ListenToLook [11], and AR-Net [26]. For the comparison, we use the performances of existing methods reported in their original papers. Comparison is done for the mAP (%) and GFLOPs used. In Table 1, the dynamic student-teacher ensemble outperforms all other methods regardless of their backbone networks. Further, in Figure 1, Dynamic-STE achieves the better efficiency-accuracy trade-off than all other methods. The dynamic student-teacher model achieves higher performance than the AR-Net with EfficientNet, a modern architecture to provide more accurate feature representations with less computational cost, by 0.5% in mAP with 28.1% less computation (15.3 GFLOPs vs. 11.0 GFLOPs). Therefore, we can conclude that the proposed method precisely recognizes actions in videos with high efficiency by simultaneously employing light and heavy networks.

4.3. Comparison on Mini-Kinetics

For extensive experiments, we also evaluate the performance of the proposed method on Mini-Kinetics. Table 2 shows the top-1 accuracy score and GFLOPs of the proposed method and those of existing methods [22, 26, 44]. In Table 2, we report the performance of the proposed model with $n_s = 12$ and $n_t = 3$. Note that the average video length of Mini-Kinetics is shorter than that of Acitiv-

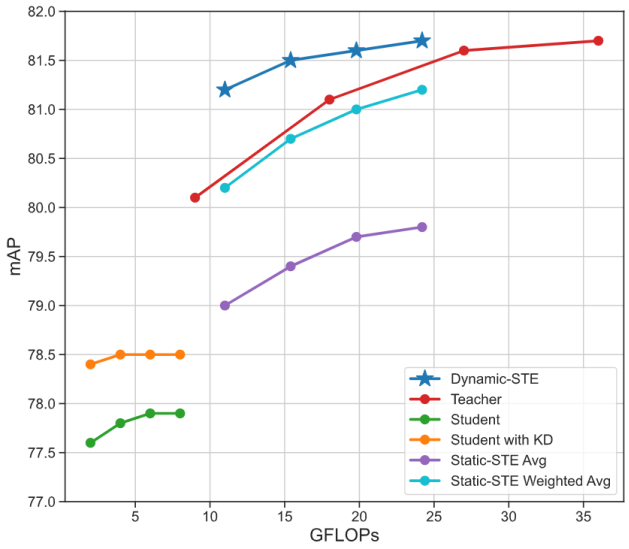


Figure 4: Accuracy vs. efficiency curves of the proposed model and those of its baselines on ActivityNet 1.3 [5]: We set n_t from 5 to 20 for Teacher and n_s from 20 to 80 for Student and Student with KD. Also, for the other models, we use n_t ranging from 5 to 11 and fix the ratio n_s/n_t to 4.

ityNet1.3. So, the negative impact of sampling is less significant in Mini-Kinetics. We see that the dynamic student-teacher model again outperforms EfficientNet-based AR-Net, which is the current state-of-the-art method, with only 40% GFLOPs of the AR-Net.

4.4. Ablation Study

Knowledge distillation baselines: We first define the knowledge distillation related baselines that are used in this sub-section. Student and Teacher denote models consisting of a single student and teacher network, respectively. The loss function in Equation (4) is used to train these models. Student with KD indicates the student network, which is trained with knowledge distillation by minimizing Equation (5), of course without cosine similarity loss, $\mathcal{L}_{\cos}$.

Static Student-Teacher Ensemble: We argue for the idea of using teacher model also during inference in order to let it assist the student model with minimal added computational cost. In this ablation, we evaluate a static set-up of student-teacher ensemble at the inference. In this set-up, all the interaction between the two models happens during training only, through knowledge distillation. During inference, the two models have no influence on each other and the video-level predictions of both networks are simply averaged to arrive at the final classification scores, hence it is static. The static student-teacher ensembles are referred as Static-STE Avg and Static-STE Weighted Avg. The former

Table 3: Impact of dynamic knowledge propagation.

Model	Frames per video	mAP (%)	GFLOPs
Teacher	20	81.7	36.0
Teacher	5	80.1	9.0
Student	20	77.6	2.0
Student with KD	20	78.4	2.0
Static-STE Avg	25	79.0	11.0
Static-STE Weighted Avg	25	80.2	11.0
Dynamic-STE	25	81.2	11.0

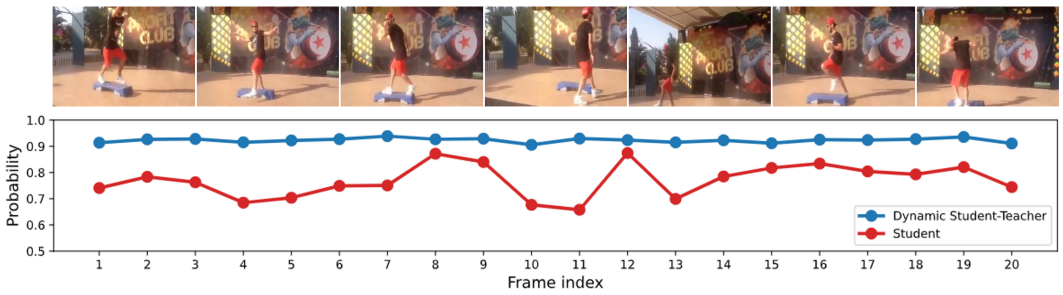
simply averages the video-level predictions of Teacher and Student with KD, whereas for Static-STE Weighted Avg, we aggregate these predictions by using the weighted average fusion layer [34].

In Table 3, we compare the static student-teacher ensembles with baselines for mAP and GFLOPs on AcitivtyNet 1.3. Static-STE Avg achieves better accuracy than all the baselines except for Teacher, which uses 20 frames per video and 3.5 times the GFlops. The mAP comparison over varying computation is shown in Figure 4. Again, Static-STE Avg provides better trade-off between accuracy and efficiency than all, while almost matches Teacher in the lower GFLOPs range. We conclude that even the static version of Student-Teacher pair compares favourably to most Knowledge distillation baselines.

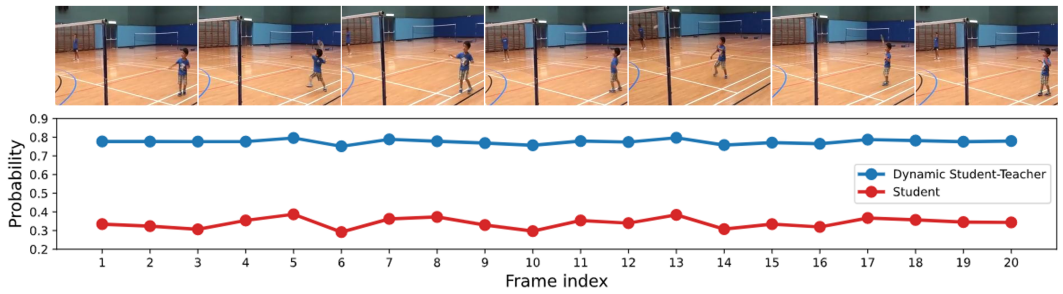
Dynamic Student-Teacher Ensemble: The dynamic student-teacher ensemble, Dynamic-STE, with the proposed dynamic knowledge propagation clearly performs better than the knowledge distillation baselines, achieving the best trade-off between accuracy and efficiency in Figure 4. Next, we compare it with the static student-teacher ensembles, which is a more comparable baseline. The proposed Dynamic-STE leads the Static-STE models with higher mAP across the varying GFLOPs. This shows the advantage of dynamic knowledge propagation over the naïve or weighted fusion. We conclude that pairing teacher model with the student model during inference is effective, and with the dynamic knowledge propagation it leads to the new state-of-the-art for efficient video recognition and promising results for future research.

Figure 5 shows qualitative results and the frame-level probabilities indicating that a frame belongs to the ground-truth action class. In these examples, we see that the proposed method provides more accurate frame-level predictions through the dynamic knowledge propagation to convey the teacher's knowledge to the student during the inference time.

Temporal feature fusion: Table 4 shows the impact of temporal feature fusion network on performance of the pro-



(a) Doing Step Aerobics



(b) Playing Badminton

Figure 5: Action recognition on two videos: (a) Doing Step Aerobics and (b) Playing Badminton. In both examples, the first row illustrates input frames, and the second row shows the sequence of probabilities for ground-truth class predicted by Student and Dynamic Student-Teacher Ensemble.

Table 4: Impact of temporal feature fusion.

Student	Teacher	mAP (%)	GFLOPs
w/o fusion	w/o fusion	79.3	11.0
w/o fusion	with fusion	80.7	11.0
with fusion	w/o fusion	79.9	11.0
with fusion	with fusion	81.2	11.0

Table 5: Impact of loss functions.

Loss functions	mAP (%)	GFLOPs
$\mathcal{L}_{\text{vid}}$	80.8	11.0
$\mathcal{L}_{\text{vid}} + \lambda_{\text{frm}}\mathcal{L}_{\text{frm}}$	81.0	11.0
$\mathcal{L}_{\text{vid}} + \lambda_{\text{cos}}\mathcal{L}_{\text{cos}}$	80.9	11.0
$\mathcal{L}_{\text{vid}} + \lambda_{\text{frm}}\mathcal{L}_{\text{frm}} + \lambda_{\text{cos}}\mathcal{L}_{\text{cos}}$	81.2	11.0

posed method on ActivityNet1.3 dataset. Absence of temporal feature fusion network for either student or teacher degrades the mAP, showing its importance for extracting more better features. Especially, the network in the teacher enables to convey temporally aggregated knowledge to the student in the dynamic knowledge propagation network, and thus improves action recognition accuracy. The best performance is achieved when both student and teacher networks employ the temporal feature fusion module.

Loss functions: Table 5 reports action classification mAP scores on ActivityNet1.3, depending on the combinations of loss functions. In this table, we observe that loss functions λ_{frm} and λ_{cos} for on knowledge distillation slightly improves mAP scores. And we obtain the best performance by combining all three types of losses. Note that in all the cases in Table 5 dynamic knowledge propagation is used even when $\mathcal{L}_{\text{cos}}$ is not used during training.

5. Conclusion

We propose a novel dynamic knowledge propagation framework with teacher and student networks that operate in tandem for efficient action recognition, also interacting during inference. Student network processes the majority of the sampled frames leaving few for teacher network. The proposed knowledge propagation effectively combines the knowledge from a handful of better quality features from teacher model with the larger number of features from student model. Extensive experiments demonstrate the effectiveness of each component of the proposed framework. Experiments also show that strategically employing teacher model also at the inference is effective for efficient video recognition. Our method exceeds the state-of-the-art methods on two video datasets, ActivityNet-v1.3 and Mini-Kinetics, while using about $1.4\times$ and $2.5\times$ less GFLOPs, respectively.

Acknowledgement. We thank Ilia Karmanov for his help in improving the writing.

References

[1] Cunling Bian, Wei Feng, Liang Wan, and Song Wang. Structural knowledge distillation for efficient skeleton-based action recognition. *IEEE Transactions on Image Processing*, 30:2963–2976, 2021. 3

[2] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proc. ACM SIGKDD*, 2006. 3

[3] Víctor Campos, Brendan Jou, Xavier Giró-i Nieto, Jordi Torres, and Shih-Fu Chang. Skip RNN: Learning to skip state updates in recurrent neural networks. In *Proc. ICLR*, 2017. 1, 2

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. IEEE CVPR*, 2017. 2, 5, 6

[5] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proc. IEEE CVPR*, 2015. 1, 2, 5, 6, 7

[6] 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. 2

[7] Linxi Fan, Shyamal Buch, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. RubiksNet: Learnable 3D-shift for efficient video action recognition. In *Proc. ECCV*, 2020. 2

[8] Quanfu Fan, Chun-Fu (Richard) Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. In *NeurIPS*, 2019. 1

[9] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 1

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proc. IEEE ICCV*, 2019. 2

[11] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *Proc. IEEE CVPR*, 2020. 1, 2, 3, 6

[12] Nuno C Garcia, Pietro Morerio, and Vittorio Murino. Modality distillation with multiple stream networks for action recognition. In *Proc. ECCV*, 2018. 3

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, 2016. 6

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Proc. NeurIPS Workshop*, 2015. 3, 4

[16] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Proc. NeurIPS*, 2019. 2

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proc. IEEE CVPR*, 2018. 2

[18] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proc. IEEE ICCV*, 2019. 2

[19] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. In *Proc. ICLR*, 2017. 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2014. 6

[21] Okan Kopuklu, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. Resource efficient 3D convolutional neural networks. In *Proc. ICCV Workshop*, 2019. 1, 2

[22] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proc. IEEE ICCV*, 2019. 1, 2, 6

[23] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proc. ECCV*, 2018. 2

[24] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *Proc. IEEE ICCV*, 2019. 1, 2

[25] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*, 2015. 2

[26] 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 *Proc. ECCV*, 2020. 1, 2, 6

[27] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proc. IEEE CVPR*, 2019. 3

[28] Didik Purwanto, Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang. Extreme low resolution action recognition with spatial-temporal multi-head self-attention and knowledge distillation. In *Proc. CVPR Workshop*, 2019. 3

[29] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proc. IEEE ICCV*, 2017. 1, 2

[30] Niamul Quader, Juwei Lu, Peng Dai, and Wei Li. Towards efficient coarse-to-fine networks for action and gesture recognition. In *Proc. ECCV*, 2020. 2

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 6

[32] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3D: Distilled 3D networks for video action recognition. In *Proc. WACV*, 2020. 3

[33] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. ICML*, 2019. 4, 6

[34] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proc. IEEE CVPR*, 2020. 7

[35] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proc. IEEE ICCV*, 2019. 1, 2

[36] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proc. IEEE CVPR*, 2018. 1, 2

[37] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proc. ACL*, 2019. 2

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017. 2, 3, 4, 5

[39] Xionghui Wang, Jian-Fang Hu, Jian-Huang Lai, Jianguo Zhang, and Wei-Shi Zheng. Progressive teacher-student learning for early action prediction. In *Proc. IEEE CVPR*, 2019. 3

[40] Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. Multi-modality cross attention network for image and sentence matching. In *Proc. IEEE CVPR*, 2020. 2

[41] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proc. ECCV*, 2018. 2

[42] Meng-Chieh Wu, Ching-Te Chiu, and Kun-Hsuan Wu. Multi-teacher knowledge distillation for compressed video action recognition on deep neural networks. In *Proc. ICASSP*, 2019. 3

[43] Wenhao Wu, Dongliang He, Xiao Tan, Shifeng Chen, and Shilei Wen. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *Proc. IEEE ICCV*, 2019. 2

[44] Zuxuan Wu, Caiming Xiong, Yu-Gang Jiang, and Larry S Davis. Liteeval: A coarse-to-fine framework for resource efficient video recognition. In *Proc. NeurIPS*, 2019. 1, 2, 6

[45] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proc. IEEE CVPR*, 2019. 1, 2, 6

[46] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. In *Proc. ECCV*, 2018. 1, 2

[47] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proc. IEEE CVPR*, 2016. 2

[48] 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. 2