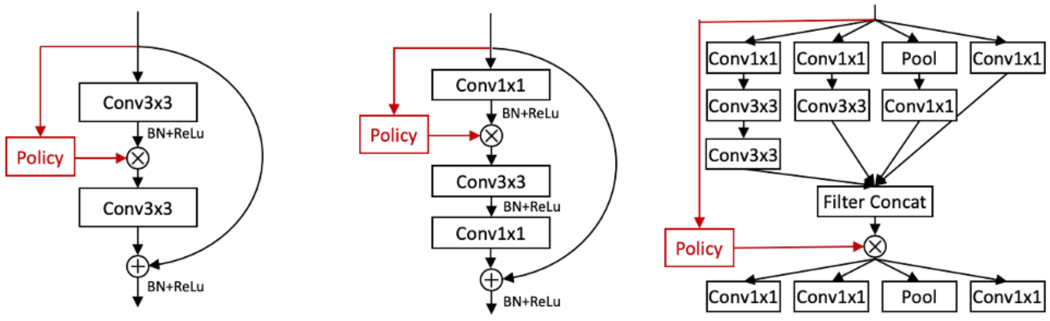
ADAFUSE: ADAPTIVE TEMPORAL FUSION NETWORK FOR EFFICIENT ACTION RECOGNITION (SUPPLEMENTARY MATERIAL)

A IMPLEMENTATION DETAILS



(a) ResNet-BasicBlock (b) ResNet-BottleNeck (c) BN-Inception

Figure 1: Detailed implementations under different architectures.

We apply adaptive temporal fusion for TSN model using ResNet18 (AdaFuse$^{\text{TSN}}_{\text{R18}}$), ResNet50 (AdaFuse$^{\text{TSN}}_{\text{R50}}$) and BN-Inception (AdaFuse$^{\text{TSN}}_{\text{Inc}}$) backbones. For the ResNet50 backbone, besides implementing the TSN model, we also apply the TSM model and explore two variants of settings (AdaFuse$^{\text{TSM}}_{\text{R50}}$, AdaFuse$^{\text{TSM+Last}}_{\text{R50}}$).

ResNet consists of a stem block and a stack of residual blocks in same topology. Each residual block contains two (for BasicBlock used in ResNet18) or three (for BottleNeck block used in ResNet50) convolution layers and other operations (residual operator, BatchNorm and ReLU), as shown in Figure 1 (a) & (b). We adopt adaptive temporal fusion in all the residual blocks. Specifically, we insert a policy network between the first and the second convolution layers in each residual block. The input feature for the policy network is from the input of each block. Locally, each policy network decides the channels of feature maps to compute in the first convolution layer and the channels to fuse for the second convolution layer, hence saves the computation budget.

BN-Inception network contains a sequence of building blocks where each of them contains a set of transformations, as shown in Figure 1 (c). At the end of each block, a "Filter Concat" operation is used to generate the output feature. We apply adaptive temporal fusion between adjacent blocks. The policy network receives the input from the input of the previous building block, decides the necessary computation for the previous building block and the channels to fuse for the next building blocks, hence achieves the computation efficiency.

We further try two variants using ResNet50 on top of the TSM model. Temporal shift is adopted at the beginning of each residual block. ResNet50 contains 3, 4, 6 and 3 BottleNeck blocks for the 1st, 2nd, 3rd and 4th stages respectively. In AdaFuse$^{\text{TSM}}_{\text{R50}}$, we apply adaptive temporal fusion in all the 16 blocks, whereas in AdaFuse$^{\text{TSM+Last}}_{\text{R50}}$, we only adopt in the last 3 blocks. Experimental results can be found in Table 3 in the main paper.

B QUALITATIVE ANALYSIS

Figure 2 shows more qualitative results that $\text{AdaFuse}^{\text{TSN}}_{\text{R50}}$ predicts on Something V1 & V2, Jester and Mini-Kinetics datasets. We only present 3 frames from each video sample. On top of each sample, we show the ground truth label and relative computation budgets in percentage. In general, $\text{AdaFuse}^{\text{TSN}}_{\text{R50}}$ saves the computation greatly for examples that contain clear appearance or actions with less motion.

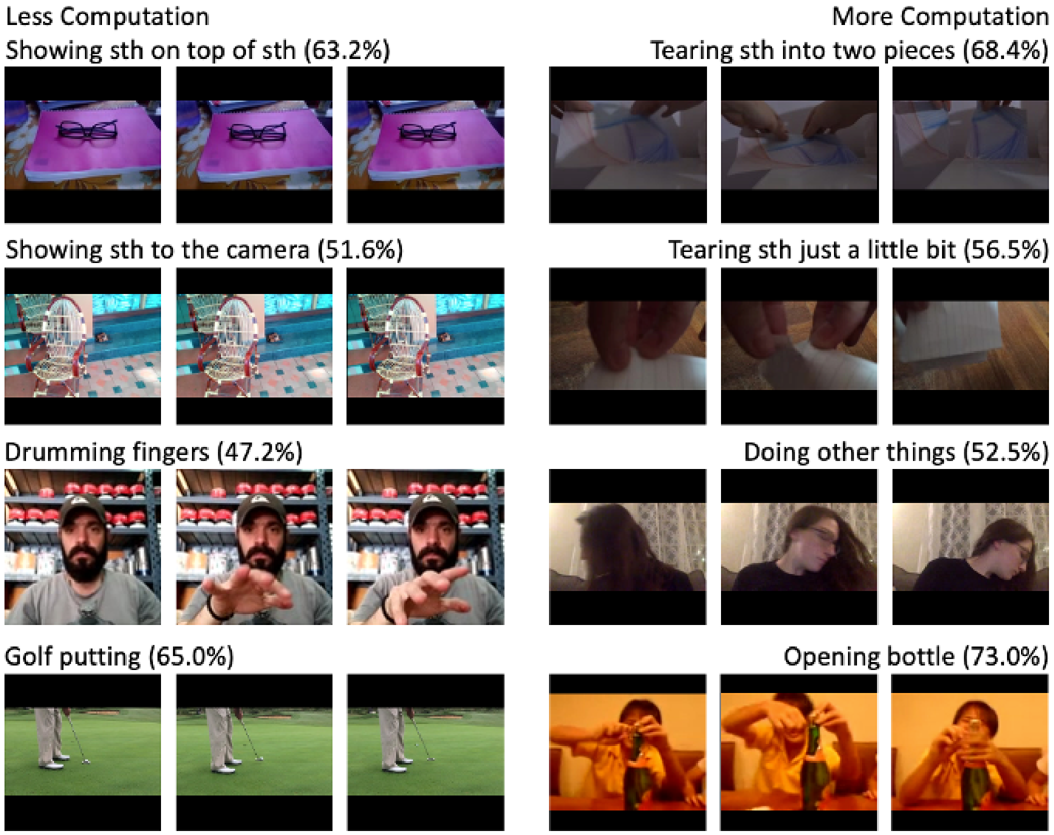

Less Computation
More Computation
Showing sth on top of sth (63.2%)
Tearing sth into two pieces (68.4%)
Showing sth to the camera (51.6%)
Tearing sth just a little bit (56.5%)
Drumming fingers (47.2%)
Doing other things (52.5%)
Golf putting (65.0%)
Opening bottle (73.0%)

Figure 2: Qualitative results on Something V1 & V2, Jester and Mini-Kinetics dataset. We only present 3 frames from each video sample. On top of each sample, we show the ground truth label and relative computation budgets in percentage. $\text{AdaFuse}^{\text{TSN}}_{\text{R50}}$ can save the computation greatly for examples which contain clear appearance or actions with less motion. Best viewed in color.