Supplementary Materials

Jiazheng Xing[1*], Mengmeng Wang[1*], Yudi Ruan[1], Bofan Chen[1], Yaowei Guo[1], Boyu Mu[1], Guang Dai[2,3], Jingdong Wang[4], Yong Liu[1†]

[1] Zhejiang University, [2] SGIT AI Lab, [3] State Grid Corporation of China, [4] Baidu Inc.

{jiazhengxing, mengmengwang, yudiruan, bofanchen, guoyaowei, muboyu}@zju.edu.cn
yongliu@iipc.zju.edu.cn, guang.gdai@gmail.com, wangjingdong@baidu.com

1. Details on GNN Propagation in GgPC

Graph neural networks(GNN) are well established for the application [2, 3, 4, 1] of few-shot image classification. In our method, we followed EGNN [3] to utilize GNN as guidance to optimize the intra- and inter-class correlation within features. For simplicity and convenience, we discuss the process of the $N_{\mathcal{S}}$-way 1-shot problem and consider that the query set $\mathcal{Q}$ contains $N_{\mathcal{Q}}$ videos. We let $\mathbf{G} = (\mathbf{V}, \mathbf{A}; \mathcal{S} \bigcup \mathcal{Q})$ be the graph to construct the relationship between support set videos $\mathcal{S}$ and query videos $\mathcal{Q}$. We use the video features as node features $\mathbf{V} = \{\mathbf{v}_i\}_{i=1,\cdots,|\mathcal{S} \bigcup \mathcal{Q}|}$ and the relationship between the node features as edge features $\mathbf{A} = \{\mathbf{a}_{ij}\}_{i,j=1,\cdots,|\mathcal{S} \bigcup \mathcal{Q}|}$, where $|\mathcal{S} \bigcup \mathcal{Q}| = N_{\mathcal{S}} + N_{\mathcal{Q}}$.

Node features are initialized by the enhanced temporal features after the mean pooling operation on the temporal dimension, i.e., $\mathbf{v}_i^0 = \widetilde{\mathbf{F}_i^{avg}} (\forall i \in \mathcal{S} \bigcup \mathcal{Q})$. Edge features $\mathbf{a}_{ij} \in \mathbb{R}^2 (\forall i, j \in \mathcal{S} \bigcup \mathcal{Q})$ are 2D vectors representing the intra- and inter-class relations of the two connected nodes and are initialized with ground-truth y, as follows:

$$\mathbf{a}_{ij}^0 = \begin{cases} [1||0], & \text{if } y_i = y_j \text{ and } i, j \le N_{\mathcal{S}}, \\ [0||1], & \text{if } y_i \ne y_j \text{ and } i, j \le N_{\mathcal{S}}, \\ [0.5||0.5], & \text{otherwise}, \end{cases} \tag{1}$$

The $\mathbf{G}$ consists of L layers, and its propagation includes node features and edge features updating. Given $\mathbf{v}_i^{l-1} \in \mathbb{R}^C$ and $\mathbf{a}_i^{l-1} \in \mathbb{R}^2$ from the layer $l-1$, node features' updating is a weighted aggregation process of other nodes through the layers' edge features, as follows:

$$\mathbf{v}_i^l = f_{node}^l \big(\texttt{Cat}\big([\sum_j \tilde{a}_{ij1}^{l-1} \mathbf{v}_j^{l-1}, \sum_j \tilde{a}_{ij2}^{l-1} \mathbf{v}_j^{l-1}], dim = 0 \big) \big) \tag{2}$$

where f_{node}^l is a MLP to transform feature and $\tilde{a}_{ijb}^{l-1} = \frac{a_{ijb}^{l-1}}{\sum_h a_{ihb}^{l-1}}$ on $b \in \{1, 2\}$. After the update of node features, the edge feature is updated through the (dis)similarities between two connected features, and the sum of all edge features' values is kept constant, given by:

$$\bar{a}_{ijb}^l = \begin{cases} \frac{f_{edge}^l (|\mathbf{v}_i^l - \mathbf{v}_j^l|) a_{ijb}^{l-1}}{\sum_h f(|\mathbf{v}_i^l - \mathbf{v}_h^l|) a_{ihb}^{l-1}} \sum_h a_{ihb}^{l-1}, & if\ b = 0 \\ \frac{(1 - f_{edge}^l (|\mathbf{v}_i^l - \mathbf{v}_j^l|)) a_{ijb}^{l-1}}{\sum_h (1 - f_{edge}^l (|\mathbf{v}_i^l - \mathbf{v}_h^l|)) a_{ihb}^{l-1}} \sum_h a_{ihb}^{l-1}, & if\ b = 1 \end{cases} \tag{3}$$

$$\mathbf{a}_{ij}^l = \bar{\mathbf{a}}_{ij}^l / \|\bar{\mathbf{a}}_{ij}^l\|_1 \tag{4}$$

where f_{edge}^l is a function to calculate the similarities between two connected nodes. Here we set f_{edge}^l to a four-layer convolution block, where each layer comprises a 1×1 convolutional layer, batch normalization, and LeakyReLU activation function.

2. Implementation Details of Experimental Setup

2.1. Network Architectures

The kernel size for the 1D channel-wise temporal convolution in CTRM is set to 3. The settings of hyperparameters in each dataset are shown in Tab.1.

	Kinetics	SSv2	UCF101	HMDB51
γ	0.1	0.5	0.1	0.1
β	0.9	0.5	0.9	0.9
α	0.4	0.6	0.5	0.5

Table 1. The settings of hyperparameters in each dataset.

2.2. Training and Inference

In HPM, when T is set to 8, L is calculated as 32. The total number of training steps is set to 10. Tab.2 presents the learning rate and other settings for various datasets. In this table, lr refers to the learning rate, st_iter indicates the number of iterations per step, $steps$ represents the number

*Equal Contribution.
†Corresponding author.

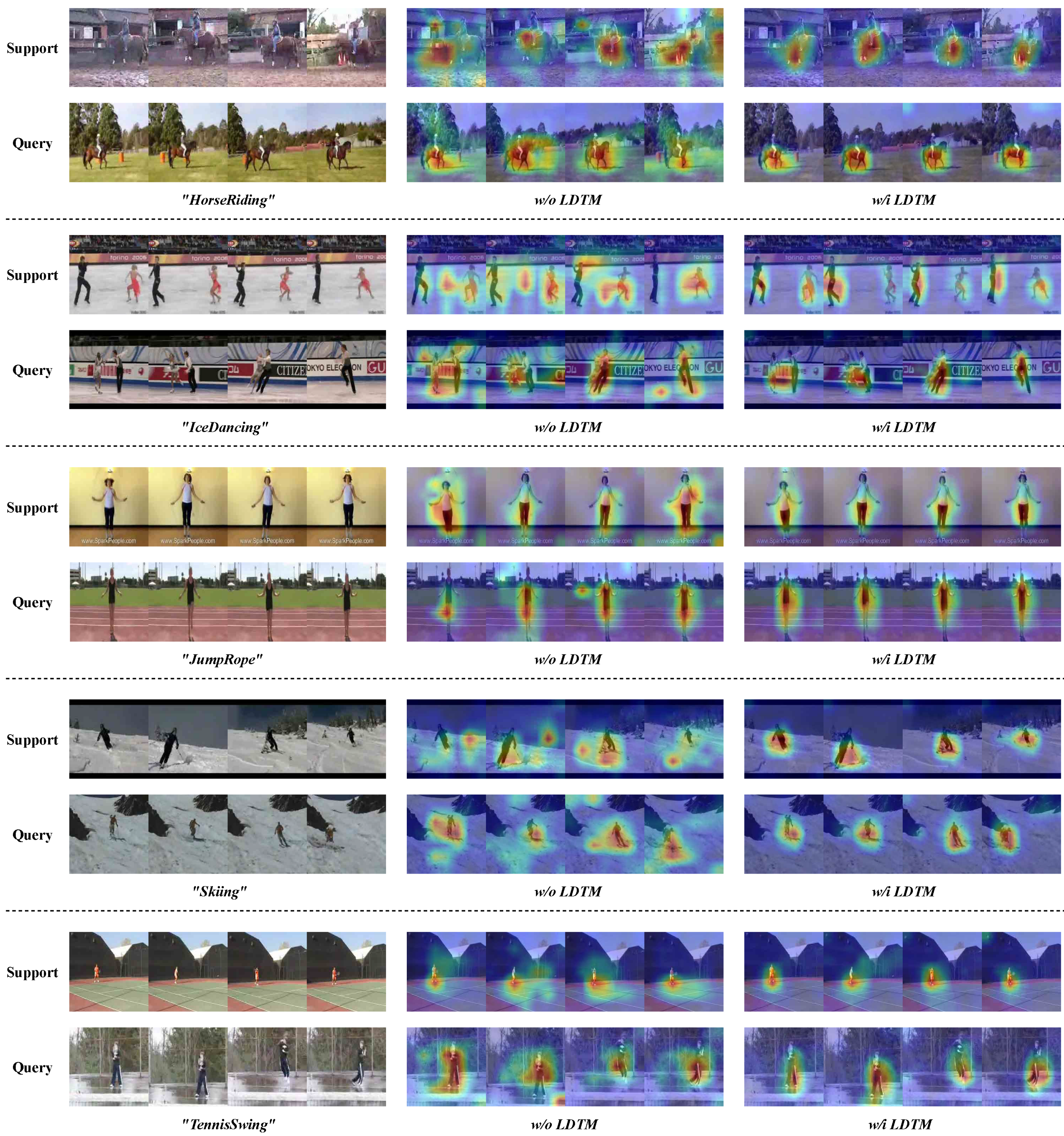

Figure 1. Attention visualization of our GgHM on UCF101 in the 5-way 1-shot setting. Corresponding to the original RGB images (left), the attention maps without LDTM modules (middle) are compared to the attention maps with our LDTM modules (right).

of steps to change the learning rate when using the multi-step scheduler, and LRS denotes the multiplication factor for updating the learning rate at each changing step.

	lr	st_iter	$steps$	LRS
Kinetics	2.2e-5	1000	[0,6,9]	[1,0.5,0.1]
SSv2	1e-4	7500	[0,6,8,9]	[1,0.5,0.1,0.01]
HMDB51	1e-4	1000	[0,2,3,5]	[1,0.5,0.1,0.01]
UCF101	5e-05	1500	[0,2,3,5]	[1,0.5,0.1,0.01]

Table 2. The settings of hyperparameters in each dataset.

3. Attention Visualization of our GgHM

Fig.1 shows the attention visualization of our GgHM on UCF101 in the 5-way 1-shot setting. Compared to the original RGB images on the left, the attention maps without LDTM modules (in the middle) are contrasted against the attention maps with our LDTM modules (on the right). Attention maps generated without the LDTM module contain numerous irrelevant or distracting focus areas. For example, the frames in "$HorseRiding$" show attention to the background and extraneous objects, diverting focus from the action. In contrast, attention maps generated using the LDTM module strongly correlate with the subject acting. Specifically, the frames in "$Skiing$" focus on the skier, and the frames in "$TennisSwing$" focus on the tennis player. These observations provide empirical evidence of the effectiveness of our LDTM module in enhancing spatiotemporal representation.

References

[1] Chaofan Chen, Xiaoshan Yang, Changsheng Xu, Xuhui Huang, and Zhe Ma. Eckpn: Explicit class knowledge propagation network for transductive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6596–6605, 2021.

[2] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[3] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11–20, 2019.

[4] Yuqing Ma, Shihao Bai, Shan An, Wei Liu, Aishan Liu, Xiantong Zhen, and Xianglong Liu. Transductive relation-propagation network for few-shot learning. In *IJCAI*, volume 20, pages 804–810, 2020.