

# Supplementary Materials for “SOAP: Enhancing ... Few-Shot Action Recognition”

Wenbo Huang  
Southeast University  
Nanjing, China  
wenbohuang1002@outlook.com

Jinghui Zhang\*  
Southeast University  
Nanjing, China  
jhzhzhang@seu.edu.cn

Xuwei Qian  
Southeast University  
Nanjing, China  
xuwei.qian@seu.edu.cn

Zhen Wu  
Southeast University  
Nanjing, China  
zhen-wu@seu.edu.cn

Meng Wang  
Tongji University  
Shanghai, China  
wangmengsd@outlook.com

Lei Zhang  
Nanjing Normal University  
Nanjing, China  
leizhang@njjnu.edu.cn

In the Supplementary Material, we provide:

- Partial Generalization Study.
- Robustness Study.
- Additional CAM Visualizations.
- Configuration of hyperparameters.
- Pseudo-code for a better understanding.

## A GENERALIZATION STUDY

### A.1 Generalization on More Complex Tasks

In a range of real-world scenarios, tasks are often more complex than a mere 5-way classification. Recognizing this complexity, we seek to enhance the challenge of these tasks by augmenting the number of ways (classes). Experiments are conducted on SthSthV2 and Kinetics. From Figure I, performance of various methods [1–3] decreases with increasing complexity. Although task complexity hinders performance boosts, SOAP-Net consistently outperforms other methods. Surprisingly, SOAP-Net decays less than other methods on complex tasks. The above results show better generalization of SOAP on more complex tasks. It also indicates spatio-temporal relation and comprehensive motion information support better generalization.

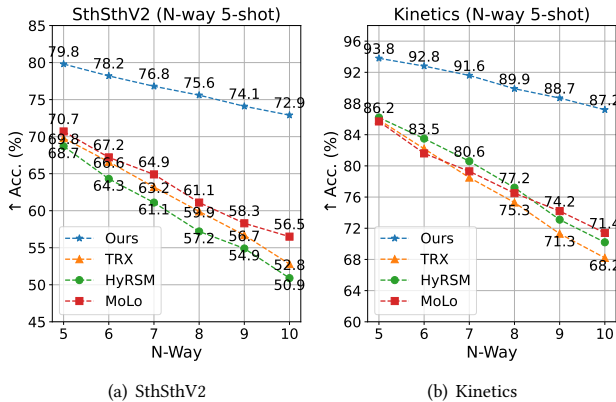


Figure I: Performance (↑ Acc. %) of More Complex Tasks.

\*Corresponding authors: Jinghui Zhang

### A.2 Generalization on Any-Shot Setting

In real-world applications, it is often challenging to ensure that every task has an equal number of samples. To create a more authentic testing environment for assessing the generalization ability of SOAP, we utilize a shot number in the range of 1 to 5, effectively establishing an any-shot setting for our experiments. Results with 95% confidence intervals are shown in Table I. SOAP-Net outperforms three recent methods under the any-shot setting, and shows similar performance in normal 1-shot or 5-shot settings. The confidence intervals indicate that our SOAP-Net exhibits more stable performance under the more realistic evaluation scenario. The above experiments underscore the potential of SOAP in practical applications.

Table I: Any-shot Performance (↑ Acc. %) comparison.

Methods	SthSthV2	Kinetics
TRX	61.3 (±0.5)	79.3 (±0.4)
HyRSM	62.2 (±0.6)	79.5 (±0.5)
MoLo	63.7 (±0.4)	80.9 (±0.3)
SOAP-Net	<b>70.2 (±0.3)</b>	<b>87.4 (±0.2)</b>

## B ROBUSTNESS STUDY

### B.1 Robustness on Sample-Level Noise

In a more authentic testing environment, noise inevitably occurs during sample collection, and removing it incurs additional costs. Specifically, a particular class might be mixed with samples from other classes. As shown in Figure II, directly replacing one or more samples with noise within a few-shot task during inference is called sample-level noise. Evaluating FSAR methods with sample-level noise simulates more realistic sample scenarios. For a clear demonstration, we conduct 10-shot experiments and reveal the results in Table II. In general, performance decreases with increasing noise. For every 10% increase in the sample-level noise ratio, performance drops by about 4%. Although the overall trend remains regardless of the method, we observe a distinction between SOAP-Net and other methods. The performance decrease with the sample-level noise ratio for SOAP-Net is about 2%, much less than others. The stable performance against sample-level noise highlights superior robustness of our SOAP.

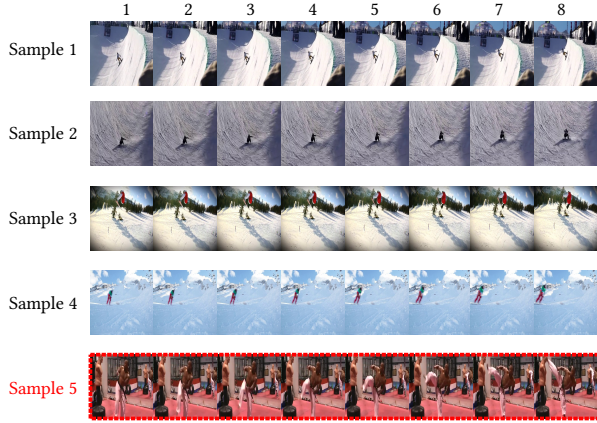


Figure II: The 5-shot example of “snowboarding”, where the sample-level noise is highlighted by red box. Sample-level noise means directly replacing a pure sample with noise within a few-shot task.

Table II: Evaluation ( $\uparrow$  Acc. %) with Sample-Level Noise.

Datasets	Methods	Sample-Level Noise Ratio				
		0%	10%	20%	30%	40%
SthSthV2	TRX	74.4	69.9	66.3	62.6	57.2
	HyRSM	73.6	68.9	64.3	60.9	54.9
	MoLo	75.3	71.6	67.1	64.0	59.8
	SOAP-Net	<b>84.9</b>	<b>83.2</b>	<b>81.6</b>	<b>79.1</b>	<b>77.9</b>
Kinetics	TRX	90.1	86.9	82.4	77.8	73.2
	HyRSM	89.7	85.6	81.4	77.2	72.9
	MoLo	90.6	86.2	82.3	76.8	70.1
	SOAP-Net	<b>96.4</b>	<b>94.1</b>	<b>91.9</b>	<b>89.6</b>	<b>87.7</b>

## B.2 Robustness on Frame-Level Noise

Due to the uncertainty of video recorder, it is not guaranteed that the view of all frames is aligned with the moving subject. In some undesirable cases, multiple irrelevant frames are mixed in. Under such conditions, higher requirements are placed on the FSAR method. As shown in Figure III, we replace some pure frames with irrelevant frames from other action classes during inference. We define this as frame-level noise. The 5-way 10-shot setting is the same as the evaluation, and the results are detailed in Table III. Compared to sample-level noise, the negative impact of frame-level noise is small, with a similar overall trend. SOAP-Net outperforms other methods under any frame-level noise number. We find the impact of frame noise on TRX and HyRSM is greater than on MoLo and our SOAP-Net. The commonality of MoLo and SOAP-Net is both apply motion information, which plays a vital role in FSAR. Our proposed method is less affected, proving comprehensive motion information provides better robustness to frame-level noise.

## C ADDITIONAL CAM VISUALIZATIONS

As a complement to the CAM visualization in the main paper, we provide additional examples in Figure IV. For the example “snowboarding”, each frame is sampled without much background variation. However, the high fluency makes motion information weak,

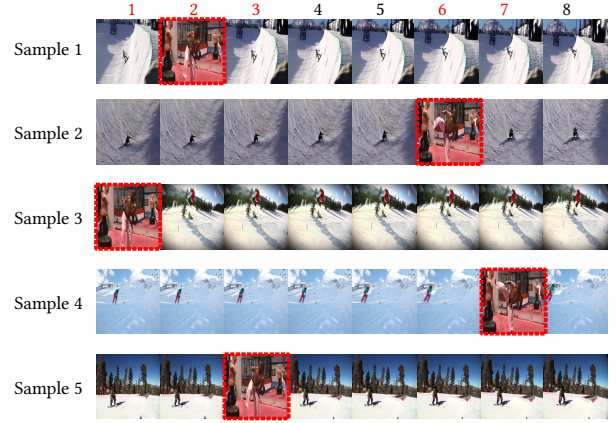


Figure III: The 5-shot example of “snowboarding”, where the frame-level noise is highlighted by red box. Frame-level noise means introducing noise frames.

Table III: Evaluation ( $\uparrow$  Acc. %) with Frame-Level Noise.

Datasets	Methods	Frame-Level Noise Number				
		0	1	2	3	4
SthSthV2	TRX	74.4	71.5	68.4	65.1	62.2
	HyRSM	73.6	70.2	66.9	64.2	61.1
	MoLo	75.3	74.1	72.2	70.1	68.8
	SOAP-Net	<b>84.9</b>	<b>84.2</b>	<b>83.6</b>	<b>82.2</b>	<b>80.9</b>
Kinetics	TRX	90.1	87.3	84.8	81.2	79.0
	HyRSM	89.7	86.5	84.4	82.2	80.1
	MoLo	90.6	89.5	88.3	87.0	85.8
	SOAP-Net	<b>96.4</b>	<b>95.5</b>	<b>94.3</b>	<b>93.1</b>	<b>91.7</b>

and the model attention mistakenly focuses on the background. Fortunately, the model with SOAP concentrates more on the skier instead of the outlying background. Due to disordered scenery like driftwood and tourists, the example “diving cliff” is more complex than the first. In the second row, we can clearly see the model is disrupted and does not know where to focus. When motion information is highlighted, people diving the cliff is easier to find.

## D HYPERPARAMETER STUDY

Table IV illustrates the SOAP-Net structure in our implementation, as detailed in the main paper. The Conv3D size is  $3 \times 3 \times 3$  in 3DEM. In CWEM, the expand channel number  $C_r$  is set to 16, while Conv1D size is 3.  $O = \{1, 2, 3\}$  and the Conv2D<sup>M</sup> sizes are  $3 \times 3$  in HMEM. For a supplementary interpretation of implementation details, extensive experiments are conducted on individual modules with diverse configuration to determine the optimal hyperparameters.

### D.1 Hyperparameter Study of 3DEM

The size of the Conv3D plays a crucial role as the main hyperparameter in 3DEM. As a result, we only apply 3DEM to conduct few-shot experiments on two datasets and have reported the results in Table V. In most cases, the best performance appears in the  $3 \times 3 \times 3$  row. Based on these findings, we conclude that the  $3 \times 3 \times 3$  Conv3D is the optimal hyperparameter for 3DEM.

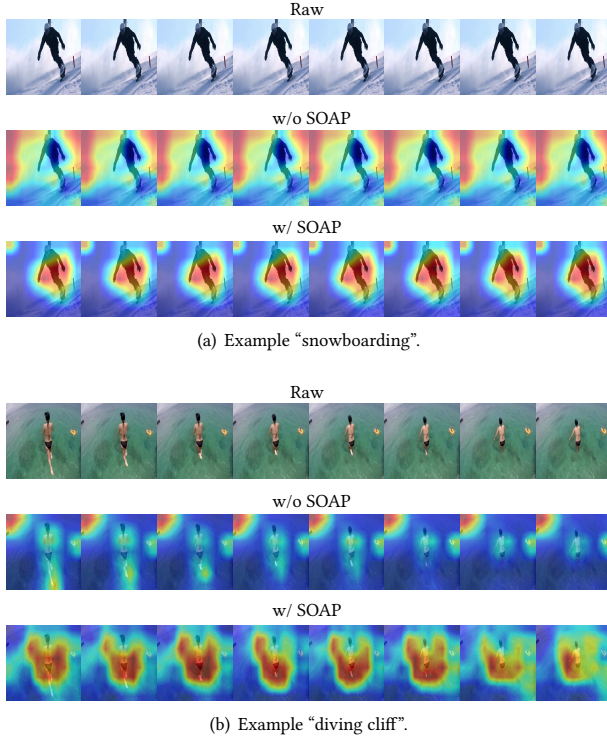


Figure IV: Additional CAM Visualizations.

Table IV: Structure of SOAP-Net. Notations are consistent with the main paper.

Modules / Operation	Input	Input Size	Output	Output Size
3DEM	$S^{ck}$	[8,3,224,224]	$\tilde{S}_1^{ck}$	[8,1,224,224]
	$Q$	[8,3,224,224]	$\tilde{Q}_1$	[8,1,224,224]
CWEM	$S^{ck}$	[8,3,224,224]	$\tilde{S}_2^{ck}$	[8,3,1,1]
	$Q$	[8,3,224,224]	$\tilde{Q}_2$	[8,3,1,1]
HMEM	$S^{ck}$	[8,3,224,224]	$\tilde{S}_3^{ck}$	[8,3,1,1]
	$Q$	[8,3,224,224]	$\tilde{Q}_3$	[8,3,1,1]
Eqn.(15)	$S^{ck}$	[8,3,224,224]	$\tilde{S}^{ck}$	[8,3,224,224]
	$S_1^{ck}$	[8,3,224,224]		
	$S_2^{ck}$	[8,3,224,224]		
	$S_3^{ck}$	[8,3,224,224]		
	$Q$	[8,3,224,224]	$\tilde{Q}$	[8,3,224,224]
Eqn.(16)	$Q_1$	[8,3,224,224]		
	$Q_2$	[8,3,224,224]		
	$Q_3$	[8,3,224,224]		
Eqn.(16)	$\tilde{S}^{ck}$	[8,3,224,224]	$S_f^{ck}$	[8,2048]
	$\tilde{Q}$	[8,3,224,224]	$Q_f$	[8,2048]

## D.2 Hyperparameter Study of CWEM

The size of Conv1D and the expand channel number  $C_r$  are essential primary hyperparameters. A well-matched combination can effectively calibrate temporal connections between each channel.

Table V: Hyperparameter Study ( $\uparrow$  Acc. %) of 3DEM.

Conv3D	SthSthV2		Kinetics	
	1-shot	5-shot	1-shot	5-shot
$1 \times 1 \times 1$	54.9	67.6	74.6	85.7
$3 \times 3 \times 3$	<b>55.6</b>	69.4	<b>76.8</b>	<b>86.6</b>
$5 \times 5 \times 5$	55.1	<b>69.5</b>	76.6	86.3
$7 \times 7 \times 7$	54.6	68.2	75.9	85.8

Therefore, after arranging and combining these two hyperparameters, we present all results using only CWEM in Table VI. Our findings indicate that Conv1D with sizes of 3 or 5 outperforms those with sizes of 1 or 7. Further experiments on  $C_r$  reveal that  $C_r = 16$  contributes to optimal performance in most cases. Taking a comprehensive consideration based on reported results, we conclude that Conv1D with a size of 3 and  $C_r = 16$  represents the best set of hyperparameters for CWEM.

Table VI: Hyperparameter Study ( $\uparrow$  Acc. %) of CWEM.

Conv1D	$C_r$	SthSthV2		Kinetics	
		1-shot	5-shot	1-shot	5-shot
1	8	54.5	67.3	74.2	85.3
	16	<b>54.7</b>	<b>67.6</b>	<b>74.4</b>	<b>85.6</b>
	32	54.6	67.4	<b>74.4</b>	85.4
	64	54.5	67.4	74.3	<b>85.6</b>
3	8	55.1	69.6	75.7	85.8
	16	<b>55.4</b>	<b>70.2</b>	<b>76.1</b>	<b>86.1</b>
	32	<b>55.4</b>	70.0	75.8	85.9
	64	55.2	69.7	75.7	85.5
5	8	54.8	69.1	75.2	85.5
	16	55.1	<b>69.3</b>	<b>75.7</b>	85.8
	32	<b>55.2</b>	<b>69.3</b>	75.5	85.6
	64	55.1	69.2	75.3	<b>85.9</b>
7	8	54.2	67.4	74.1	85.1
	16	<b>54.6</b>	<b>67.6</b>	74.2	85.5
	32	54.5	67.5	<b>74.3</b>	<b>85.6</b>
	64	54.3	67.3	74.2	85.2

## D.3 Hyperparameter Study of HMEM

Similar to CWEM, HMEM also has two hyperparameters: the size of Conv2D<sup>M</sup> and the design of  $\mathcal{O}$ . The former determines the visual receptive field size, while the latter determines the number of branches and frame tuple sizes. The results obtained using only HMEM through permutation and combination are presented in Table VII. Performance with  $3 \times 3$  and  $5 \times 5$  Conv2D<sup>M</sup> is slightly better than other settings. As indicated in the main paper, performance generally improves with more branches, but excessive branches can lead to degradation. Considering the  $\mathcal{O}$  design, it is confirmed that a configuration of  $3 \times 3$  Conv2D<sup>M</sup> with  $\mathcal{O} = \{1, 2, 3\}$  is the most suitable hyperparameter setting for HMEM.

## E PSEUDO-CODE

For a better understanding of HMEM, we provide a summary of the two main processes: the sliding window algorithm (Algorithm 1) and motion information calculation (Algorithm 2), using

**Table VII: Hyperparameter Study ( $\uparrow$  Acc. %) of HMEM.**

Conv2D <sup>M</sup>	O Design	SthSthV2		Kinetics	
		1-shot	5-shot	1-shot	5-shot
1 × 1	O = {3}	57.1	71.8	77.2	87.3
	O = {2, 3}	57.5	<b>72.1</b>	77.6	87.8
	O = {1, 2, 3}	<b>57.9</b>	<b>72.1</b>	<b>78.1</b>	<b>88.6</b>
	O = {1, 2, 3, 4}	57.3	71.4	77.1	87.2
3 × 3	O = {3}	57.6	71.4	77.7	88.1
	O = {2, 3}	57.8	71.9	78.2	88.7
	O = {1, 2, 3}	<b>58.3</b>	<b>72.3</b>	<b>78.5</b>	<b>88.9</b>
	O = {1, 2, 3, 4}	57.2	71.0	77.6	88.0
5 × 5	O = {3}	57.5	71.9	77.5	88.2
	O = {2, 3}	57.8	<b>72.2</b>	77.7	88.4
	O = {1, 2, 3}	<b>58.1</b>	72.1	<b>78.2</b>	<b>88.7</b>
	O = {1, 2, 3, 4}	57.1	71.6	77.2	87.7
7 × 7	O = {3}	57.6	70.4	76.2	87.3
	O = {2, 3}	<b>58.0</b>	70.8	76.6	87.5
	O = {1, 2, 3}	57.9	<b>71.1</b>	<b>77.3</b>	<b>88.1</b>
	O = {1, 2, 3, 4}	56.8	70.2	76.1	87.1

pseudocode. Both processes consist of one layer loops with a computational complexity of  $O(N_w)$  due to  $|S_t| = N_w$ . In order to comprehensively calculate motion information (Algorithm 3), the above two algorithms are included in a larger loop for hyperparameter  $O$  set. As a result, the computational complexity of the entire HMEM is  $O(|O| N_w)$ .

**Algorithm 1: Sliding window algorithm SW ( $\cdot, \cdot$ )**


---

**Input:**  $I = [I_1, \dots, I_F] \in \mathbb{R}^{F \times C \times H \times W}$ ,  $T$  ( $T \in O, T < F, T \in \mathbb{N}^*$ )  
**Output:**  $S_t$

```

1  $F \leftarrow \text{Shape}(I, 0); N_w \leftarrow F - T + 1;$ 
2  $S_t \leftarrow \emptyset;$  // Defining an empty list
3 for each  $i \in [0, N_w - 1]$  do
4    $S_t \leftarrow \text{Append}(S_t, I[:, i : i + T, \dots]);$ 
5 end
6 return  $S_t$ 

```

---

**Algorithm 2: Motion information calculation MIC ( $\cdot$ )**


---

**Input:**  $S_t$   
**Output:**  $M$

```

1  $M \leftarrow \emptyset;$  // Defining an empty list
2 for each  $i \in [0, |S_t| - 1]$  do
3    $m \leftarrow \text{Conv}(S_t[i + 1]) - S_t[i];$  // Conv2DM
4   if  $M = \emptyset$  then
5      $M \leftarrow m;$ 
6   end
7   else
8      $M \leftarrow \text{Concat}([M, m], \text{dim} = 0);$ 
9   end
10 end
11 return  $M$ 

```

---

**Algorithm 3: Motion information calculation**


---

**Input:**  $I = [I_1, \dots, I_F] \in \mathbb{R}^{F \times C \times H \times W}$ ,  $O$   
**Output:**  $M_c$

```

1  $M_c \leftarrow \emptyset;$  // Defining an empty list
2 for  $T \in O$  do
3    $S_t \leftarrow \text{SW}(I, T);$  // Slide window algorithm
4    $M \leftarrow \text{MIC}(S_t);$  // Motion information calculation
5   if  $M_c = \emptyset$  then
6      $M_c \leftarrow M;$ 
7   end
8   else
9      $M_c \leftarrow \text{Concat}([M_c, M], \text{dim} = 0);$ 
10  end
11 end
12 return  $M_c$ 

```

---

**CONTRIBUTION STATEMENT**

- **Wenbo Huang:** Proposing the idea, implementing code, conducting experiments, data collection, figure drawing, table organizing, and completing original manuscript.
- **Jinghui Zhang:** Providing experimental platform, supervision, writing polish, and funding acquisition.
- **Xuwei Qian:** Idea improvement and writing polish.
- **Zhen Wu:** Data verification and writing polish.
- **Meng Wang:** Funding acquisition
- **Lei Zhang:** Rebuttal assistance and funding acquisition.

**REFERENCES**

- [1] Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. 2021. Temporal-relational crosstransformers for few-shot action recognition. In *CVPR*. 475–484.
- [2] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Changxin Gao, Yingya Zhang, Deli Zhao, and Nong Sang. 2023. MoLo: Motion-augmented Long-short Contrastive Learning for Few-shot Action Recognition. In *CVPR*. 18011–18021.
- [3] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Mingqian Tang, Zhengrong Zuo, Changxin Gao, Rong Jin, and Nong Sang. 2022. Hybrid relation guided set matching for few-shot action recognition. In *CVPR*. 19948–19957.