Conditional Model Selection for Efficient Video Understanding

Mihir Jain[1]*
Haitam Ben Yahia[1]*
Amir Ghodrati[1]
Fatih Porikli[2]
Amirhossein Habibian[1]

[1] Qualcomm AI Research[§],
Qualcomm Technologies Netherlands B.V.

[2] Qualcomm AI Research[§],
Qualcomm Technologies, Inc.

{mijain,hyahia,ghodrati,fporikli,ahabibia}@qti.qualcomm.com

Abstract

Video action classification and temporal localization are two key components of video understanding where we witnessed significant progress leveraging neural network architectures. Recently, the research focus in this area shifted towards computationally efficient solutions to support real-world applications. Existing methods mainly aim to pick salient frames or video clips with fixed architectures. As an alternative, here, we propose to learn policies to select the most efficient neural model conditioned on the given input video. Specifically, we train a novel model-selector offline with model-affinity annotations that consolidate recognition quality and efficiency. Further, we incorporate the disparity between appearance and motion to estimate action background priors that enable efficient action localization without temporal annotations. To the best of our knowledge, this is the first attempt at computationally efficient action localization. We report classification results on two video benchmarks, Kinetics and multi-label HVU, and show that our method achieves state-of-the-art results while allowing a trade-off between accuracy and efficiency. For localization, we present evaluations on Thumos'14 and MultiThumos, where our approach improves or maintains the state-of-the-art performance while using only a fraction of the computation.

1 Introduction

Massive growth in the generation of video content demands for developing algorithms to understand videos automatically. To this end, a large body of research has been done to address the tasks of video classification and localization. As a result, a wide variety of diverse methods are developed, each tackling the problem from a particular perspective such as architectural design (2D image-based models [25], 3D models [7, 42], transformers [1, 32]), complementary modalities (motion [8, 31] and audio [12, 23]), and efficiency (light [25, 36] and heavy [9] models). While each approach has pros and cons, an important question is: what model should one pick to solve the task at hand? The answer to this question is becoming harder with the rapid growth of video models. A well-known ML technique to

* Equal Contribution. Correspondence to hyahia@qti.qualcomm.com

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

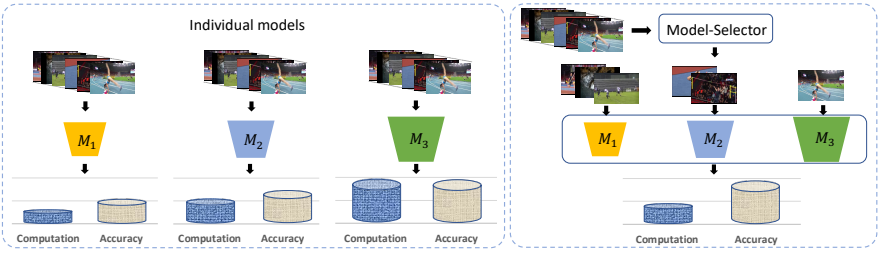


Figure 1: **Overview of model selection.** Left: usually, models, can be more accurate, but often require more computation. However, this setting is restricted to using one model for the whole dataset. Right: our approach where a model-selector will select the appropriate model, given the input video, to obtain better computation and better accuracy.

remedy this issue is ensemble learning [11, 38, 44]. This approach aggregates the prediction of each base model and results in a more reliable final prediction. However, model ensembles are usually expensive as multiple models are applied on every input video. Moreover, they assume that there exists a single combination of models that is optimal for all the video samples. In contrast, we argue that the selection process should be done on a per-video basis. We are motivated by the fact that videos come at different complexities. For example, some videos have less temporal complexity and can be reliably classified using 2D CNN models, while some others might require motion clues.

In this paper, we propose a committee-based conditional compute model that learns how to select a model per example to accurately classify it while adjusting the computation to the complexity of the video. Specifically, we train a *model-selector* with a supervision signal that considers accuracy and efficiency for each training video. We generate such supervision signal based on the recognition quality and computational cost of each model and refer to them as *model-affinity*. Our classification method is effective, efficient and complementary to the existing approaches for efficient video understanding such as using salient frames[13] or efficient architectures[7, 25]. Further, we propose an approach to learn a model-selector for efficient action localization without temporal annotations. We exploit the disparity between appearance and motion to estimate where a motion-based model can be more effective. To the best of our knowledge, this is the first attempt at efficient action localization.

Our key contribution is the novel model-selection framework conditioned on the input video. In the proposed framework, given a pool of models of varying capacities, a model-selector is learned to select a cost-effective model for a given video. Our second contribution is a method to generate motion-affinity based on the recognition quality (given video labels) and computational cost of each model. Further, to generate motion-affinity for localization in absence of temporal annotations, we propose a method to exploit the disparity between appearance and motion. Finally, our method exceeds the state-of-the-art methods on two datasets for video classification and two datasets for weakly-supervised localization.

2 Related work

Efficient action recognition: Extensive studies have been conducted to design efficient video models. There are two lines of work in the literature to achieve this goal. The first approach focuses on designing single lightweight architectures, for example, by using neural

architecture search [7, 36], decomposing 3D kernels into spatial and temporal kernels [37, 43, 51], or shifting channels [6, 25]. On the other hand, a large body of research has been focusing on selecting a subset of salient frames to efficiently process a video conditioned on the input [12, 13, 29, 49, 50]. This is commonly done by training agents to find which frame to observe next [5, 50, 52, 56], reformulating the problem in the early exiting framework [13], finding relevant frames off-line [12, 21], gating frames [17], skipping RNN states [7], or adaptively selecting the input resolution for each frame [29]. While a single network is employed to attain efficiency in the above approaches, we propose to select an appropriate model adaptively from a collection of models to obtain a favorable speed-accuracy trade-off. Our method is orthogonal to current practices and complements them.

Weakly-supervised action localization: Several weakly-supervised approaches have been proposed. Most methods utilize motion information and build on one or more of the multiple-instance learning, attention, cross-attention, and background modeling. Since multiple instance learning and attention were employed in [33, 35, 46], they have been integral to the methods for localization. Several losses are proposed for discriminability of action instances over a video [31] or similarity between a video pair with a common class [35]. To alleviate the confusion due to background, [34] developed the top-down class-guided attention to model the background. Further, for background modeling, temporal relations among video segments are exploited in [34] and a background suppression network is designed in [24]. A video is segmented into interpretable fragments and used to generate action proposals in [19]. To distinguish action from near-action context, [39] designed a class-agnostic frame-wise probability, conditioned on frame attention, using conditional variational auto-encoder. In [35], the attentions from RGB and flow streams are fused, while [23] combines visual cues with audio by cross-attention to improve localization. A common and critical factor for the success of the above works (all two-stream models) is the use of optical flow along with RGB, especially to differentiate the action foreground from the background. We exploit this factor in our model-selection method for action localization.

Committee-based learning: Model ensembles is a well-known ML technique [11, 38, 44] for obtaining better predictive performance than the constituent models alone. The technique aggregates the prediction of each base model and results in a more reliable final prediction [16, 22, 27, 47, 48]. Another family of committee-based models is model cascading. It speeds up the model ensembles by sequentially applying each model and using heuristic criteria to determine when to exit from the cascade [14, 15, 40, 47]. Different from previous works that focus on designing ensembles or cascade schemes, we focus on learning a policy function to select a single desired model from a given pool of models, obtaining a favorable speed-accuracy trade-off. Some of the recent works [10, 28, 45] proposed learning schemes similar to ours, however, none of them are focused on video classification and localization. Only one [45] of them focuses on visual recognition suggesting the possible benefit of model selection for object detection. Unlike these works, our framework can also learn with weak supervision and its applicability is validated for multiple video recognition tasks.

3 Approach

This section describes our model selection process conditioned on input samples for video classification and weakly-supervised action localization. An overview of our method is shown in Figure 2. Our goal is to learn a policy model that selects the most suitable model at inference time from a pool of models with varying capacities. To this end, we first assess each available model offline on a set of training samples. This assessment assigns an affinity

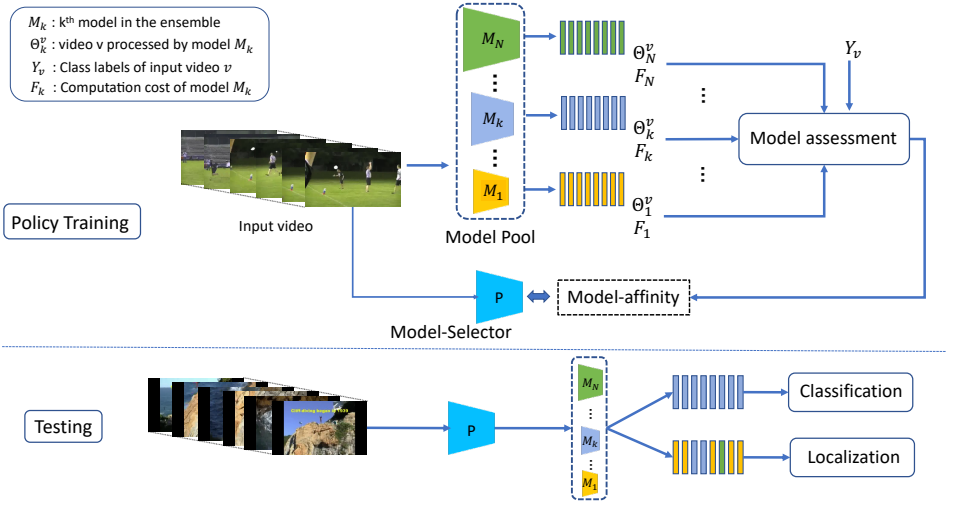


Figure 2: **Overview of method:** During training, we extract features with our model pool. These along with the computational cost of the model and video labels are used for *model assessment*, which generates *model-affinity*. This cost-effectiveness per model will be used as a supervision signal to the *model-selector*. During testing, we simply use the model-selector and run the corresponding model per video for classification or per clip for localization.

score to each training sample for every model in the pool. The affinity score incorporates the accuracy of the model for the given sample and the associated computational cost of the model. Using these affinity scores, we train a *model-selector* that maps the input samples to the corresponding model-affinity. Next, we explain in detail each component of our method.

3.1 Policy model training

The two main components of the policy model training are *model assessment* and the *model-selector*. Suppose we are given a pool of N models M_1 to M_N with increasing capacity. The model assessment module generates model-affinity scores $G_k(x)$ arranged into a vector for a given training sample x for each model M_k in the pool. It measures how well-suited each model from the pool is for efficient recognition of the given sample. Please see Sections 3.2 and 3.3 for specific designs of the model affinity. This model-affinity acts as a supervision signal to train the model-selector. To minimize the computational load at inference time, we design the model-selector as a simple multi-layer perceptron (MLP) mounted on the most efficient (smallest) model in the pool. Given a batch of samples $x \in B$ and their model affinity scalar $G_k(x)$, the model-selector P is learned with the following loss:

$$\mathcal{L} = \sum_{x \in B} \sum_{k=1}^{N} -G_k(x) \log P(M_k, x) \tag{1}$$

where $P(M_k, x)$ is the probability that model-selector selects model M_k for sample x. Here, $G_k(x)$ can also be represented as a binary scalar by considering the most suited model as a pseudo-label; in that case, Eq. 1 represents the cross-entropy loss.

3.2 Model selection for video classification

For classification tasks, our model-affinity needs to indicate the model with the best efficiency-accuracy trade-off in the model pool for a particular video x. To accommodate this, we define

$$G_k(x) = \mathbb{1}(\underset{i}{\operatorname{argmin}}[L_{cls}(M_i, x, Y_x) + \alpha F_i] = k) \tag{2}$$

as the model-affinity binary scalar, where $\mathbb{1}()$ is the indicator function, $L_{cls}(M_i, x, Y_x)$ is the downstream video classification loss, F_i is the normalized cost of model M_i and Y_x is the label of video x. Intuitively, this means $G_k(x) = 1$, when M_k has lowest loss + weighted FLOPs in the model pool and 0 otherwise. In the multi-label video classification case, $L_{cls}(M_i, x, Y_x)$ is the binary cross-entropy loss, whereas in the single-label case it is regular cross-entropy. A regularizing factor α penalizes large number of FLOPs and F_k is normalized as $F_k = \frac{flops(M_k)}{\sum_{i=1}^{N} flops(M_i)}$, to avoid imbalanced model supervision. When $\alpha = 0$, the $G_k(x)$ indicates the model with lowest loss. We observe that in some cases this strategy results in a lower amount of FLOPs than the model with the most compute. This is because models that perform better on average do not necessarily perform better on every example. Models with lower compute might have different inductive biases that are complimentary or simply generalize better for certain examples due to their smaller size. However, generally speaking, an $\alpha > 0$ will encourage better trade-offs between accuracy and efficiency. For example, in the single-label classification case, we prefer the model with the lowest compute, if the argmax over our class probabilities is the same for all models in the model pool.

3.3 Model selection for weakly-supervised action localization

For video classification, video-level annotations are available for each sample (i.e. video) during training. Differently, in weakly-supervised localization, each clip in the video needs to be recognized as belonging to an action class or the background, without the clip-level annotations. Hence, model assessment needs to be done for each clip with video-level labels and without the knowledge of which clips belong to the background. With such weak-supervision, the localization losses in the literature are defined either for the whole video [39] or for a pair of videos[35], but not for a single clip. We approach this problem by keeping an appearance-based model M_{rgb} and a motion-based model M_{motion} in the model pool. We observe that the motion model and RGB model often disagree on action background where action-related appearance context is still present, *e.g.* running track is visible just after an instance of 'long-jump'. We show this in an ablation in Section 5.2. We propose the idea of using disparity between class activations of M_{rgb} and M_{motion} as a proxy for estimating the background clips with action context, expecting mostly high M_{rgb} activations and low M_{motion} activations. Also, often on key moments of action, M_{motion} activations are higher. We model the likelihood of where motion should be given priority as:

$$D_{motion}(x) = \frac{|M_{rgb}(x,c) - M_{motion}(x,c)|}{\Gamma_v}. \tag{3}$$

$M_{rgb}(x,c)$ and $M_{motion}(x,c)$ are activations for class c, predicted by M_{rgb}. Γ_v is set to maximum absolute difference between the two activations over time for video v.

The distribution of D_{motion} can vary widely for different videos, so in order to get the pseudo-labels for a given video we apply a video-specific threshold τ_v on D_{motion}. When

$D_{motion}(x) > \tau_v$, the pseudo-label $Q_k(x)$ indicates the motion model, otherwise it indicates the appearance model. For the desired computation budget per video, τ_v is set such that the top β fraction of clips are assigned to M_{motion}. Consequently, model-affinity is defined as:

$$G_k(x) = Q_k(x)(|D_{motion}(x) - \tau_v| + 1). \tag{4}$$

Note here that as the disparity D_{motion} goes below τ_v the model-affinity increases in favour of M_{rgb} and when it is high it favours M_{motion}, which is typically more expensive. One can alternatively also learn model-selector P, purely with pseudo-labels, i.e., $G_k(x) = Q_k(x)$. We analyze this in Section 5.2.

Discussion: Our model assessment for localization assumes that similar appearance cues are often present both during, and in between actions (verified in Figure 5-a). This can be a limitation of using motion disparity in rare cases, where videos are devoid of related background in between actions, *e.g.* videos of sports highlights. An alternative could be to design a clip-level localization loss that works with weak supervision. However, we did not get encouraging results in our experiments adapting existing losses [31, 35]. Ultimately, it comes down to applying the best model on the training videos and use the result as pseudo-labels to learn the model-selector. Our framework allows for this as shown in Figure 5-c, where we train model-selector using pseudo-labels. Though motion disparity better exploits frequent scenarios, our overall framework has wider applicability.

3.4 Model selection at inference

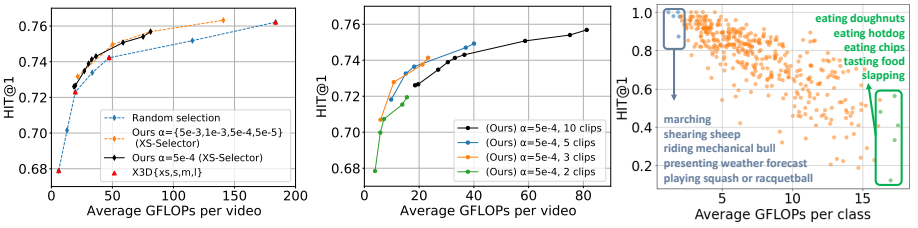
During inference, our model-selector, P, is applied to the input video to obtain confidence for each model in the pool. According to the compute requirements, we set a threshold on these confidence scores and select the most efficient model above the threshold. This is done at the video-level for classification and at the clip-level for localization.

4 Video classification

4.1 Experimental setup

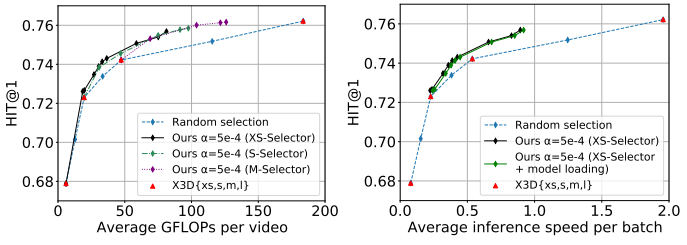
Datasets: We conduct our video classification experiments on two large datasets, Kinetics-400 [20] and HVU [4]. Kinetics includes 250K and 20K clips for training and evaluation, respectively. Each video is annotated as 1 out of 400 action classes. Additionally, we evaluate on a subset of Kinetics-400 called Mini-Kinetics. For *HVU*, we follow the standard evaluation and use 470K train and 30K evaluation video clips. HVU, which is multi-labeled, further differs from Kinetics, which is limited to only action classes. HVU covers a broad range of classes, *e.g.* objects, attributes, and scenes. Each model is evaluated in terms of accuracy vs. efficiency. Following literature, we rely on HIT@1 for the Kinetics 400 dataset and mean average precision (mAP) for HVU to quantify the classification accuracies. For efficiency we report the average number of floating point operations (in GFLOPs).

Implementation details: For Kinetics experiments, we consider a pool of {XS, S, M, L} X3D [7] models. We train our model selector using XS model predictions, as input, using an Adam optimizer with a learning rate of $1e-3$ decaying by 0.1 at epochs 2 and 4. We train the model for 5 epochs. For HVU experiments, we consider a pool of X3D-M and EfficientNet-B0 [41] models pre-trained on Kinetics and Imagenet, respectively. The model selector is trained over EfficientNet-B0 predictions as input for 1 epoch using a learning rate of $1e-3$. We include more details in the supplementary materials.



(a) (b) (c)

Figure 3: **HIT@1 vs. efficiency on Kinetics-400:** Ablation showing our method with varying values for alpha and number of views in (a) and (b) respectively. (a) Shows our method outperforms random selection and models in model pool and (b) shows we can enjoy efficiency gain from aggregating less clips. In (c) we show average GFLOPs per class.



(a) (b)

Figure 4: **Kinetics-400 Model-Selector & Latency:** In (a) we see results for 3 different model selectors. Starting at their respective model selector backbone the results all converge to a similar curve (though in different GFLOP regions). In (b) we have measured latency against HIT@1 instead of GFLOPs as in Figure 3-a.

4.2 Analysis

Varing the α parameter: We analyse the effect of the parameter α (from Eq. 2) by training the X3d XS based selector with different α ranging from $5e^{-5}$ to $5e^{-3}$. As shown in Figure 3-a, we see that our model-selector at test time compares favourably against a random selector baseline. It demonstrates that our model-selector has learned to pick better models. Moreover, we observe that by adjusting the decision threshold for $\alpha = 5e^{-4}$, our selector smoothly balances the overall accuracy vs. the computation cost without retraining. Finally, we observe that our selection outperforms all the individual models from the pool.

Clip-level ablation: In Figure 3-b we also study whether our proposed method is complementary to frame sampling based methods. It shows that we can indeed get better trade-offs by lowering the number of clips per video. We draw the same conclusion on the HVU dataset, which is added to the supplementary material.

Compute per class: Since our model-selector is flexible in assigning compute conditioned on the input, we also measure the average amount of compute for particular classes. In Figure 3-c, we show HIT@1 for different classes against its average compute. Note that it assigns cheaper models to classes it can easily detect like *shearing sheep* and *marching*, and expensive models to more complex classes like *tasting food* and *eating chips*.

Model-selector ablation: In Figure 4-a, we see that different model selectors all obtain

HVU	GFLOPs	mAP.
3D Resnet-18* [12]	38.6	35.4
HATNet* [4]	41.8	39.6
FrameExit ($\beta = 1^{e-2}$) [13]	5.7	46.1
FrameExit ($\beta = 1^{e-3}$) [13]	11.7	47.7
X3D-M [8]	9.46	49.1
Ours	11.5	**50.1**

Mini-Kinetics	GFLOPs	HIT@1
LiteEval [19]	99.0	61.0
SCSampler [21]	41.9	70.8
ARNet [25]	32.0	71.7
FrameExit [13]	7.8	75.3
X3D-M [8]	14.19	81.0
Ours	7.9	**83.5**

Table 1: **State-of-the-art comparisons**, where we pick our model with similar average GFLOPs per video as alternative methods. *GFLOPs are per clip, instead of per video. Note that we outperform SOTA on both datasets at a similar number of GFLOPs

(similar) favourable accuracy-efficiency trade-offs. In fact, all overlap in certain GFLOP ranges with comparable trade-offs. However, a clear limitation of choosing a more expensive model-selector is seen by the fact that the model-selector is run all the time, hence we cannot obtain performance at lower GFLOPs than the model selector. Thus using X3D-XS as the model selector is a more flexible option.

Latency impact: In Figure 4-b, we show an efficiency-accuracy plot with inference speed instead of GFLOPs. These measurements are obtained with a 2080TI 12GB GPU, using a batch size of 32. The curve in black shows the performance with keeping the model pool in memory. Note that we can keep around 150 sets of our model pool on GPU at test time before running out of memory. However, to simulate memory-constrained environments, we also measure loading the model at every selection in green, which shows minimal overhead.

4.3 State-of-the-art comparison

In Table 1, we compare against state-of-the-art models. Here, we use 2 clips for HVU and 3 clips for Mini-Kinetics. We pick our model based on the GFLOPs of the state-of-the-art model FrameExit [13] and evaluate for accuracy. We see a difference of 2.4 mAP and 8.2 HIT@1 percentile points over FrameExit at roughly the same GFLOPs for HVU and Mini-Kinetics respectively. This is in large part due to our model pool, however, we also see an increase over this comparing against X3D-M which is of roughly equal size. This shows the effectiveness of our model-selector.

5 Weakly-supervised action localization

5.1 Experimental setup

Datasets: For evaluation we use Thumos'14 [18] and MultiThumos [53] datasets. The Thumos'14 dataset has 20 action classes with about 15.5 action instances per video. We follow the convention to train on the validation set of 200 videos and evaluate on the test set of 212 videos. MultiThumos has the same set of videos as in Thumos'14, but it extends to 65 classes with 1.5 labels per frame, making it a more challenging multi-label dataset. We report mean Average Precision (mAP) under different intersection over union (IoU) thresholds.

Implementation Details: We mount our model selection on W-TALC [35] method and train two localization models, M_{rgb} using X3D-M features and M_{motion} I3D-motion features. The X3D-M and I3D-motion [3] features come from models pretrained on Kinetics dataset. Each video stream is divided into 16-frame non-overlapping clips. To temporally align to

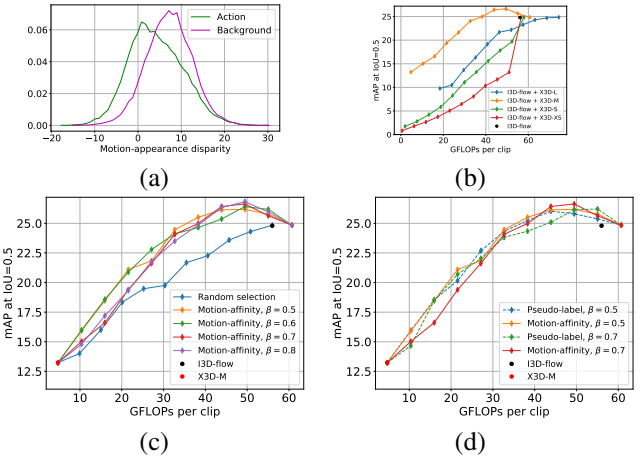


Figure 5: **Ablation for localization on Thumos'14:** (a) Distribution of disparity between motion and appearance. (b) Ablation on β and comparison with random baseline. (c) Training model-selector with pseudo-labels. (d) Varying model-selector backbone ($\beta = 0.7$).

this, we adapt the X3D-M by extracting frames at 25 Hz and set stride to 1. We do not retrain it and use the weights provided by the authors. During inference, we further enhance the action proposals by ActionBytes [19] post-processing.

5.2 Analysis

In this section, we analyze the components of our method on Thumos'14 dataset.

Motion vs appearance for background localization: In Figure 5-a, we show distribution of disparity between motion and appearance activations (Eq. 3) in the action foreground and background. The distribution for the background is shifted towards the right confirming that often appearance activation is higher due to action context in the background. This enables us to use the disparity, D_{motion}, to estimate the background and motion affinity.

Impact of computation budget factor β: Computation budget factor β controls the emphasis on computation while training model-selector. Figure 5-c plots the mAP against computation in GFLOPs. Here, for all values of β our model-selector performs better than a random selection. For smaller values of β, the selector does slightly better at lower compute, but in the more interesting range (mAP>24%), higher values do a bit better. Due to complementary nature of X3D-M and I3D-flow models, the random selection baseline is competitive at lower compute but it cannot select well when more motion clips are used.

Motion-affinity as pseudo-labels: An alternative way to train model-selector is using motion-affinity as pseudo-labels, i.e., $G_k(x) = Q_k(x)$. In Figure 5-d, we evaluate this for two values of β and compare with the default motion-affinity supervision. There is not much difference between the two, but in the more interesting range of mAP-GFLOP trade-off, our default method performs better and is also more robust to hyper-parameter β. As a result of this and the ablation over β, we choose our default motion-affinity and $\beta = 0.7$ here on.

Ablation on the choice of model-selector: Impact of varying model-selector backbone is analyzed in Figure 5-b. The more efficient X3D-XS and X3D-S models fail to perform well. Other than being too small, they are affected by the temporal alignment that requires

them to represent a 16-frame clip by only 4 or 13 frames. The temporal alignment has less impact on X3D-M and X3D-L, but the latter is affected by the low resolution (320 $\times$ 180) of the Thumos'14 videos. For X3D-L, the smaller dimension of 180 needs to be scaled up to 356 versus 256 for X3D-M. Hence, unless the X3D-L is modified and retrained to match the input to I3D-flow spatiotemporally, X3D-M is the obvious choice here.

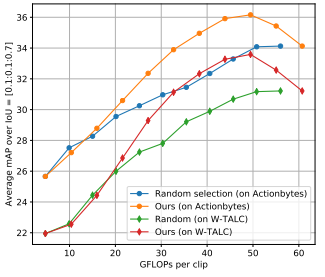


Figure 6: **Average mAP vs. GFLOPs** plots (Thumos'14) with W-TALC and ActionBytes as underlying methods.

Method	Thumos'14 mAP at IoU (%) 0.5	0.7	Avg. mAP	MultiThumos mAP at IoU (%) 0.5	Avg. mAP	GFLOPs
Liu *et al.* [22]	23.1	7.0	32.4	-	-	112
Nguyen *et al.* [30]	26.8	9.0	36.3	-	-	112
BaS-Net [21]	27.0	10.4	35.3	-	-	112
DGAM [32]	28.8	11.4	37.0	-		112
ActionBytes [15]	29.0	9.5	-	12.1	21.5	112
†ActionBytes (flow) [15]	28.6	10.0	34.1	11.5	19.8	56
A2CL-PT [31]	30.1	10.6	37.8	-	-	112
TSCN [6]	28.7	10.2	37.8	-	-	112
Ours	30.0	10.4	35.9	11.7	20.7	43.8

Table 2: **Comparison on Thumos'14 and MultiThumos** We report mAP values varying IoUs and also average mAP over IoU=[0.1:0.7:0.1] for Thumos'14 and IoUs=[0.1:0.5:0.1] for MultiThumos. † Reproduced results.

Varying underlying localization method: Figure 6 plots average of mAPs over IoUs [0.1:0.7:0.1] against computation in GFLOPs per clip. Here, in addition to applying our method on the W-TALC method, we also plot for an alternative localization method, ActionBytes. For both the methods we achieve better accuracy-efficiency trade-off compared to a random selection from X3D-M and I3D-flow models.

5.3 State-of-the-art comparison

In Table 2, we compare our model-selection method (on ActionBytes) with the SOTA methods. We report average mAP over IoUs for Thumos'14 and MutltiThumos datasets. All competing methods have the same 2-stream I3D backbone, except 'Actionbytes (flow)'. GFLOPs are equal for the same backbones, because we do not include relatively insignificant computational costs of different localization heads. We are better or comparable to these methods while spending about 40% of their compute (78% of ActionBytes-flow). This shows the efficacy of our model-selection method for efficient weakly-supervised action localization.

6 Conclusion

We propose a novel model-selection framework that, for a given video, selects a model that can effectively recognize actions while spending just enough computation. For this, we train a model-selector supervised by motion-affinity. It is a supervision signal designed to optimize for accuracy and efficiency. In the absence of temporal annotations, we exploit disparity between appearance and motion to estimate action background and generate motion-affinity for weakly-supervised localization. We demonstrate the utility of our framework for both classification and localization. For classification, our method exceeds SOTA methods, with 8.2 mAP and 2.4 HIT@1 absolute points, on the Mini-Kinetics and HVU datasets respectively, while using about the same amount of computation. For localization, we report on Thumos'14 and MultiThumos datasets, maintaining or improving the performance of SOTA methods while using about 40% of the computation.

References

[1] A. Arnab, M. Dehghani, G. Heigold, Chen Sun, Mario Lucic, and C. Schmid. Vivit: A video vision transformer. *ArXiv*, abs/2103.15691, 2021.

[2] 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. *arXiv preprint arXiv:1708.06834*, 2017.

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

[4] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Manohar Paluri, Jürgen Gall, Rainer Stiefelhagen, and Luc Van Gool. Large scale holistic video understanding. In *European Conference on Computer Vision*, pages 593–610. Springer, 2020.

[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 International Joint Conference on Artificial Intelligence*, 2018.

[6] 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 *European Conference on Computer Vision*, pages 505–521. Springer, 2020.

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

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6201–6210, 2019.

[10] Cong Feng and Jie Zhang. Reinforcement learning based dynamic model selection for short-term load forecasting. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2019.

[11] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55 (1):119–139, 1997.

[12] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10457–10467, 2020.

[13] Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. Frameexit: Conditional early exiting for efficient video recognition. In *CVPR*, 2021.

[14] Jiaqi Guan, Yang Liu, Qiang Liu, and Jian Peng. Energy-efficient amortized inference with cascaded deep classifiers. *arXiv preprint arXiv:1710.03368*, 2017.

[15] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017.

[16] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

[17] Noureldien Hussein, Mihir Jain, and Babak Ehteshami Bejnordi. Timegate: Conditional gating of segments in long-range activities. *arXiv preprint arXiv:2004.01808*, 2020.

[18] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[19] Mihir Jain, Amir Ghodrati, and Cees G. M. Snoek. ActionBytes: Learning from trimmed videos to localize actions. In *CVPR*. 2020.

[20] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[21] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6232–6242, 2019.

[22] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[23] Jun-Tae Lee, Mihir Jain, Hyoungwoo Park, and Sungrack Yun. Cross-attentional audio-visual fusion for weakly-supervised action localization. In *International Conference on Learning Representations*, 2020.

[24] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11320–11327, 2020.

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

[26] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1298–1307, 2019.

[27] Ekaterina Lobacheva, Nadezhda Chirkova, Maxim Kodryan, and Dmitry Vetrov. On power laws in deep ensembles. *arXiv preprint arXiv:2007.08483*, 2020.

[28] Bingqian Lu, Jianyi Yang, Lydia Y Chen, and Shaolei Ren. Automating deep neural network model selection for edge inference. In *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*, pages 184–193. IEEE, 2019.

[29] 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. *arXiv preprint arXiv:2007.15796*, 2020.

[30] Kyle Min and Jason J Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. In *European Conference on Computer Vision*, pages 283–299. Springer, 2020.

[31] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3C-Net: Category count and center loss for weakly-supervised action localization. In *ICCV*. 2019.

[32] Daniel Neimark, O. Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *ArXiv*, abs/2102.00719, 2021.

[33] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*. 2018.

[34] Phuc Xuan Nguyen, Deva Ramanan, and Charless C. Fowlkes. Weakly-supervised action localization with background modeling. In *ICCV*. 2019.

[35] Sujoy Paul, Sourya Roy, and Amit K. Roy-Chowdhury. W-TALC: Weakly-supervised temporal activity localization and classification. 2018.

[36] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Tiny video networks. *arXiv preprint arXiv:1910.06961*, 2019.

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

[38] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[39] Baifeng Shi, Qi Dai, Yadong Mu, and Jingdong Wang. Weakly-supervised action localization by generative attention modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1019, 2020.

[40] Matthew Streeter. Approximation algorithms for cascading prediction models. In *International Conference on Machine Learning*, pages 4752–4760. PMLR, 2018.

[41] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[42] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017.

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

[44] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[45] Emanuele Vitali, Anton Lokhmotov, and Gianluca Palermo. Dynamic network selection for the object detection task: why it matters and what we (didn't) achieve. *arXiv preprint arXiv:2105.13279*, 2021.

[46] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. UntrimmedNets for weakly supervised action recognition and detection. In *CVPR*. 2017.

[47] Xiaofang Wang, Dan Kondratyuk, Eric Christiansen, Kris M Kitani, Yair Movshovitz-Attias, and Elad Eban. On the surprising efficiency of committee-based models. *arXiv preprint arXiv:2012.01988*, 2020.

[48] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

[49] Zuxuan Wu, Caiming Xiong, Yu-Gang Jiang, and Larry S Davis. Liteeval: A coarse-to-fine framework for resource efficient video recognition. In *Advances in Neural Information Processing Systems*, pages 7780–7789, 2019.

[50] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2019.

[51] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 1(2):5, 2017.

[52] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016.

[53] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018.

[54] Tan Yu, Zhou Ren, Yuncheng Li, Enxu Yan, Ning Xu, and Junsong Yuan. Temporal structure mining for weakly supervised action detection. 2019.

[55] Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, Junsong Yuan, and Gang Hua. Two-stream consensus network for weakly-supervised temporal action localization. In *European Conference on Computer Vision*, pages 37–54. Springer, 2020.

[56] 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.