AdaMML: Adaptive Multi-Modal Learning for Efficient Video Recognition

Rameswar Panda[1,†], Chun-Fu (Richard) Chen[1,†], Quanfu Fan[1], Ximeng Sun[2],
Kate Saenko[1,2], Aude Oliva[1,3], Rogerio Feris[1]
†: Equal Contribution
[1]MIT-IBM Watson AI Lab, [2]Boston University, [3]MIT

Abstract

Multi-modal learning, which focuses on utilizing various modalities to improve the performance of a model, is widely used in video recognition. While traditional multi-modal learning offers excellent recognition results, its computational expense limits its impact for many real-world applications. In this paper, we propose an adaptive multi-modal learning framework, called AdaMML, that selects on-the-fly the optimal modalities for each segment conditioned on the input for efficient video recognition. Specifically, given a video segment, a multi-modal policy network is used to decide what modalities should be used for processing by the recognition model, with the goal of improving both accuracy and efficiency. We efficiently train the policy network jointly with the recognition model using standard back-propagation. Extensive experiments on four challenging diverse datasets demonstrate that our proposed adaptive approach yields $35\% - 55\%$ reduction in computation when compared to the traditional baseline that simply uses all the modalities irrespective of the input, while also achieving consistent improvements in accuracy over the state-of-the-art methods. Project page: https://rpand002.github.io/adamml.html.

1. Introduction

Videos are rich in multiple modalities: RGB frames, motion (optical flow), and audio. As a result, multi-modal learning which focuses on utilizing various modalities to improve the performance of a video recognition model, has attracted much attention in the recent years. Despite encouraging progress, multi-modal learning becomes computationally impractical in real-world scenarios where the videos are untrimmed and span several minutes or even hours. Given a long video, some modalities often provide irrelevant/redundant information for the recognition of the action class. Thus, utilizing information from all the input modalities may be counterproductive as informative modalities are often overwhelmed by uninformative ones in long videos. Furthermore, some modalities require more compu-

tation than others and hence selecting the cheaper modality with good performance can significantly save computation leading to more efficient video recognition.

Let us consider the video in Figure 1, represented by eight uniformly sampled video segments from a video. We ask, *Do all the segments require both RGB and audio stream to recognize the action as "Mowing the Lawn" in this video?* The answer is clear: No, the lawn mower is moving with relevant audio only in the third and sixth segment, therefore we need both RGB and audio streams for these two video segments to improve the model confidence for recognizing the correct action, while the rest of the segments can be processed with only one modality or even skipped (e.g., the first and last video segment) without losing any accuracy, resulting in large computational savings compared to processing all the segments using both modalities. Thus, in contrast to the commonly used one-size-fits-all scheme for multi-modal learning, we would like these decisions to be made individually per input segment, leading to different amounts of computation for different videos. Based on this intuition, we present a new perspective for efficient video recognition by adaptively selecting input modalities, on a per segment basis, for recognizing complex actions.

In this paper, we propose AdaMML, a novel and differentiable approach to learn a decision policy that selects optimal modalities conditioned on the inputs for efficient video recognition. Specifically, our main idea is to learn a model (referred to as the multi-modal policy network) that outputs the posterior probabilities of all the binary decisions for using or skipping each modality on a per segment basis. As these decision functions are discrete and non-differentiable, we rely on an efficient Gumbel-Softmax sampling approach [23] to learn the decision policy jointly with the network parameters through standard back-propagation, without resorting to complex reinforcement learning as in [60, 61]. We design the objective function to achieve both competitive performance and efficiency required for video recognition. We demonstrate that adaptively selecting input modalities by a lightweight policy network yields not only significant savings in computation (e.g., about 47.3% and 35.2% less
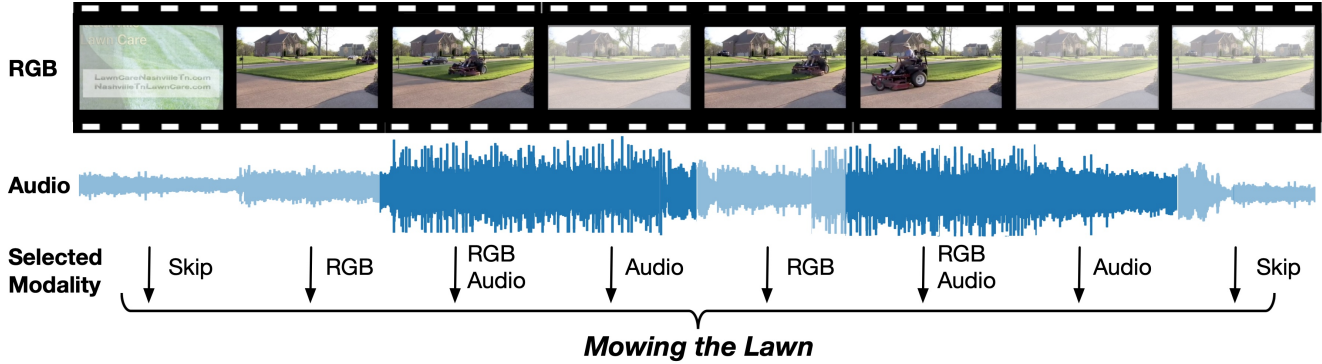
Figure 1: **A conceptual overview of our approach**. Rather than processing both RGB and Audio modalities for all the segments, our approach learns a policy to select the optimal modalities per input segment, that is needed to correctly recognize an action in a given video. In the figure, the lawn mower is moving with relevant audio only in the third and sixth segment, therefore those segments could be processed using both modalities, while the rest of the segments require only one modality (e.g., only audio is relevant for the fourth segment as the lawn mower moves outside of the camera but its sound is still clear) or even skipped (e.g., both of the modalities are irrelevant in the first and the last segment), without losing any accuracy. Note that our approach can be extended to any number of modalities as shown in experiments.

GFLOPS compared to a weighted fusion baseline that simply uses all the modalities, on Kinetics-Sounds [2] and ActivityNet [6] respectively), but also consistent improvement in accuracy over the state-of-the-art methods.

The main contributions of our work are as follows:

- We propose a novel and differentiable approach that automatically determines what modalities to use per segment per input for efficient video recognition. This is in sharp contrast to current multi-modal learning approaches that utilizes all the input modalities without considering their relevance to the video recognition.

- We efficiently train the multi-modal policy network jointly with the recognition model using standard back-propagation through Gumbel-Softmax sampling.

- We conduct extensive experiments on four video benchmarks (Kinetics-Sounds [2], ActivityNet [6], FCVID [24] and Mini-Sports1M [25]) with different multi-modal learning tasks (RGB + Audio, RGB + Flow, and RGB + Flow + Audio) to demonstrate the superiority of our approach over state-of-the-art methods.

2. Related Work

Efficient Video Recognition. Video recognition has been one of the most active research areas in computer vision recently [8]. In the context of deep neural networks, it is typically performed by either 2D-CNNs [25, 51, 12, 53, 12, 32, 63] or 3D-CNNs [48, 7, 20, 13]. While extensive studies have been conducted in the last few years, limited efforts have been made towards efficient video recognition. Specifically, methods for efficient recognition focus on either designing new lightweight architectures (e.g., Tiny Video Networks [39], channel-separated CNNs [49], and X3D [13]) or selecting salient frames/clips [61, 60, 30, 17, 57, 22, 34,

35, 37]. Our approach is most related to the latter which focuses on conditional computation for videos and is agnostic to the network architecture used for recognizing videos. Representative methods typically use reinforcement learning (RL) policy gradients [61, 60] or audio [30, 17] to select relevant video frames. LiteEval [59] proposes a coarse-to-fine framework that uses a binary gate for selecting either coarse or fine features. Unlike existing works, our proposed approach focuses on the multi-modal nature of videos and adaptively selects the right modality per input instance for recognizing complex actions in long videos. Moreover, our framework is fully differentiable, and thus is easier to train than complex RL policy gradients [61, 60, 57].

Multi-Modal Learning. Multi-modal learning has been studied from multiple perspectives, such as two stream networks that fuse decisions from multiple modalities for classification [41, 7, 26, 27, 3], and cross-modal learning that takes one modality as input and make prediction on the other modality [29, 2, 62, 1, 15, 42]. Recent work in [52] addresses the problem of joint training in multi-modal networks, without deciding which modality to focus for a given input sample as in our current approach. Our proposed AdaMML framework is also related to prior works in joint appearance and motion modeling [43, 31, 10] that focuses on combining RGB and optical flow streams. Design of different fusion schemes [38] through neural architecture search [64] is also another recent trend for multi-modal learning. In contrast, we propose an instance-specific general framework for automatically selecting the right modality per segment for efficient video recognition.

Adaptive Computation. Many adaptive computation methods have been recently proposed with the goal of improving computational efficiency [4, 5, 50, 54, 18, 14, 33, 34]. While BlockDrop [58] dynamically selects which layers to execute per sample during inference, GaterNet [9] proposes a gating
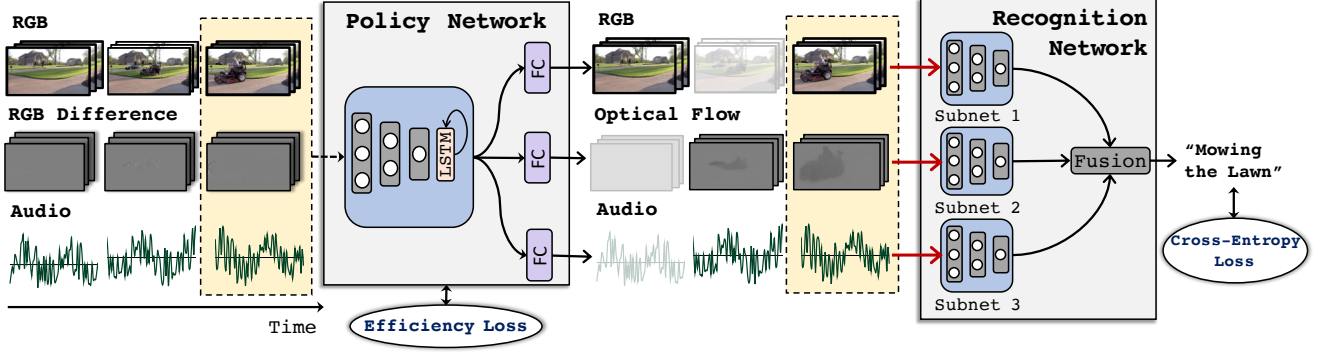
Figure 2: **Illustration of our approach**. `AdaMML` consists of a lightweight policy network and a recognition network composed of different sub-networks that are trained jointly (via late fusion with learnable weights) for recognizing videos. The policy network decides what modalities to use on a per segment basis to achieve the best recognition accuracy and efficiency in video recognition. In training, policies are sampled from a Gumbel-Softmax distribution, which allows us to optimize the policy network via backpropagation. During inference, an input segment is first fed into the policy network and then selected modalities are routed to the recognition network to generate segment-level predictions. Finally, the network averages all the segment-level predictions to obtain the video-level prediction. Best viewed in color.

network to learn channel-wise binary gates for the main network. Channel gating network [21] identifies regions in the features that contribute less to the classification result, and skips the computation on a subset of the input channels for these ineffective regions. SpotTune [19] learns to adaptively route information through fine-tuned or pre-trained layers for different tasks. Adaptive selection of different regions for fast object detection is presented in [36, 16]. While our approach is inspired by these methods, in this paper, our goal is to adaptively select optimal modalities per input instance to improve efficiency in video recognition. To the best of our knowledge, this is the first work on data-dependent selection of different modalities for efficient video recognition.

3. Proposed Method

Given a video V containing a sequence of segments $\{s_1, s_2, \cdots, s_T\}$ over K input modalities $\{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_K\}$, our goal is to seek an adaptive multi-modal selection policy that decides what input modalities should be used for each segment in order to improve the accuracy, while taking the computational efficiency into account for video recognition.

3.1. Approach Overview

Figure 2 illustrates an overview of our approach. Treating the task of finding an optimal multi-modal selection policy as a search problem quickly becomes intractable as the number of potential configurations grows exponentially with the number of video segments and modalities. Instead of handcrafting the selections, we develop a policy network that contains a very lightweight joint feature extractor and an LSTM module to output a binary policy vector per segment per input, representing whether to keep or drop an input modality for efficient multi-modal learning.

During training, the policy network is jointly trained with the recognition network using Gumbel-Softmax sampling [23]. At test time, first an input video segment is fed into the policy network, whose output decides the right modalities to use for the given segment, and then the selected input modalities are routed to the corresponding sub-networks in the recognition network to generate the segment-level predictions. Finally, the network averages all the segment-level predictions as the video-level prediction. Note that the additional computational cost incurred by the lightweight policy network (MobileNetV2 [40] in our case) is negligible in comparison to the recognition model.

3.2. Learning Adaptive Multi-Modal Policy

Multi-Modal Policy Network. The policy network contains a lightweight joint feature extractor and an LSTM module for modeling the causality across different time steps in a video. Specifically, at the t-th time step, the LSTM takes in the joint feature f_t of the current video segment s_t, previous hidden states h_{t-1}, cell outputs o_{t-1} to compute the current hidden state h_t and cell states o_t:

$$h_t, o_t = \text{LSTM}(f_t, h_{t-1}, o_{t-1}). \qquad (1)$$

Given the hidden state, the policy network estimates a policy distribution for each modality and samples binary decisions $u_{t,k}$ indicating whether to select modality k at time step t ($\mathbf{U} = \{\mathbf{u}_{t,k}\}_{l \leq T, k \leq K}$) via Gumbel-Softmax operation as described next. Given the decisions, we forward the current segment to corresponding sub-networks to get a segment-level prediction and average all segment-level predictions to generate video-level prediction for an input video.

Training using Gumbel-Softmax Sampling. `AdaMML` makes decisions about skipping or using each modality per segment per input. However, the fact that the policy

is discrete makes the network non-differentiable and therefore difficult to be optimized with standard backpropagation. One way to solve this is to convert the optimization to a reinforcement learning problem and then derive the optimal parameters of the policy network with policy gradient methods [55, 46]. However, RL policy gradients are often complex, unwieldy to train and require techniques to reduce variance during training as well as it is slow to converge in many applications [58, 59, 23, 57]. As an alternative, in this paper, we adopt Gumbel-Softmax sampling [23] to resolve this non-differentiability and enable direct optimization of the discrete policy in an efficient way.

The Gumbel-Softmax trick [23] is a simple and effective way to replace the original non-differentiable sample from a discrete distribution with a differentiable sample from a corresponding Gumbel-Softmax distribution. Specifically, at each time step t, we first generate the logits $z_k \in \mathbb{R}^2$ (i.e, output scores of policy network for modality k) from hidden states h_t by a fully-connected layer $z_k = FC_k(h_t, \theta_{FCk})$ for each modality and then use the Gumbel-Max trick [23] to draw discrete samples from a categorical distribution as:

$$\hat{P}_k = \arg\max_{i \in \{0,1\}}(\log z_{i,k} + G_{i,k}), \quad k \in [1, ..., K] \quad (2)$$

where $G_{i,k} = -\log(-\log U_{i,k})$ is a standard Gumbel distribution with $U_{i,k}$ sampled from a uniform i.i.d distribution $Unif(0, 1)$. Due to non-differentiable property of $\arg\max$ operation in Equation 2, Gumbel-Softmax distribution [23] is thus used as a continuous relaxation to $\arg\max$. Accordingly, sampling from a Gumbel-Softmax distribution allows us to backpropagate from discrete samples to the policy network. We represent $\hat{P}_k$ as a one-hot vector and then one-hot coding is relaxed to a real-valued vector P_k using softmax:

$$P_{i,k} = \frac{\exp((\log z_{i,k} + G_{i,k})/\tau)}{\sum_{j \in \{0,1\}} \exp((\log z_{j,k} + G_{j,k})/\tau)}, \quad (3)$$

where $i \in \{0, 1\}$, $k \in [1, ..., K]$, τ is a temperature parameter, which controls the discreteness of P_k, as $\lim_{\tau \to +\infty} P_k$ converges to a uniform distribution and $\lim_{\tau \to 0} P_k$ becomes a one-hot vector. More specifically, when τ becomes closer to 0, the samples from the Gumbel Softmax distribution become indistinguishable from the discrete distribution (i.e, almost the same as the one-hot vector). In summary, during the forward pass, we sample the policy using Equation 2 and during the backward pass, we approximate the gradient of the discrete samples by computing the gradient of the continuous softmax relaxation in Equation 3.

3.3. Loss Function

Let $\Theta = \{\theta_\Phi, \theta_{LSTM}, \theta_{FC_1}, ..., \theta_{FC_K}, \theta_{\Psi_1}, ..., \theta_{\Psi_K}\}$ denote the total trainable parameters in our framework, where θ_Φ and θ_{LSTM} represent the parameters of the joint

feature extractor and LSTM used in the policy network respectively. $\theta_{FC_1}, ..., \theta_{FC_K}$ represent the parameters of the fully connected layers that generate policy logits from the LSTM hidden states and $\theta_{\Psi_1}, ..., \theta_{\Psi_K}$ represent the parameters of K sub-networks that are jointly trained for recognizing video. During training, we minimize the following loss to encourage both correct predictions as well as minimize the selection of modalities that require more computation.

$$\mathbb{E}_{(V,y) \sim \mathcal{D}_{train}} \left[-y \log(\mathcal{P}(V; \Theta)) + \sum_{k=1}^{K} \lambda_k \mathcal{C}_k \right],$$

$$\mathcal{C}_k = \begin{cases} (\frac{|U_k|_0}{C})^2 & \text{if correct} \\ \gamma & \text{otherwise} \end{cases} \quad (4)$$

where $\mathcal{P}(V; \Theta)$ and y represents the prediction and one-hot encoded ground truth label of the training video sample V and λ_k represents the cost associated with processing k-th modality. U_k represents the decision policy for k-th modality and $\mathcal{C}_k = (\frac{|U_k|_0}{C})^2$ measures the fraction of segments that selected modality k out of total C video segments; when a correct prediction is produced. We penalize incorrect predictions with γ, which including λ_k controls the trade-off between efficiency and accuracy. We use these parameters to vary the operating point of our model, allowing different models to be trained depending on the target budget constraint. While the first part of the Equation 4 represents the standard cross-entropy loss to measure the classification quality, the second part drives the network to learn a policy that favors selection of modality that is computationally more efficient in recognizing videos (e.g., processing RGB frames requires more computation than the audio streams).

4. Experiments

In this section, we conduct extensive experiments on four standard datasets to show that AdaMML outperforms many strong baselines including state-of-the-art methods while significantly reducing computation and qualitative analysis to verify the effectiveness of our adaptive policy learning.

4.1. Experimental Setup

Datasets and Tasks. We evaluate the performance of our approach using four datasets, namely Kinetics-Sounds [2], ActivityNet-v1.3 [6], FCVID [24], and Mini-Sports1M [25]. Kinetics-Sounds is a subset of Kinetics [7] and consists of $22, 521$ videos for training and $1, 532$ videos testing across 31 action classes [17]. ActivityNet contains $10, 024$ videos for training and $4, 926$ videos for validation across 200 action categories. FCVID has $45, 611$ videos for training and $45, 612$ videos for testing across 239 classes. Mini-Sports1M (assembled by [17]) is a subset of full Sports1M dataset [25] containing 30 videos per class in training and 10 videos per

Dataset	Kinetics-Sounds				ActivityNet			
		Selection Rate (%)				Selection Rate (%)		
Method	Acc. (%)	RGB	Audio	GFLOPs	mAP (%)	RGB	Audio	GFLOPs
RGB	82.85	100	—	141.36	73.24	100	—	141.36
Audio	65.49	—	100	3.82	13.88	—	100	3.82
Weighted Fusion	87.86	100	100	145.17	72.88	100	100	145.17
AdaMML	**88.17**	46.47	94.15	**76.45** (-47.3%)	**73.91**	76.25	56.35	**94.01** (-35.2%)

Table 1: **Video recognition results with RGB + Audio modalities on Kinetics-Sounds and ActivityNet**. On both datasets, our proposed approach AdaMML outperforms the weighted fusion baseline while offering significant computational savings (shown in blue).

		Selection Rate (%)		
Method	Acc. (%)	RGB	Flow	GFLOPs
RGB	82.85	100	—	141.36
Flow	75.73	—	100	163.39
Weighted Fusion	83.47	100	100	304.75
AdaMML-Flow	83.82	56.04	36.39	151.54 (-50.3%)
AdaMML-RGBDiff	**84.36**	44.61	37.40	**137.03** (-55.0%)

Table 2: **RGB + Flow on Kinetics-Sounds**. AdaMML-RGBDiff obtains best performance with more than 50% savings in GFLOPs.

		Selection Rate (%)			
Method	Acc. (%)	RGB	Flow	Audio	GFLOPs
RGB	82.85	100	—	—	141.36
Flow	75.73	—	100	—	163.39
Audio	65.49	—	—	100	3.82
Weighted Fusion	88.25	100	100	100	308.56
AdaMML-Flow	88.54	56.13	20.31	97.49	**132.94** (-56.9%)
AdaMML-RGBDiff	**89.06**	55.06	26.82	95.12	141.97 (-54.0%)

Table 3: **RGB + Flow + Audio on Kinetics-Sounds**. AdaMML-RGBDiff obtains the best accuracy of 89.06% which is 6.21% more than RGB only performance with similar GFLOPS.

class in testing with a total of 487 action classes. We consider three groups of multi-modal learning tasks such as (I) RGB + Audio, (II) RGB + Flow, and (III) RGB + Flow + Audio on different datasets. More details about the datasets can be found in the supplementary material.

Data Inputs. For each input segment, we take around 1-second of data and temporally align all the modalities. For RGB, we uniformly sample 8 frames out of 32 consecutive frames ($8 \times 224 \times 224$); and for optical flow, we stack 10 interleaved horizontal and vertical optical flow frames [51]. For audio, we use a 1-channel audio-spectrogram as input [26] (256×256, which is 1.28 seconds audio segment). Note that since computing optical flow is very expensive, we utilize RGB frame difference as a proxy to flow in our policy network and compute flow when needed. For RGB frame difference, we follow similar approach used in optical flow and use an input clip $15 \times 8 \times 224 \times 224$ by simply computing the frame differences. For the policy network, we further subsample the input data for non-audio modality, e.g., the RGB input becomes $4 \times 160 \times 160$.

Implementation Details. For the recognition network, we use TSN-like ResNet-50 [51] for both RGB and Flow modalities, and MobileNetV2 [40] for the audio modality. We simply apply late-fusion with learnable weights over the predictions from each modality to obtain the final prediction. We use MobileNetV2 for all modalities in the policy network to extract features and then apply two additional FC layers with dimension $2,048$ to concatenate the features from all modalities as the joint-feature. The hidden dimension of LSTM is set to 256. We use K parallel FC layers on top of LSTM outputs to generate the binary decision policy for each modality. The computational cost for processing RGB + Audio in the policy network and the recognition network are 0.76 and 14.52 GFLOPs, respectively.

Training Details. During policy learning, we observe that optimizing for both accuracy and efficiency is not effective with a randomly initialized policy. Thus, we fix the policy network and "warm up" the recognition network using the unimodality models (trained with ImageNet weights) for 5 epochs to provide a good starting point for policy learning. We then alternatively train both policy and recognition networks for 20 epochs and then fine-tune the recognition network with a fixed policy network for another 10 epochs. We use same initialization and total number of training epochs for all the baselines (including our approach) for a fair comparison. We use 5 segments from a video during training in all our experiments (C set to 5). We use Adam [28] for the policy network and SGD [45] for the recognition network following [56, 44]. We set the initial temperature τ to 5, and gradually anneal down to 0 during the training, as in [23]. Furthermore, at test time, we use the same temperature τ that corresponded to the training epoch in the annealing schedule. The weight decay is set to 0.0001 and momentum in SGD is 0.9. λ_k is set to the ratio of the computational load between modalities and γ is 10. More implementation details are included in the supplementary material.

Baselines. We compare our approach with the following baselines and existing approaches. First, we consider unimodality baselines where we train recognition models using each modality separately. Second, we compare with a joint training baseline, denoted as "Weighted Fusion", that simply

Method	ActivityNet		FCVID	
	mAP (%)	GFLOPs	mAP (%)	GFLOPs
FrameGlimpse	60.14	33.33	67.55	30.10
FastForward	54.64	17.86	71.21	66.11
AdaFrame	71.5	78.69	80.2	75.13
LiteEval	72.7	95.1	80.0	94.3
AdaMML	**73.91**	94.01	**85.82**	93.86

Table 4: **Comparison with state-of-the-art methods on ActivityNet and FCVID**. `AdaMML` outperforms LiteEval [59] in terms of accuracy ($\sim$1%–5%) with similar computation on both datasets.

Method	Kinetics-Sounds		Mini-Sports1M	
	Acc. (%)	GFLOPs	mAP (%)	GFLOPs
LiteEval	72.02	104.06	43.64	151.83
AdaMML	**88.17**	76.45	**46.08**	138.32

Table 5: **Comparison with LiteEval [59] on Kinetics-Sounds and Mini-Sports1M**. `AdaMML` outperforms LiteEval by a significant margin in both accuracy and GFLOPs on both datasets.

uses all the modalities (instead of selecting optimal modalities per input) via late fusion with learnable weights. This serves as a very strong baseline for classification, at the cost of heavy computation. Finally, we compare our method with existing efficient video recognition approaches, including FrameGlimpse [61], FastForward [11], AdaFrame [60], LiteEval [59] and ListenToLook [17]. We directly quote the numbers reported in published papers when possible and use author's provided source codes for LiteEval on both Kinetics-Sounds and Mini-Sports1M datasets.

Evaluation Metrics. We compute either video-level mAP (mean average precision) or top-1 accuracy (average predictions of 10 224×224 center-cropped and uniformly sampled segments) to measure the overall performance of different methods. We also report the average selection rate, computed as the percentage of total segments within a modality that are selected by the policy network in the test set, to show adaptive modality selection in our proposed approach. We measure computational cost with giga floating-point operations (GFLOPs), which is a hardware independent metric.

4.2. Main Results

Comparison with Weighted Fusion Baseline. We first compare `AdaMML` with the unimodality and weighted fusion baseline on Kinetics-Sounds and ActivityNet dataset under different task combinations (Table 1-3). Note that our approach is not entirely focused on accuracy. In fact, our main objective is to achieve both competitive performance and efficiency required for video recognition. As for efficient recognition, it is very challenging to achieve improvements in both accuracy and efficiency. However, as shown in Table 1, `AdaMML` outperforms the weighted fusion baseline

Method	Network		mAP (%)	GFLOPs
	RGB	Audio		
ListenToLook	ResNet-18	ResNet-18	76.61	112.65
AdaMML $_{112 \times 112}$	ResNet-18	ResNet-18	79.48	70.87
AdaMML $_{224 \times 224}$	ResNet-18	ResNet-18	80.05	82.33
AdaMML $_{224 \times 224}$	ResNet-50	MobileNetV2	84.73	110.14
AdaMML $_{224 \times 224}$	EfficientNet-b3	EfficientNet-b0	85.62	30.55

Table 6: **Comparison with ListenToLook [17] on ActivityNet**. `AdaMML` outperforms ListenToLook by 3.44% in mAP while offering 26.9% computational savings in terms of GFLOPs.

while offering 47.3% and 35.2% reduction in GFLOPs, on Kinetics-Sounds and ActivityNet, respectively. Interestingly on ActivityNet, while performance of the weighted fusion baseline is worse than the best single stream model (i.e., RGB only), our approach outperforms the best single stream model on both datasets by adaptively selecting input modalities that are relevant for the recognition of the action class.

Table 2 and Table 3 show the results of RGB + Flow and RGB + Flow + Audio combinations on the Kinetics-Sounds. Overall, `AdaMML-Flow` (which uses optical flow in policy network) outperforms the joint training baseline while offering 50.3% (304.75 vs 151.54) and 56.9% (308.56 vs 132.94) reduction in GFLOPs on RGB + Flow and RGB + Flow + Audio combinations, respectively. `AdaMML-RGBDiff` (that uses RGBDiff in policy learning) achieves similar performance compared to `AdaMML-Flow` while alleviating computational overhead of computing optical flow (for irrelevant video segments), which shows that RGBDiff is in fact a good proxy for predicting on-demand flow computation during test time. In summary, our consistent improvements in accuracy over the weighted fusion baseline with $35\% - 55\%$ computational savings, shows the importance of adaptive modality selection for efficient video recognition.

Comparison with State-of-the-art Methods. Table 4 shows that `AdaMML` outperforms all the compared methods to achieve the best performance of 73.91% and 85.82% in mAP on ActivityNet and FCVID respectively. Our approach achieves 1.21% and 5.82% mAP improvement over LiteEval [59] with similar GFLOPs on ActivityNet and FCVID respectively. Moreover, `AdaMML` (tested using 5 segments) outperforms LiteEval by 2.70% (80.0 vs 82.70) in mAP, while saving 39.2% in GFLOPs (94.3 vs 57.3) on FCVID.

Table 5 further shows that `AdaMML` significantly outperforms LiteEval by 16.15% and 2.44%, while reducing GFLOPS by 26.5% and 8.6%, on Kinetics-Sounds and Mini-Sports1M respectively. In summary, `AdaMML` is clearly better than LiteEval in terms of both accuracy and computational cost on all datasets, making it suitable for efficient recognition. Note that FrameGlimpse [61], FastForward [11] and AdaFrame [60] have less computation as they require access to future frames unlike LiteEval and `AdaMML` that makes decision based on the current time stamp only.

Method	RGB + Audio		RGB + Flow		RGB + Flow + Audio	
	Acc. (%)	GFLOPs	Acc. (%)	GFLOPs	Acc. (%)	GFLOPs
Average Fusion	88.15	145.17	83.30	304.75	88.18	308.56
Class-wise Weighted Fusion	87.86	145.17	83.82	304.75	87.75	308.56
Max Fusion	86.49	145.17	83.47	304.75	88.06	308.56
FC2 Fusion*	87.73	145.17	83.30	304.75	87.84	308.56
Weighted Fusion	87.86	145.17	83.47	304.75	88.25	308.56
AdaMML	**88.17**	**76.45**	**84.36**	**137.03**	**89.06**	**141.97**

*: concatenating feature vectors from all modalities and add two addition fully-connected layers to fuse features.

Table 7: **Comparison with fusion strategies on Kinetics-Sounds**. AdaMML consistently outperforms hand-designed fusion strategies with overall $50\% - 60\%$ computational savings.

In addition, we also compare with ListenToLook [17] that uses both RGB and Audio to eliminate video redundancies. As ListenToLook utilizes weight distillation from Kinetics400 pretrained model, we use Kinetics400 pretrained weights instead of ImageNet weights to initialize our uni-modality models for a comparison on ActivityNet in Table 6. With the same network architecture (ResNet18) and frame resolution (112×112), AdaMML outperforms ListenToLook by a margin of 2.87% in mAP while using 37.1% less computation. This once again shows that our proposed approach of adaptively selecting right modalities on a per segment basis is able to yield not only significant savings in computation but also improvement in accuracy. To show that the benefits of our method extend even to more recent and efficient networks, we use EfficientNet [47] in our approach and observe that it provides the best recognition performance of 85.62% in mAP with only 30.55 GFLOPs ($\sim 73\%$ less computation compared to AdaMML (ResNet50 — MobileNetV2)).

4.3. Ablation Studies

Comparison with Additional Fusion Strategies. We compare with four additional fusion strategies including weighted fusion on different combinations of modalities. Table 7 shows that our approach AdaMML consistently outperforms all the hand-designed fusion strategies while offering 47.3%, 55.03% and 53.99% reduction in GFLOPs on RGB + Audio, RGB + Flow and RGB + Flow + Audio combinations on Kinetics-Sounds respectively. Furthermore, AdaMML with RGBDiff as the proxy alleviates the computational overhead of computing optical flow (which is often very expensive) making it suitable in online scenarios. Similarly, AdaMML offers 19.2% computational savings while outperforming these fusion strategies by a margin of about 2% in mAP on RGB + Audio combination on ActivityNet.

Policy Design. We investigate the effectiveness of our design of policy by either selecting or skipping both modalities at same time instead of taking the decisions per modality. In other words, we use a single FC layer in the policy network which outputs the binary decisions where 1 indicate the use of both modalities and 0 indicate skipping of both modalities in our framework. AdaMML outperforms the alternative design (88.02 vs 88.17) while saving 18% GFLOPS on Kinetics-Sounds. Selection of both modalities at the same

Method	AcitivityNet	Kinetics-Sounds
	mAP (%)	Acc. (%)
Random (Train)	70.34	84.34
Random (Test)	58.31	85.75
Random (Train + Test)	70.85	86.28
AdaMML	**73.91**	**88.17**

Table 8: **Comparison with random policy on RGB + Audio**. Random (X) denotes random selection of modalities during X-phase of the learning. AdaMML outperforms all the variants showing effectiveness of learned policy in video recognition.

time increases the computation as it favors selection of more RGB stream. On the other hand, AdaMML selects comparatively less RGB stream and focuses more on the cheaper audio stream as many actions can be recognized by only audio without looking into the RGB frames.

Comparison with Random Policy. We perform three different experiments by randomly selecting a modality with 50% probability during both training and testing. Table 8 shows that our approach AdaMML outperforms all the three variants by a large margin (e.g., 15.60% and 2.42% improvement over Random (Test) on ActivityNet and Kinetics-Sounds respectively) which demonstrates the effectiveness of our learned policy in selecting the optimal modalities per input instance while recognizing videos. We also compare these variants using the *same selection rate as ours* and AdaMML still outperforms them on ActivityNet and Kinetics-Sounds (e.g., 8.18% and 2.14% increase over Random (Test) and 2.23% and 1.55% increase over Random (Train + Test)).

Ablation on Training Losses. As discussed in Section 3.3, λ_k and γ controls the trade-off between accuracy and computational efficiency. We investigate the effect of efficiency loss in RGB + Audio experiment on Kinetics-Sounds and observe that training without efficiency loss (both λ_k and γ set to 0) achieves a video accuracy of 88.82% (an improvement of 0.65%) while requiring 47.3% more computation than AdaMML that uses efficiency loss during training. Similarly, using equal cost weights for both modalities (by setting $\lambda_{rgb}=\lambda_{audio}=1$) achieves an accuracy of 86.82% compared to 88.17% using AdaMML, with very less utilization of audio (only 39.13% in contrast to 94.15% using our approach). As processing audio stream is much cheaper, we use $\lambda_{rgb} = 1$ and $\lambda_{audio} = 0.05$ to favor selection of cheaper modalities and achieves an accuracy of 88.17% with 76.45 GFLOPs on Kinetics-Sounds. We further test the effect of penalty factor γ in Equation 4 by varying it from $[0, 2, 5, 10]$ and observe that it has little effect on the final performance with the best performance at $\gamma = 10$ in all our experiments.

Effectiveness of LSTM. We investigate the effectiveness of LSTM in modeling video causality on the RGB + Audio experiment and observe that directly predicting a choice
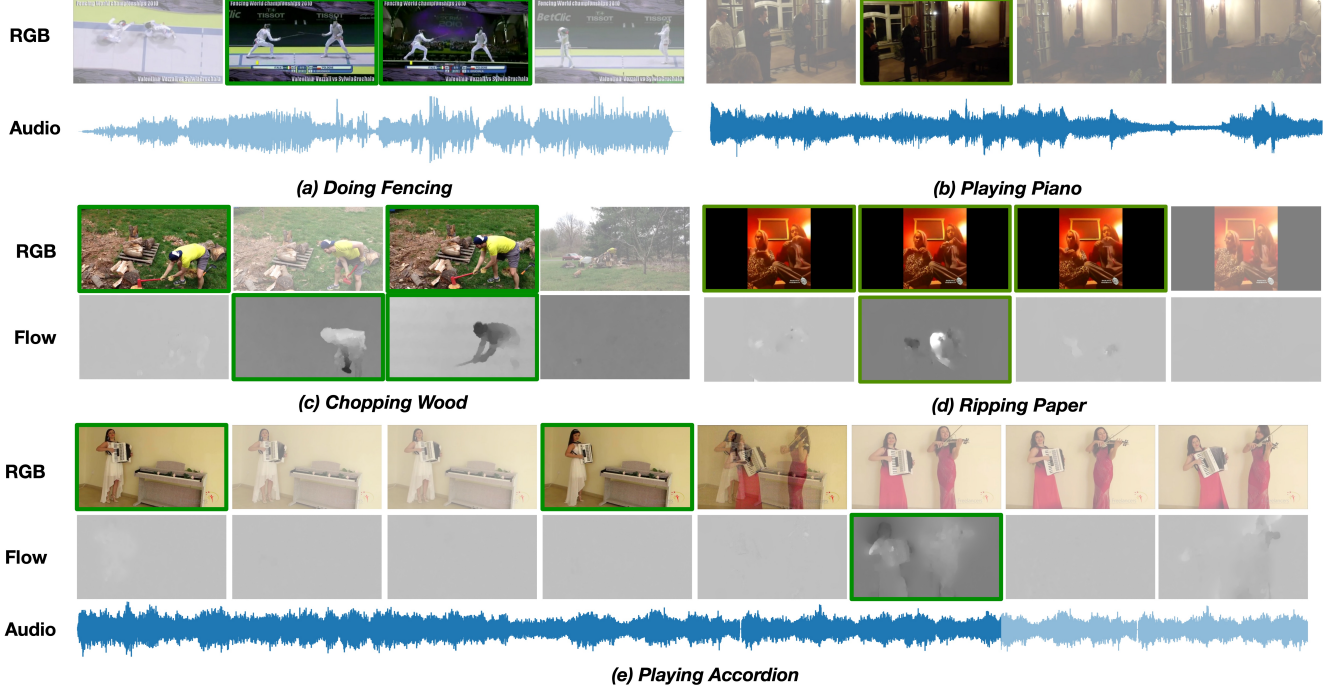
Figure 3: **Qualitative examples showing the effectiveness of `AdaMML` in selecting the right modalities per video segment (marked by green borders).** (a, b) RGB + Audio: `AdaMML` selects RGB stream for second and third segments in (a) while skips irrelevant audio coming from the reporter and background song. Similarly in (b), it is able to select RGB modality for only one segment while selecting the entire audio stream as the action can be easily recognized with audio (Playing Piano). (c, d) RGB + Flow: Our approach selects flow stream only when it is informative for the action, e.g., second and third segments in (c) and only second segment in (d). (e) RGB + Flow + Audio: `AdaMML` selects audio for most of the segments (not for last two segments as the audio is not clear with the mixing of sound from both instruments) while selecting flow only for the sixth segment where the motion related to the action is clearly visible. Best viewed in color.

via a single fully-connected layer (i.e., by removing LSTM from the policy network) decreases the video accuracy from 88.17% to 86.82% on Kinetics-Sounds. This confirms that LSTM is critical for good performance as it makes the policy network aware of all useful information seen so far.

Sampling Hyperparameters. We test the effect of temperature (Equation 3) in RGB+Audio experiment on Kinetics-Sounds dataset by varying it from [0, 0.5, 5, 10] and observe that higher values (5, 10) show better performance (by 0.5%-0.7%) than lower ones. So, we start at a high temperature (set to 5 in all our experiments) and anneal it to a small non-zero value, as in [23]. Similarly, we also vary the annealing factor from [0, 0.5, 0.965] and notice that setting it to 0.965 leads to the best accuracy of 88.17% while 0 leads to an accuracy of 87.40% on Kinetics-Sounds.

4.4. Qualitative Results

Figure 3 shows the selected modalities using our approach on different cases ((a, b) RGB + Audio, (c, d) RGB + Flow, and (e) RGB + Flow + Audio). As seen from Figure 3.(a), our approach is able to select RGB modality for the segments that are more informative of the action and skip the audio stream as audio in that video is irrelevant to the action

"fencing" (majority of audio comes from the reporter and background song). Similarly in Figure 3.(b), it is able to select RGB modality for only one segment while selecting the entire audio stream as the action can be easily recognized with audio ("Playing Piano"). Overall, we observe that `AdaMML` focuses on the right modalities to use per segment for correctly classifying videos while taking efficiency into account (e.g., in Figure 3.(e), it mainly focus on audio for most of the segments while selecting RGB only for two informative segments and flow stream for the sixth segment for recognizing the action "Playing Accordion").

5. Conclusion

In this paper, we present `AdaMML`, a novel and differentiable approach for adaptively determining what modalities to use per segment per instance for efficient video recognition. In particular, we trained a multi-modal policy network to predict these decisions with the goal of achieving both competitive accuracy and efficiency. We efficiently train the policy network jointly with the recognition model using standard back-propagation. We demonstrate the effectiveness of our proposed approach on four standard datasets, outperforming several competing methods.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 2

[2] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. 2, 4

[3] John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017. 2

[4] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015. 2

[5] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 2

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2, 4

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

[8] Chun-Fu Richard Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. Deep analysis of cnn-based spatio-temporal representations for action recognition. In *CVPR*, 2021. 2

[9] Zhourong Chen, Yang Li, Samy Bengio, and Si Si. You look twice: Gaternet for dynamic filter selection in cnns. In *CVPR*, 2019. 2

[10] Nieves Crasto, Philippe Weinzaepfel, Karteek Alahari, and Cordelia Schmid. Mars: Motion-augmented rgb stream for action recognition. In *CVPR*, 2019. 2

[11] 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

[12] 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. 2

[13] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 2

[14] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *CVPR*, 2017. 2

[15] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013. 2

[16] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Dynamic zoom-in network for fast object detection in large images. In *CVPR*, 2018. 3

[17] Gao, Ruohan and Oh, Tae-Hyun, and Grauman, Kristen and Torresani, Lorenzo. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020. 2, 4, 6, 7

[18] Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. 2

[19] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *CVPR*, 2019. 3

[20] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018. 2

[21] Weizhe Hua, Yuan Zhou, Christopher M De Sa, Zhiru Zhang, and G Edward Suh. Channel gating neural networks. In *NeurIPS*, 2019. 3

[22] Noureldien Hussein, Mihir Jain, and Babak Ehteshami Bejnordi. Timegate: Conditional gating of segments in long-range activities. *arXiv preprint arXiv:2004.01808*, 2020. 2

[23] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 1, 3, 4, 5, 8

[24] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *TPAMI*, 2017. 2, 4

[25] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2, 4

[26] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019. 2, 5

[27] Douwe Kiela, Edouard Grave, Armand Joulin, and Tomas Mikolov. Efficient large-scale multi-modal classification. In *AAAI*, 2018. 2

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[29] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 2

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

[31] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion feature network: Fixed motion filter for action recognition. In *ECCV*, 2018. 2

[32] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 2

[33] Mason McGill and Pietro Perona. Deciding how to decide: Dynamic routing in artificial neural networks. In *ICML*, 2017. 2

[34] 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. 2

[35] Yue Meng, Rameswar Panda, Chung-Ching Lin, Prasanna Sattigeri, Leonid Karlinsky, Kate Saenko, Aude Oliva, and Rogerio Feris. Adafuse: Adaptive temporal fusion network for efficient action recognition. In *ICLR*, 2021. 2

[36] Mahyar Najibi, Bharat Singh, and Larry S Davis. Autofocus: Efficient multi-scale inference. In *ICCV*, 2019. 3

[37] Bowen Pan, Rameswar Panda, Camilo Fosco, Chung-Ching Lin, Alex Andonian, Yue Meng, Kate Saenko, Aude Oliva, and Rogerio Feris. Va-red2: Video adaptive redundancy reduction. In *ICLR*, 2021. 2

[38] Juan-Manuel Pérez-Rúa, Valentin Vielzeuf, Stéphane Pateux, Moez Baccouche, and Frédéric Jurie. Mfas: Multimodal fusion architecture search. In *CVPR*, 2019. 2

[39] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Tiny video networks. *arXiv preprint arXiv:1910.06961*, 2019. 2

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 3, 5

[41] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, pages 568–576, 2014. 2

[42] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NeurIPS*, 2013. 2

[43] 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*, 2018. 2

[44] Ximeng Sun, Rameswar Panda, and Rogerio Feris. Adashare: Learning what to share for efficient deep multi-task learning. In *NeurIPS*, 2020. 5

[45] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013. 5

[46] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 4

[47] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 7

[48] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

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

[50] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *ECCV*, 2018. 2

[51] 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. 2, 5

[52] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal networks hard? In *CVPR*, 2020. 2

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2

[54] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *ECCV*, 2018. 2

[55] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992. 4

[56] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *CVPR*, 2019. 5

[57] 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. 2, 4

[58] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *CVPR*, 2018. 2, 4

[59] 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. 2, 4, 6

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

[61] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016. 1, 2, 6

[62] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, 2018. 2

[63] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 2

[64] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017. 2