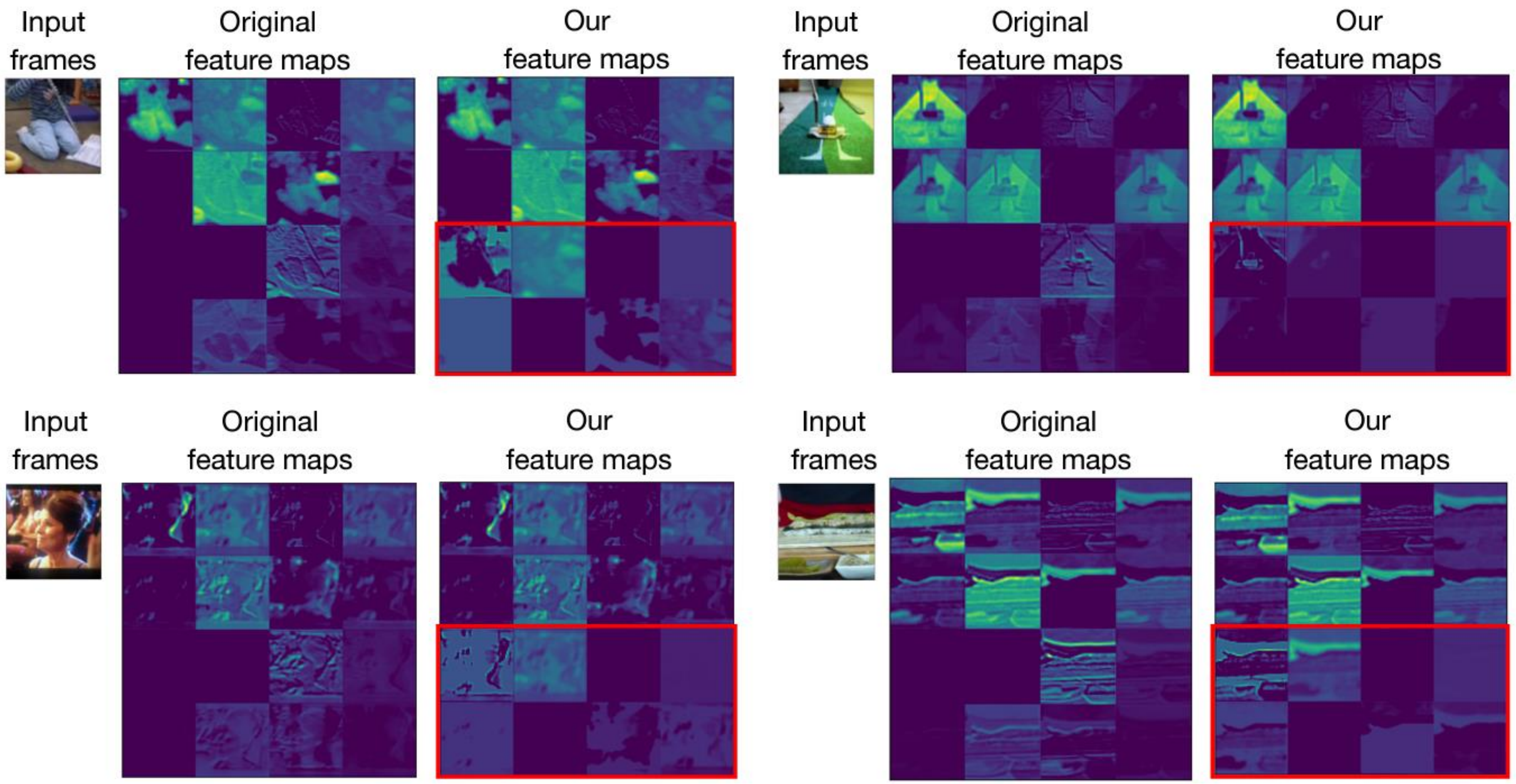
VA-RED²: Video Adaptive Redundancy Reduction (ICLR 2021)

Bowen Pan[1], Rameswar Panda[2], Camilo Fosco[1], Chung-Ching Lin[3],
Alex Andonian[1], Yue Meng[2], Kate Saenko[2,4], Aude Oliva[1,2], Rogerio Feris[2]
[1] MIT CSAIL, [2] MIT-IBM Watson AI Lab, [3] Microsoft, [4] Boston University



발표: 정채연

Motivation

Feature Redundancy in Well-Trained DNN (GhostNet)

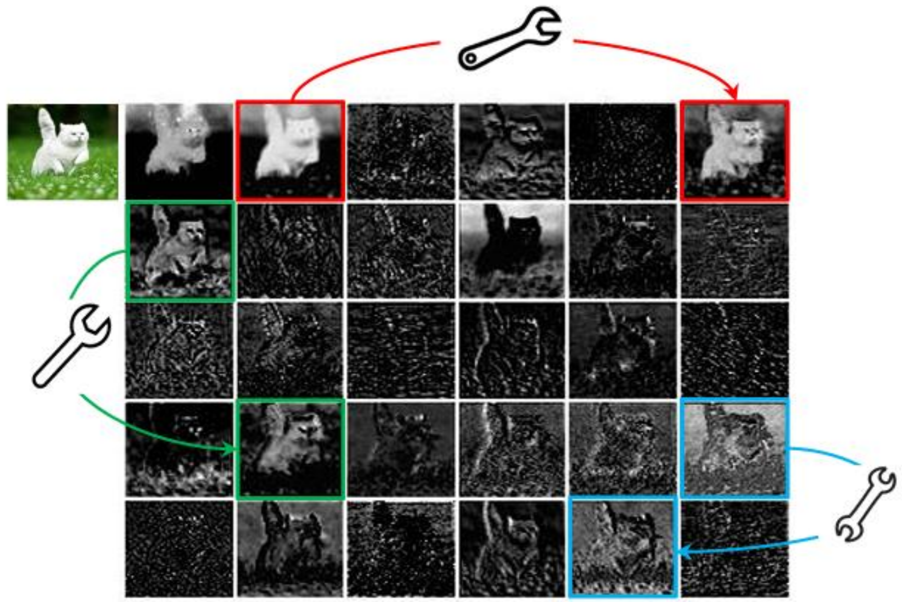
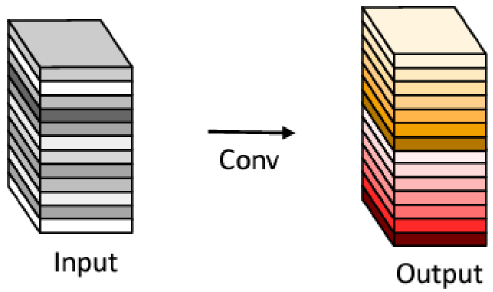
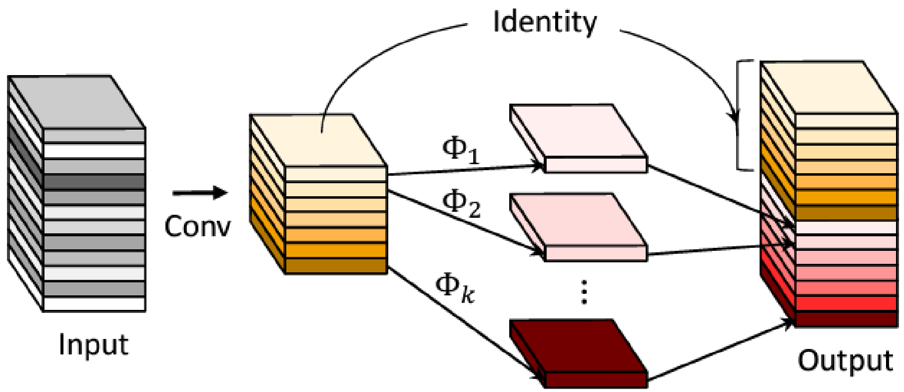


Figure 1. Visualization of some feature maps generated by the first residual group in ResNet-50, where three similar feature map pair examples are annotated with boxes of the same color. One feature map in the pair can be approximately obtained by transforming the other one through cheap operations (denoted by spanners).



(a) The convolutional layer.

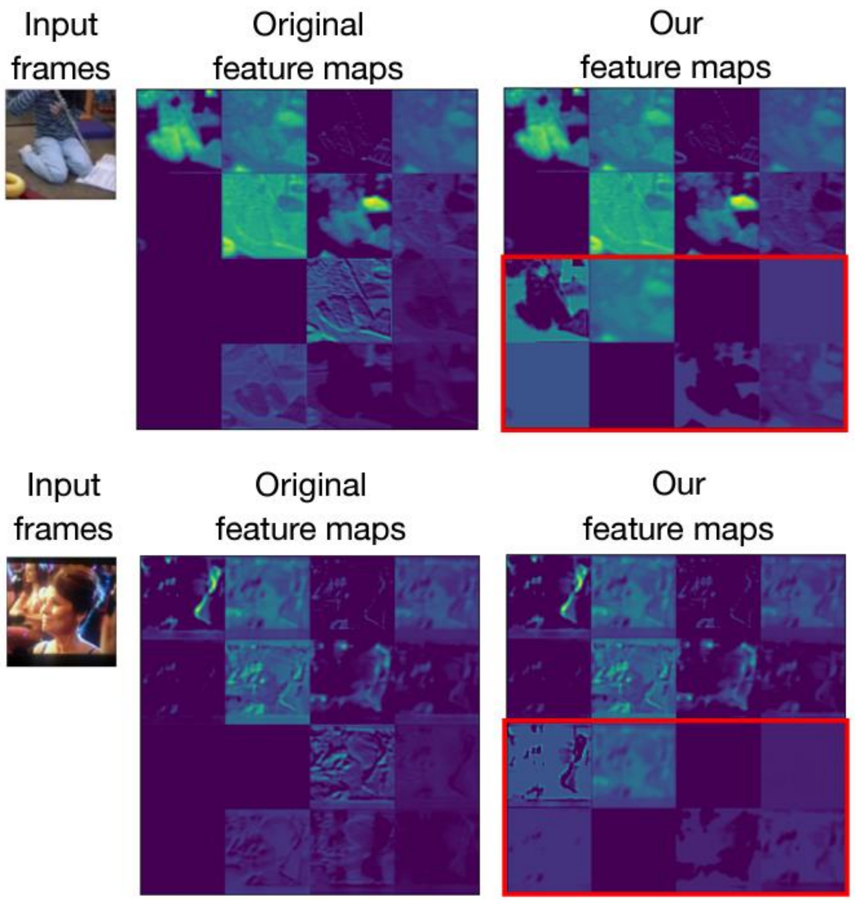


(b) The Ghost module.

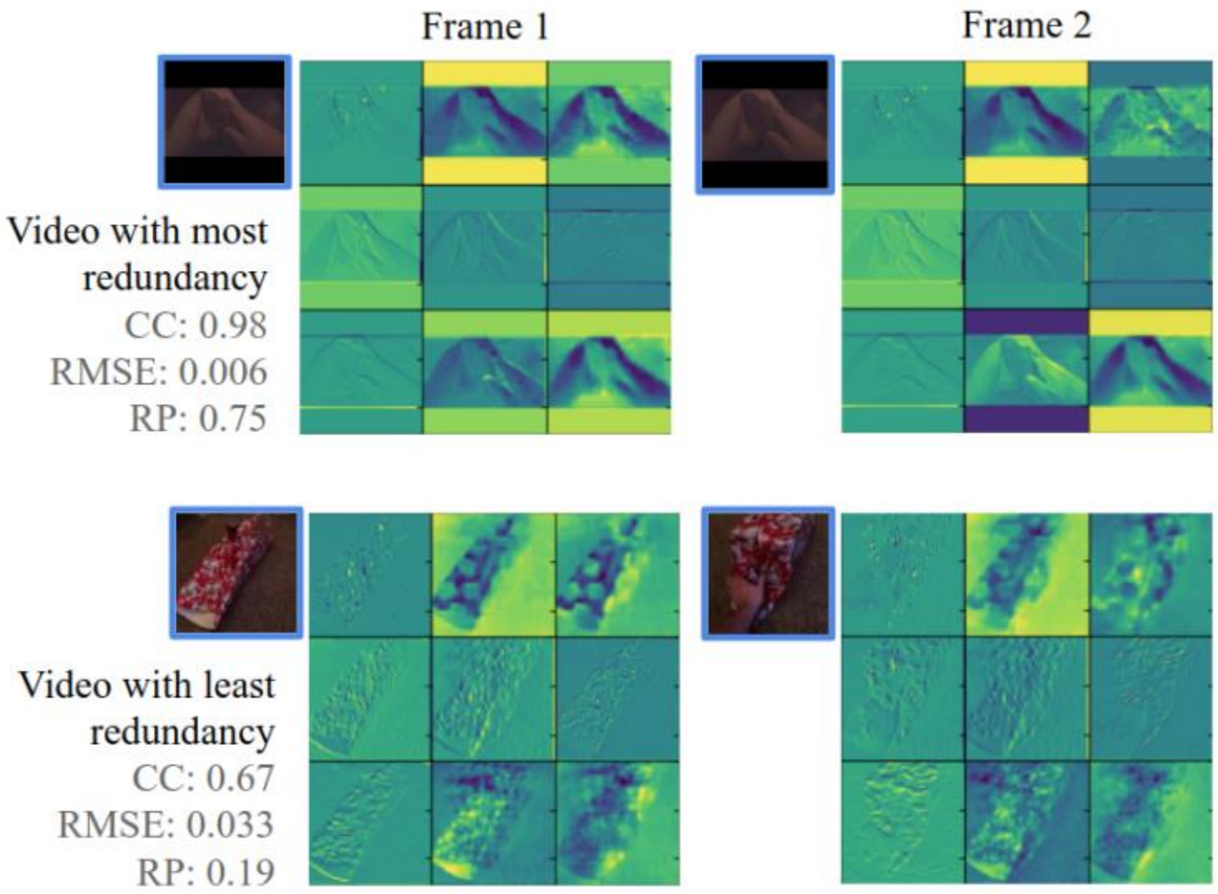
GhostNet: More Features from Cheap Operations (CVPR 2020) [link]

Motivation

Channel and Temporal Redundancy in Videos



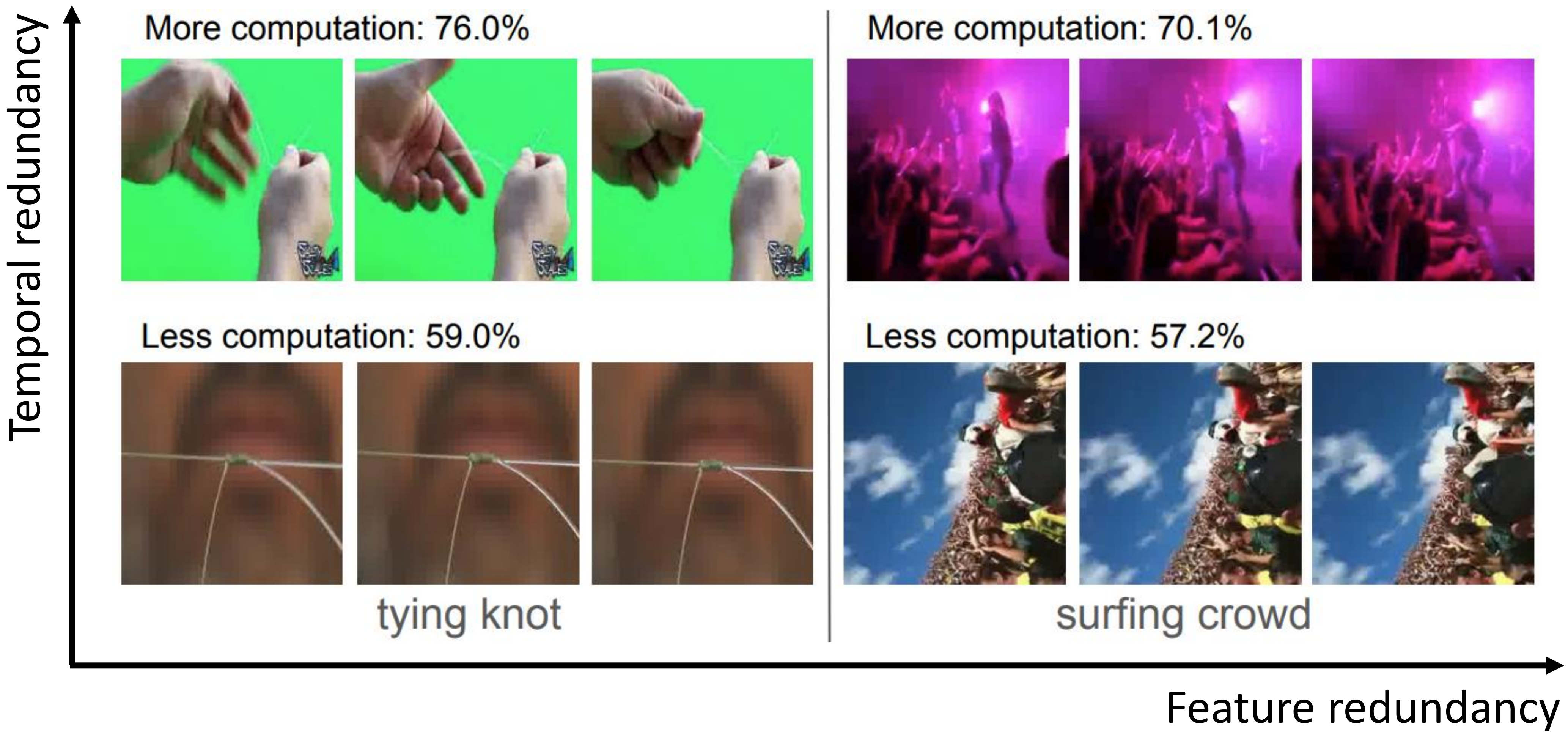
Channel redundancy



Temporal redundancy

Motivation

Input-Dependent Channel and Temporal Redundancy



Related Work

1) ~~A novel **input-dependent adaptive framework** for efficient video recognition.~~

---

2) ~~An adaptive policy jointly learned with the network weights in a fully differentiable way.~~

---

3) ~~Our approach is **model-agnostic** and can be applied to any backbones to reduce feature redundancy in both time and channel domains.~~

---

4) ~~Striking results of VA-RED2 over baselines using various datasets.~~

---

5) ~~A **generalization** of our framework to video action recognition, spatio-temporal localization, and semantic segmentation tasks, achieving promising results while offering significant reduction in computation over competing methods.vv~~

Contributions of **VA-RED^2**

1) A novel **input-dependent adaptive framework** for efficient video recognition.

2) An adaptive policy jointly learned with the network weights in a fully differentiable way.

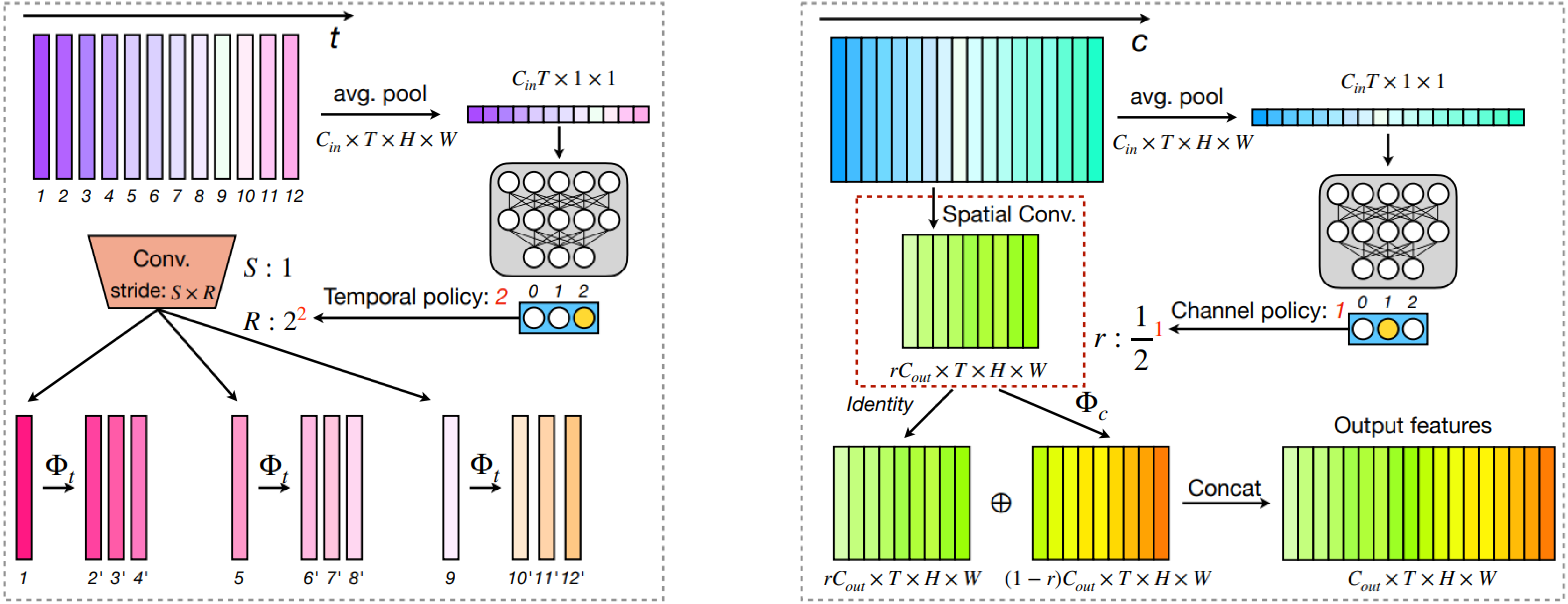
3) Our approach is **model-agnostic** and can be applied to any backbones to reduce feature redundancy in both time and channel domains.

4) Striking results of VA-RED2 over baselines using various datasets.

5) A **generalization** of our framework to video action recognition, spatio-temporal localization, and semantic segmentation tasks, achieving promising results while offering significant reduction in computation over competing methods.

Method

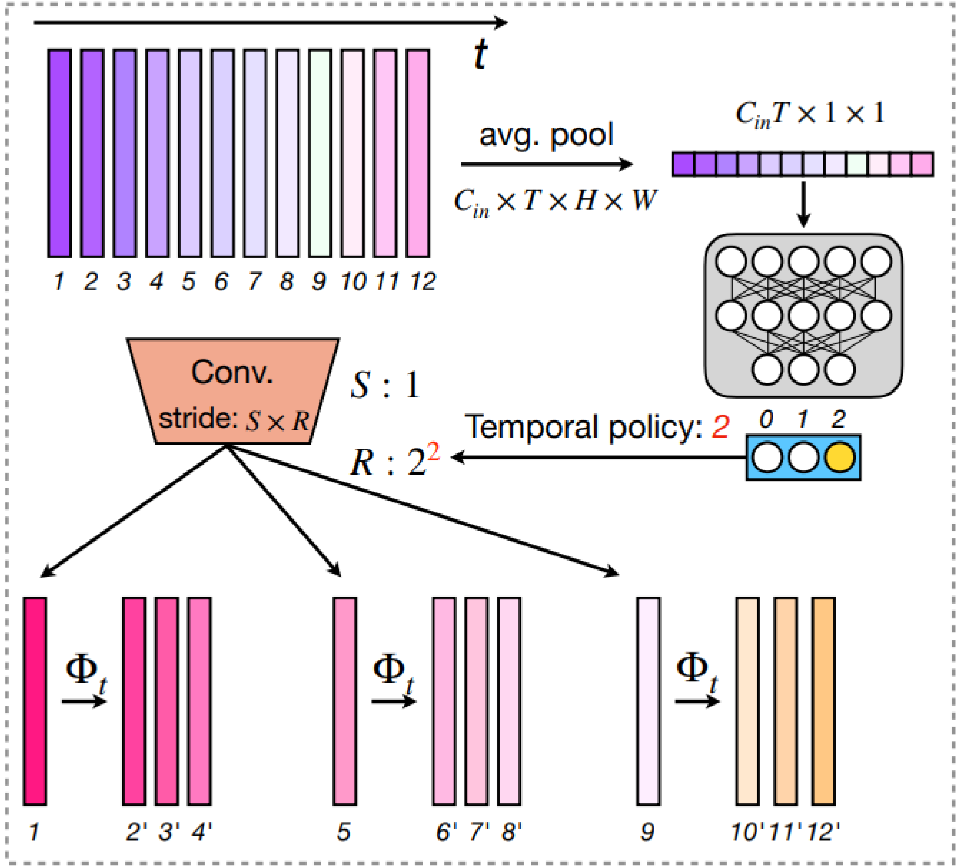
Video Adaptive Redundancy Reduction



(a) Temporal-wise dynamic convolution (b) Channel-wise dynamic convolution

Method

Video Adaptive Temporal Redundancy Reduction



(a) *Temporal-wise dynamic convolution*

$$Y_l[j + iR] = \begin{cases} \Phi^t_{i,j}(Y_l'[i]) & \text{if } j \in \{1, ..., R-1\} \\ Y_l'[i] & \text{if } j = 0 \end{cases},$$

$C_{out} \times T_o/R \times H_o \times W_o$

$$i \in \{0, 1, ..., T_o/R - 1\}$$

$$R = 2^{\underline{p_l}(X_l)[0]}$$

Soft modulation gate

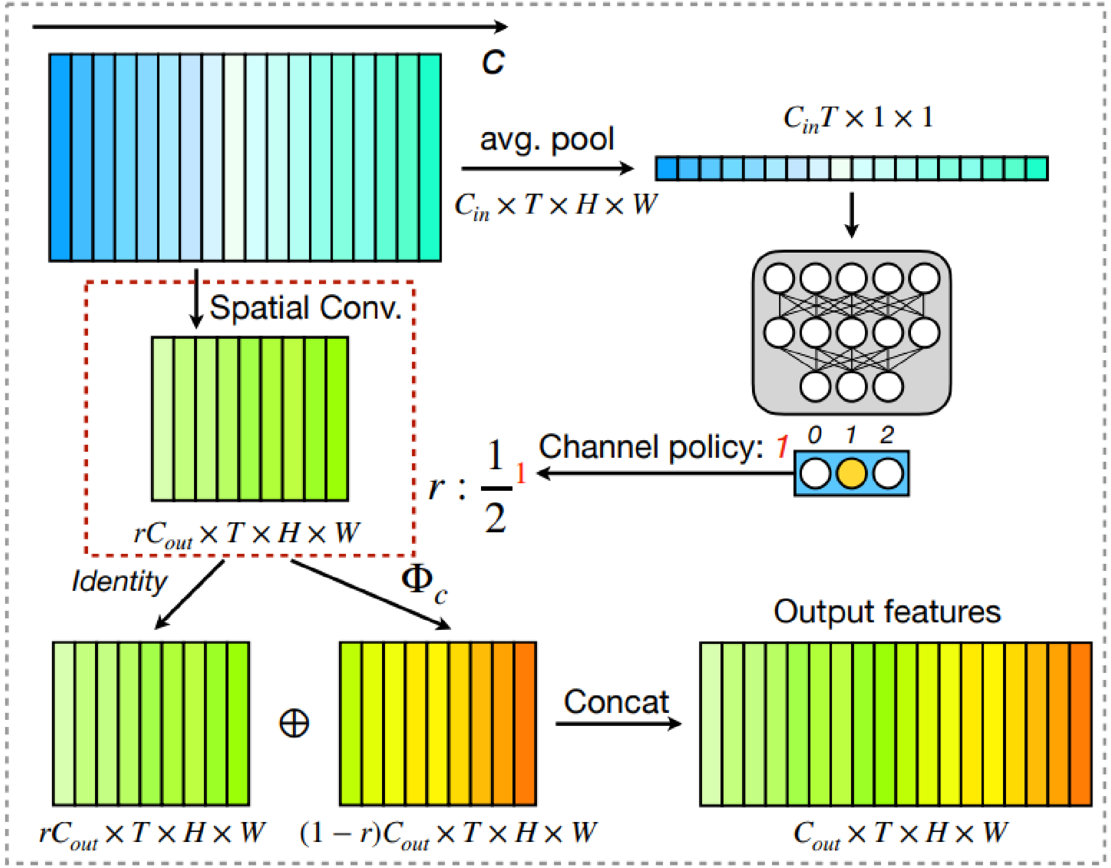
Method

Video Adaptive Channel Redundancy Reduction

$$Y_l = [Y_l', \Phi^c(Y_l')]$$

$rC_{out} \times T_o \times H_o \times W_o$

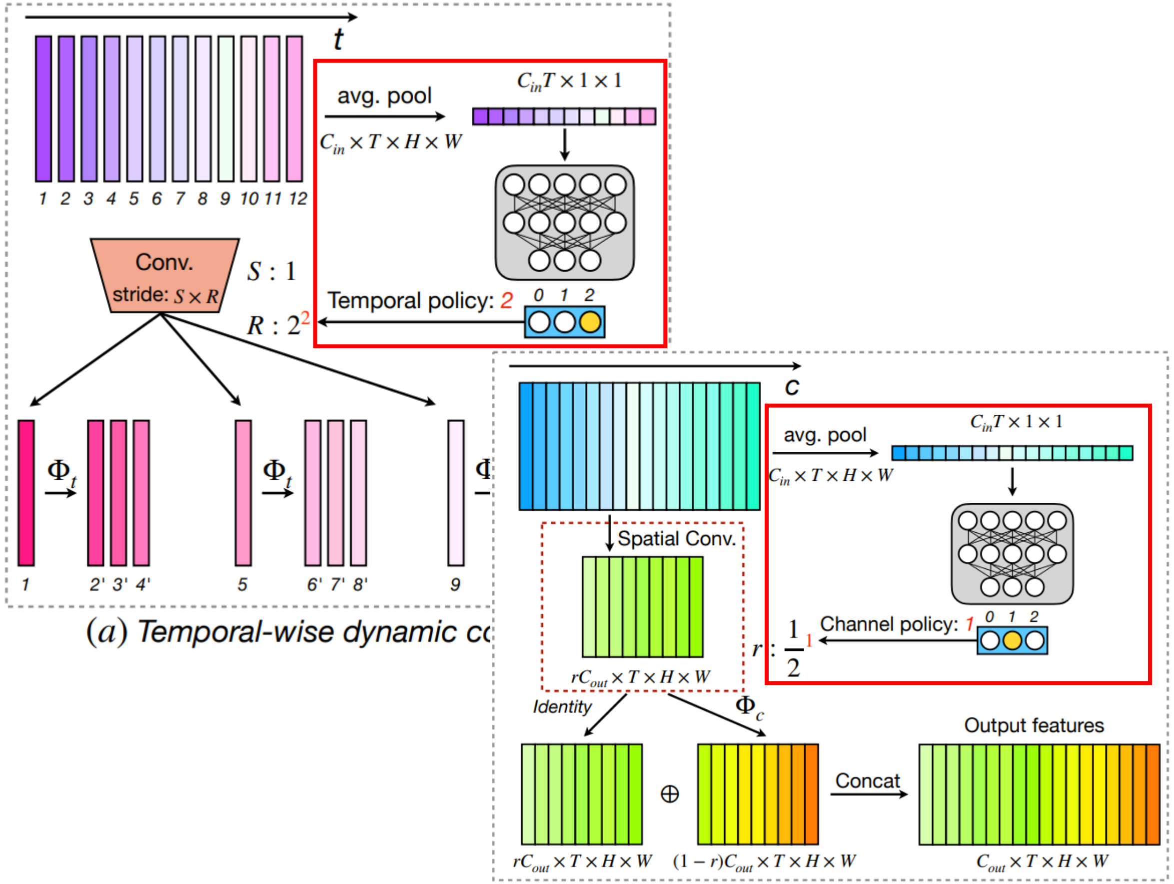
$$r = (\tfrac{1}{2})^{p_l(X_l)[1]}$$



(b) Channel-wise dynamic convolution

Method

Soft Modulation Gate for Differentiable Optimization



(a) Temporal-wise dynamic co

(b) Channel-wise dynamic convolution

$V_t^l \in R^{S_t}$ and $V_c^l \in R^{S_c}$

$$[V_t^l, V_c^l] = p_l(X_l)$$
$$= \phi(\mathcal{F}(\omega_{p,2}, \delta(\mathcal{N}(\mathcal{F}(\omega_{p,1}, G(X_l))))) + \beta_p^l)$$

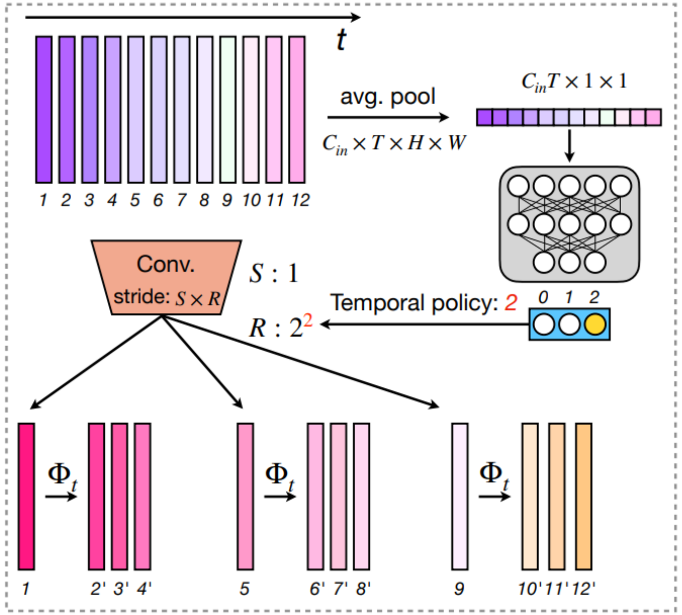
$\max(\tanh(\cdot), 0)$

$$Y_c^l = \sum_{i=1}^{S_c} V_c^l[i] \cdot f_l^c(X_l, r = (\frac{1}{2})^{(i-1)})$$

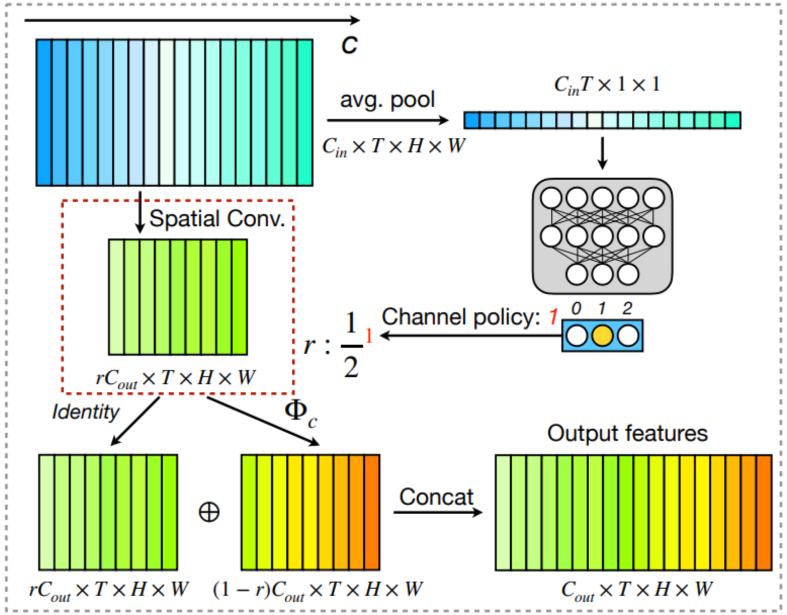
$$Y_l = \sum_{j=1}^{S_t} V_t^l[j] \cdot f_l^t(Y_c^l, R = 2^{(j-1)})$$

Method

Computation Cost of Video Adaptive Redundancy Reduction



(a) *Temporal-wise dynamic convolution*



(b) *Channel-wise dynamic convolution*

$$\mathcal{C}(f_l^t) = \frac{1}{R} \cdot \mathcal{C}(f_l) + \sum_{i,j} \mathcal{C}(\Phi_{i,j}^t) \approx \frac{1}{R} \cdot \mathcal{C}(f_l)$$

$$\mathcal{C}(f_l^{t,c}) \approx \frac{r}{R} \cdot \mathcal{C}(f_l)$$

Method

Losses

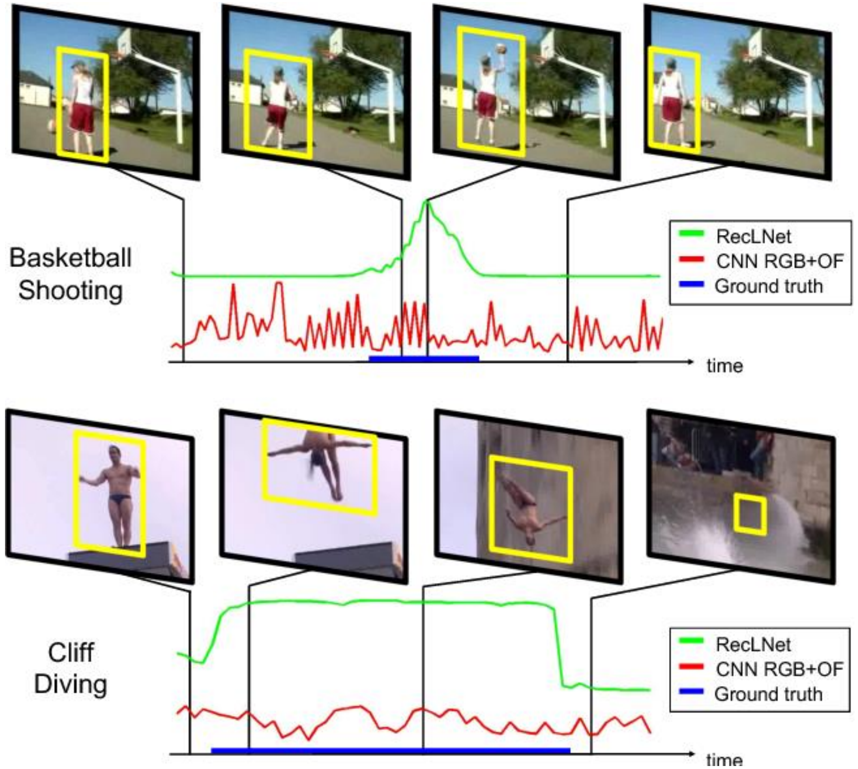
$$\mathcal{L} = \mathcal{L}_a + \lambda_e \mathcal{L}_e$$

$\mathcal{L}_a$: accuracy loss

$$\mathcal{L}_e = (\mu_0 \sum_{l=1}^{L} \frac{\mathcal{C}(f_l)}{\sum_{k=1}^{L} \mathcal{C}(f_k)} \cdot \frac{r_l^s}{R_l^s})^2, \mu_0 = \begin{cases} 1 & \text{if correct} \\ 0 & \text{otherwise} \end{cases}$$

Experiments

Video action recognition



Spatio-temporal localization

Semantic segmentation

Experiments

Video Action Recognition

length	sp.	GFLOPs_{Avg}	GFLOPs_{Max}	GFLOPs_{Min}	avg speed	clip-1	video-1	video-5
8	✗	27.7	27.7	27.7	192.1	56.4	66.8	86.8
	2	20.0(−28%)	22.1(−20%)	18.0(−35%)	**205.5**	57.7	**68.0**	**87.4**
	3	21.6(−22%)	23.2(−16%)	19.8(−29%)	201.4	**58.2**	67.7	**87.4**
16	✗	55.2	55.2	55.2	97.1	57.5	67.5	87.1
	2	40.4(−27%)	43.2(−22%)	36.6(−34%)	**108.7**	**60.6**	**70.0**	**88.7**
32	✗	110.5	110.5	110.5	49.6	60.5	69.4	88.2
	2	79.3(−28%)	89.5(−19%)	72.4(−34%)	**53.4**	**63.3**	**72.3**	**89.7**

Model: R(2+1) D

Dataset: Mini-Kinetics-200

Experiments

Video Action Recognition

Table 2: **Action recognition results on Mini-Kinetics-200.** We set the search space as 2 and train all the models with 16 frames. The metric speed uses *clip/second* as the unit.

Model	Dy.	GFLOPs	Speed	clip-1	video-1
R(2+1)D	✗	55.2	97.1	57.5	67.5
	✓	40.4	**108.7**	**60.6**	**70.0**
I3D	✗	56.0	116.4	59.7	68.3
	✓	26.5	**141.7**	**62.2**	**71.1**
X3D	✗	6.20	169.4	**66.5**	**72.2**
	✓	5.03	**178.2**	65.5	72.1

Table 3: **Action recognition results with Temporal Pyramid Network (TPN) on Mini-Kinetics-200.** TPN-8f and TPN-16f indicate that we use 8 frames and 16 frames as input to the model respectively.

Model	Dy.	GFLOPs	clip-1	video-1
TPN-8f	✗	28.5	58.9	67.2
	✓	21.5	**59.2**	**68.8**
TPN-16f	✗	56.8	59.8	68.5
	✓	41.5	**60.8**	**70.6**

Experiments

Video Action Recognition

Table 4: **Comparison with CorrNet (Wang et al., 2020) and AR-Net (Meng et al., 2020) on Mini-Kinetics-200.** We set the search space as 2 and train all the models with 16 frames.

Model	Dy.	GFLOPS	clip-1	video-1	Method	Params	GFLOPs	clip-1
CorrNet	✗	60.8	59.9	68.2	AR-Net	63.0M	44.8	67.2
	✓	**45.5**	**60.4**	**70.0**	VA-RED^2	**23.9M**	**43.4**	**68.3**

Table 5: **Action recognition results on Kinetics-400.** We set the search space as 2, meaning models can choose to compute all feature maps or $\frac{1}{2}$ of them both on temporal and channel-wise convolutions.

Model	Dy.	16-frame					32-frame				
		GFLOPs	speed	clip-1	video-1	video-5	GFLOPs	speed	clip-1	video-1	video-5
R(2+1)D	✗	55.2	97.1	57.3	65.6	86.3	110.5	49.6	61.5	69.0	88.6
	✓	40.3	**105.9**	**58.4**	**67.6**	**87.6**	80.7	**53.0**	61.5	**70.0**	**88.9**
I3D	✗	56.0	116.4	55.1	66.5	86.7	112.0	57.6	57.2	64.9	86.5
	✓	32.1	**140.7**	**58.6**	**67.1**	**87.2**	64.3	**71.7**	**61.0**	**68.6**	**88.4**
X3D	✗	6.42	169.4	63.2	70.6	90.0	[X3D-M is designed for 16 frames]				
	✗	5.38	**177.6**	**65.3**	**72.4**	**90.7**					

Table 6: **Action recognition results on Moments-In-Time.** We set the search space as 2, i.e., models can choose to compute all feature maps or $\frac{1}{2}$ of them both on temporal and channel-wise convolutions. The speed uses *clip/second* as the unit.

Model	Dy.	GFLOPs	speed	clip-1	video-1
R(2+1)D	✗	55.2	97.1	27.0	28.8
	✓	42.5	**105.5**	**27.3**	**30.1**
I3D	✗	56.0	116.4	25.7	26.8
	✓	32.1	**140.7**	**26.3**	**28.5**
X3D	✗	6.20	169.4	24.8	24.8
	✓	5.21	**177.4**	**26.7**	**27.7**

Experiments

Spatio-Temporal Action Localization

Table 8: **Action localization results on J-HMDB.** We set the search space as 2 for dynamic models. The speed uses $clip/second$ as the unit.

Model	Dy.	GFLOPs	speed	mAP	Recall	Classif.
I3D	✗	43.9	141.1	44.8	**67.3**	87.2
	✓	21.3	**167.4**	**47.2**	65.6	**91.1**
X3D	✗	5.75	176.3	47.9	65.2	**93.2**
	✓	4.85	**184.6**	**50.0**	**65.8**	93.0

Experiments

Semantic Segmentation

Table 11: **VA-RED^2 on semantic segmentation.** We choose dilated ResNet-18 as our backbone architecture and set the search space as 2. Models are trained for 100K iterations with batch size of 8.

Model	Original model		Channel-wise reduction using VA-RED^2			
	GFLOPs	mean IoU	GFLOPs$_{avg}$	GFLOPs$_{max}$	GFLOPs$_{min}$	mean IoU
Dilated ResNet-18	10.6	31.2%	**7.8**	9.1	7.3	**31.3%**

Experiments

Comparison with Other Pruning Methods & Effect of Efficiency Loss

Table 7: **Comparison with network pruning methods.** We choose R(2+1)D on Mini-Kinetics-200 dataset with different number of input frames. Numbers in green/blue quantitatively show how much our proposed method is better/worse than these pruning methods.

Method	Frames	GFLOPs	clip-1
Weight-level	8	19.9 **(-0.1)**	54.5 **(+3.2)**
	16	40.3 **(-0.1)**	57.7 **(+2.9)**
	32	79.6 **(-0.3)**	59.6 **(+3.7)**
CGNet	8	23.8 **(+3.8)**	56.2 **(+1.5)**
	16	47.6 **(+7.2)**	57.8 **(+2.8)**
	32	95.3 **(+16.0)**	61.8 **(+1.5)**

Table 9: **Effect of efficiency loss on Kinetics-400.** *Eff.* denotes the efficiency loss.

Model	*Eff.*	GFLOPs	clip-1	video-1
R(2+1)D	No	49.8	57.9	66.7
	Yes	40.3	**58.4**	**67.6**
I3D	No	56.0	58.0	66.5
	Yes	32.1	**58.6**	**67.1**

Experiments

Ablation Study

Table 10: **Ablation experiments on dynamic modeling along temporal and channel dimensions.** We choose R(2+1)D-18 on Mini-Kinetics-200 and set the search space to 2 in all the dynamic models.

Dy. Temp.	Dy. Chan.	8-frame				16-frame			
		GFLOPs	speed	clip-1	video-1	GFLOPs	speed	clip-1	video-1
✗	✗	27.7	192.1	56.4	66.8	55.2	97.1	57.5	67.5
✓	✗	23.5	198.6	57.1	66.8	46.1	105.0	58.6	67.6
✗	✓	22.7	196.5	57.0	66.7	46.3	102.0	59.2	68.3
✓	✓	20.0	**205.5**	**57.7**	**68.0**	40.4	**108.7**	**60.6**	**70.0**

Experiments

Visualization and Analysis

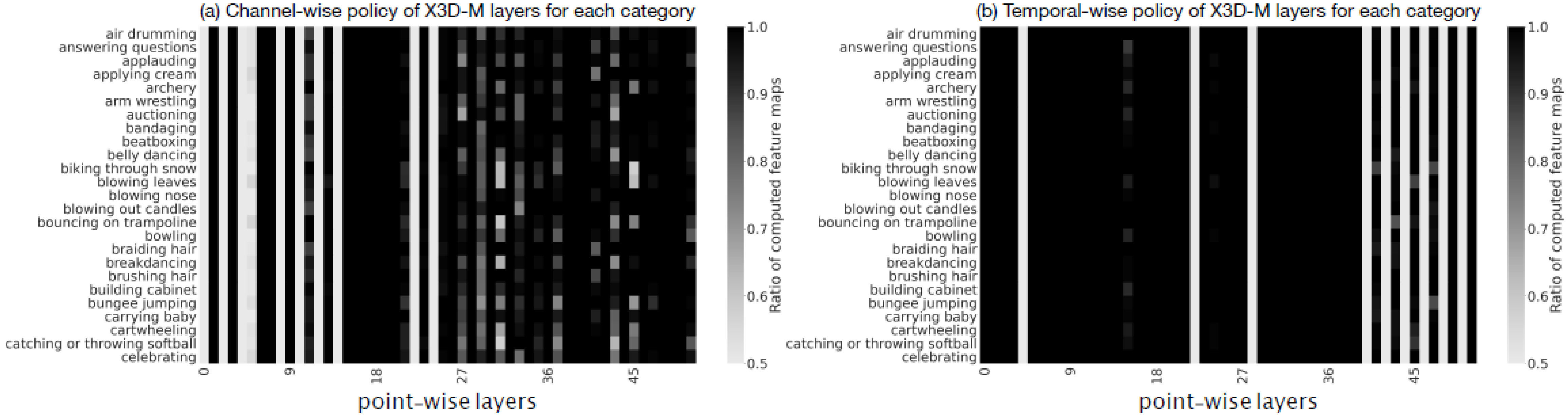


Figure 3: **Ratio of computed feature per layer and class on Mini-Kinetics-200 dataset.** We pick the first 25 classes of Mini-Kinetics-200 and visualize the per-block policy of X3D-M on each class. Lighter color means fewer feature maps are computed while darker color represents more feature maps are computed.

Experiments

Visualization and Analysis

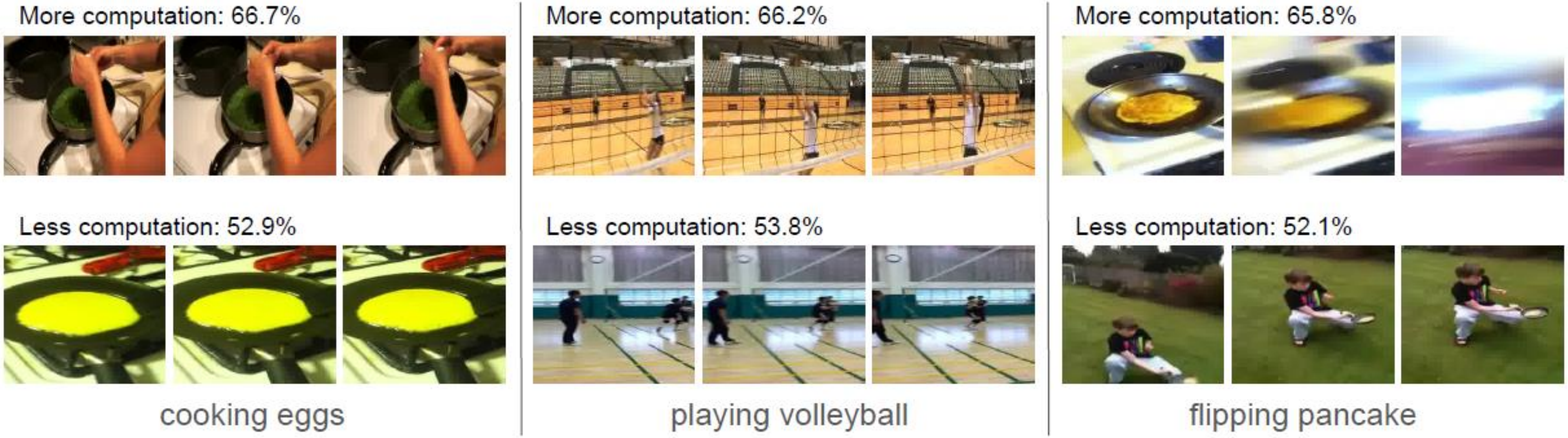


Figure 4: **Validation video clips from Mini-Kinetics-200.** For each category, we plot two input video clips which consume the most and the least computational cost respectively. We infer these video clips with 8-frame dynamic R(2+1)D-18 model trained on Mini-Kinetics-200 and the percentage indicates the ratio of actual computational cost of 2D convolution to that of the original fixed model. Best viewed in color.

Experiments

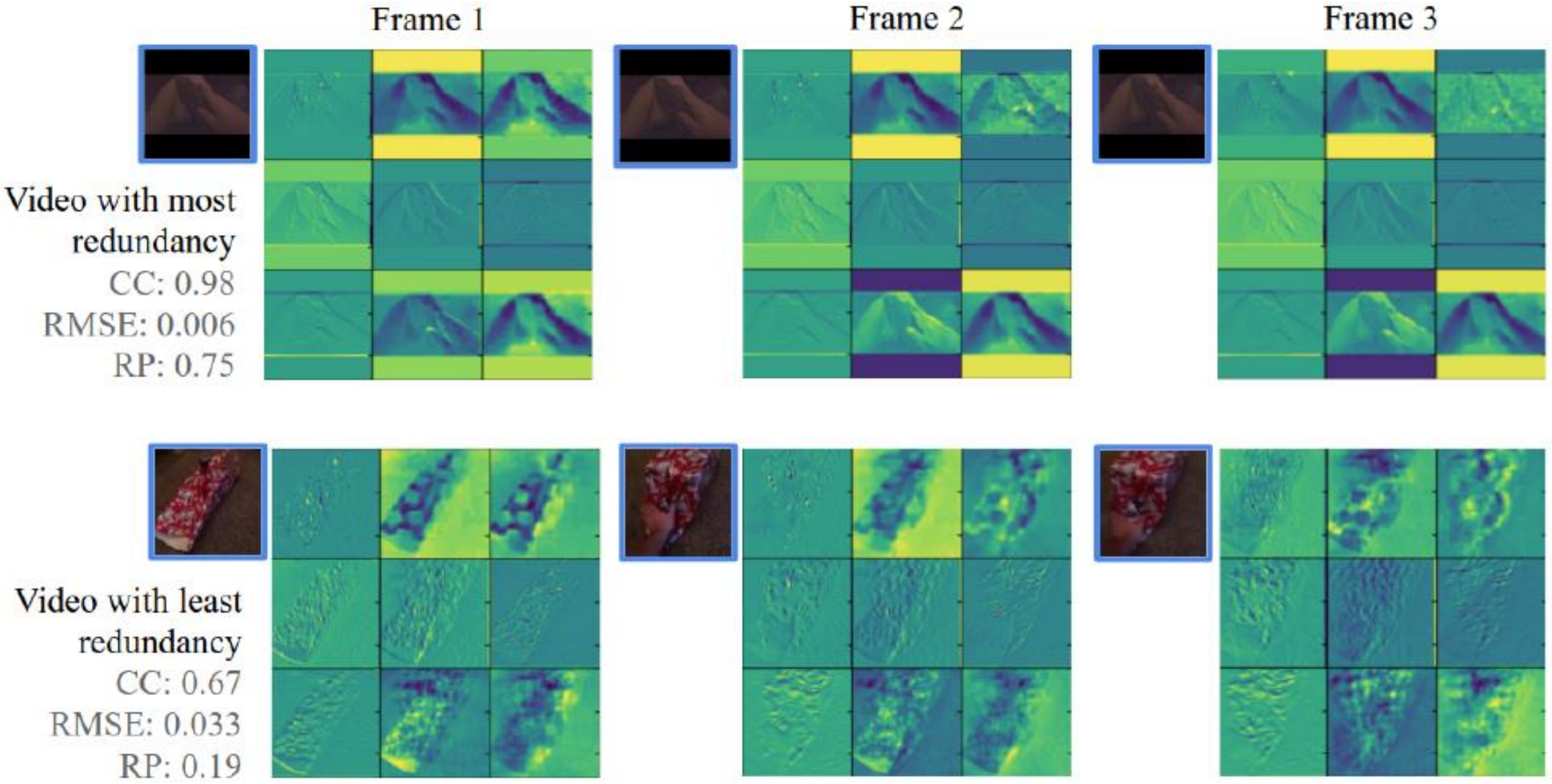
Redundancy Experiments

Dataset	Model	Dimension	CC	RMSE	RP
Moments-In-Time	I3D	Temporal	0.77	0.083	0.62
	I3D	Channel	0.71	0.112	0.48
	R(2+1)D	Temporal	0.73	0.108	0.49
	R(2+1)D	Channel	0.68	0.122	0.43
Kinetics-400	I3D	Temporal	0.81	0.074	0.68
	I3D	Channel	0.76	0.091	0.61
	R(2+1)D	Temporal	0.78	0.081	0.64
	R(2+1)D	Channel	0.73	0.088	0.58

CC: Correlation Coefficient
RMSE: Root Mean Square Error
RP: Redundancy Proportion

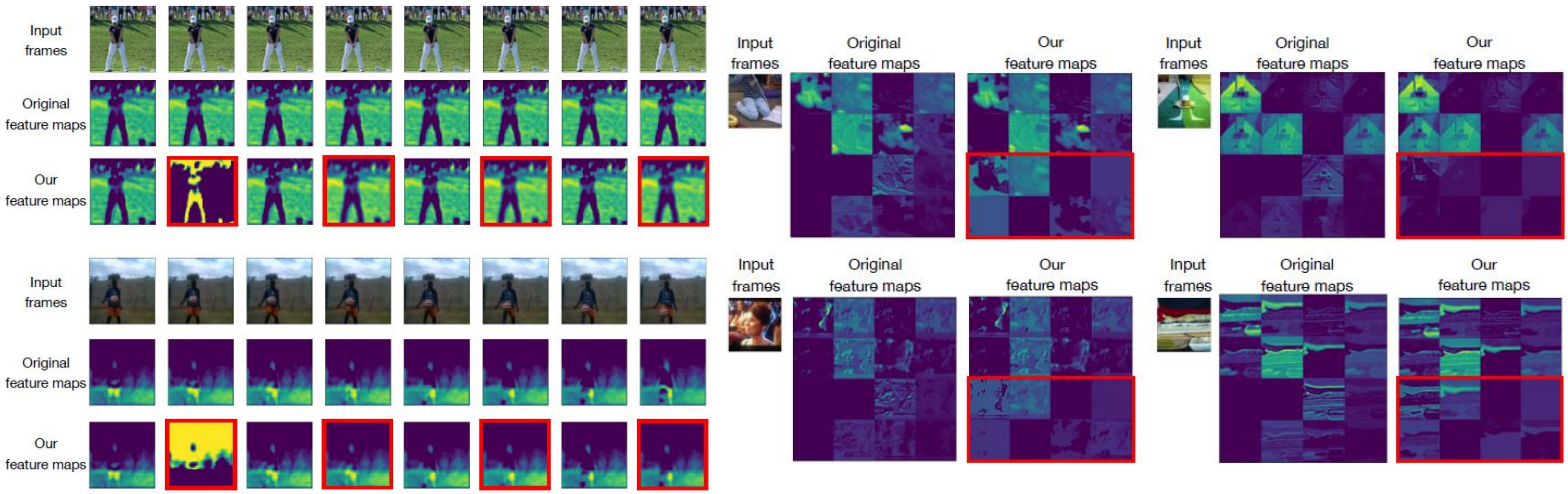
Experiments

Redundancy Experiments



Experiments

Feature Map Visualization



Thank you.