Supplementary – In the Eye of Transformer: Global-Local Correlation for Egocentric Gaze Estimation

Bolin Lai
bolin.lai@gatech.edu
Miao Liu
mliu328@gatech.edu
Fiona Ryan
fkryan@gatech.edu
James M. Rehg
rehg@gatech.edu

College of Computing
Georgia Institute of Technology
Atlanta, GA

This is supplemental material for the paper titled "In the Eye of Transformer: Global-Local Correlation for Egocentric Gaze Estimation". We organize the content in the following way:

- A – Data Processing
- B – Implementation Details
- C – Experiments on Action Recognition
- D – Details of Different Global Visual Embedding Strategies
- E – Self-Correlation in GLC
- F – Performance Evaluation with AUC
- G – More Visualization Examples of Gaze Estimation
- H – Future Work

A Data Processing

At training time, we randomly sample 8 frames from each video with a sampling interval of 8 as input (*i.e.* selecting 8 frames from a 72-frame window with equal spacing). All videos are spatially downsampled to 256 in height while keeping the original aspect ratio. We further implement multiple data augmentations including random flipping, shifting, and resizing. We then randomly crop each frame to get an input with dimensions $8 \times 256 \times 256$. The output from the decoder is a downsampled heatmap with dimension $8 \times 56 \times 56$. For visualization, the output heatmap is upsampled to match the input size by trilinear interpolation. At inference time, the input clip is center-cropped. For gaze labels, we generate a gaussian kernel centered at the gaze location in each input frame with a kernel size of 19 following [2]. We use a uniform distribution for frames where gaze is not tracked in training and only calculate metrics on frames with fixated gaze in testing as in [7]. For the EGTEA Gaze+ [7] dataset, we determine which frames to calculate metrics on by using the provided label of gaze fixations and saccades. On the Ego4D [4] dataset, no label of gaze type is available. We calculate the euclidean spatial distance of gaze between adjacent frames and consider the tracked gaze to be a saccade if the distance is above a threshold, and treat it as fixation otherwise. We adopt an empirical threshold of 40.

B Implementation Details

We show the parameter details of each layer in Table 2. Data is input to the local token embedding module to get local tokens. Then, these tokens are fed to the global token embedding module which consists of three convolutional layers and one linear layer. Both local and global tokens are flattened into vectors of length of 96. In the following encoder, Global-Local Correlation Module (GLC), and decoder blocks, the number of local tokens is either downsampled or upsampled, while the number of global tokens remains as one. Hence we write the number of tokens in the output size as *(1 global token + number of local tokens)*. After generating the output from decoder block4, a convolutional layer is applied only on the local tokens to compress the 8 channels to 1. We then convert this to a probability distribution by applying softmax to each frame.

C Experiments on Action Recognition

In addition to egocentric gaze estimation in the main paper, we also examine the application of our GLC module to the egocentric video action recognition task, and find that our method performs competitively with methods designed specifically for this task on EGTEA Gaze+. To this end, we remove the decoder in the gaze estimation model and keep only the visual token embedding, transformer encoder, and GLC modules. Generally, there are two ways to obtain activity class category prediction: adding a class embedding token at the first layer of transformer, or using pooling across all global tokens to obtain a final embedding. Then a fully-connected layer followed by softmax is used to predict probabilities for each category. We implement both strategies and compare our approaches with previous works in Table 1. We conduct these experiments only on EGTEA Gaze+ [7] using the same split as gaze estimation. Note that the Ego4D [4] social benchmark does not contain action labels.

For vanilla MViT [3], class token embedding performs better than the pooling operation. For both methods, simply adding global embedding has a minor influence on the overall performance (-0.2% on top1 accuracy, -0.5% on top5 accuracy and $+1.32\%$ on mean class accuracy while using the class token, and -0.39%, on top1 accuracy, -0.19% on top5 accuracy and -1.19% on mean class accuracy while using pooling layer). This result suggests that simply adding global context as an additional token has minor influence on the action recognition performance.

In addition, adding our GLC module can only improve the model performance by a small margin when using class token embedding to predict action classes. We hypothesize

Methods	Cls Token	Pooling	Top1-Acc	Top5-Acc	Mean Cls Acc
MViT [3]	✓		64.64	89.22	54.02
MViT [3]		✓	63.45	88.72	55.34
MViT + Global Token	✓		64.44	88.72	55.28
MViT + Global Token		✓	63.06	88.53	54.15
MViT + Global Token + GLC	✓		64.79	88.67	56.77
MViT + Global Token + GLC		✓	65.33	**89.12**	**57.26**

Table 1: Results of action recognition on EGTEA Gaze+. We implemented two methods for classification – adding an additional class token or using global average pooling. “-"" means the result is unavailable. The complete models are highlighted.

that this is because only the class token is input into the linear layer for final prediction and re-weighted tokens from GLC are left unused. In contrast, when applying global average pooling on all local tokens, GLC improves top1, top5 and mean class accuracy over the counterpart that doesn't use GLC (*MViT+Global Token*) by +2.27%, +0.59% and +3.11%, respectively. Gains over corresponding the MViT baseline are +1.88%, +0.4% and +1.92% on the three metrics. These results indicate our proposed GLC module is a robust and general design that also improves the action recognition performance. However, the impact on action recognition is smaller compared with egocentric gaze estimation.

We note that our model achieves a competitive performance for action recognition on EGTEA Gaze+ without additional design for this specific task. Our top1 accuracy of 65.33% exceeds Wang et al. (2020) [9] by +1.23%, and is only a −1.17% difference from Hao et al. (2022)'s [5] recent state-of-the-art method for this benchmark of 66.5%. We also want to emphasize that we conduct these action recognition experiments to demonstrate the generalization ability of our proposed GLC module rather than aim to produce SOTA results on action recognition.

Additionally, we visualize the global-local correlation weights of the GLC in Fig. 1. Importantly, the learned global-local correlation is vastly different from the gaze distribution when the model is trained for action recognition; in contrast, a stronger connection between the learned global-local correlation and gaze distribution can be observed when the model is trained for gaze estimation (see Fig. 4). How to design a weakly-supervised model for egocentric gaze estimation remains an open question.

D Details of Different Global Visual Embedding Strategies

We present further details of the four global visual embedding strategies we studied in Section 4.2 of the main paper. As demonstrated in Fig. 2, (a) implements max pooling on input frames directly, and (b) implements max pooling on local visual tokens. For (c) and (d), we replace max pooling operations in (a) and (b) with a sequence of convolutional layers. The specific parameters of (d) are detailed in Table 2. For global embedding in (c), input video frames are fed into a convolutional layer that is identical to the layer used for local token embedding (*i.e.*, kernel is $3 \times 7 \times 7$ and stride is $2 \times 4 \times 4$.) Then, the output is passed to a sequence of convolutional layers identical to (d).

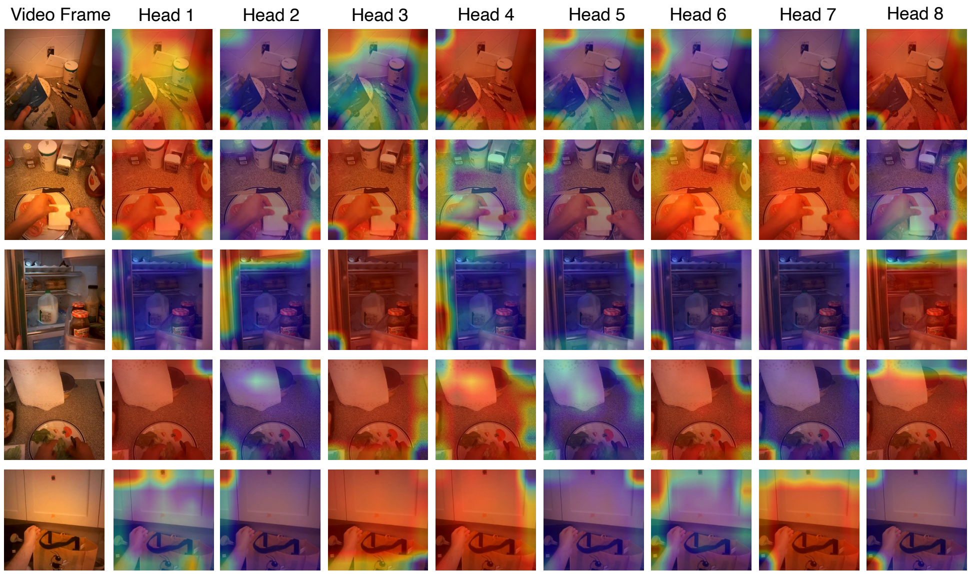

Figure 1: Visualization of the eight heads in global-local correlation module for action recognition.

E Self-correlation in GLC

In the GLC module, we calculate both self-correlation of the local token and the global-local correlation between global and local tokens. One possible concern is the effect of the self-correlation. What if we remove the self-correlation and only calculate the correlation between the local and global token? Suppose the query, key and value vectors of the global token and local token are $\mathbf{q}_{global}$, $\mathbf{k}_{global}$, $\mathbf{v}_{global}$ and $\mathbf{q}_{local}$, $\mathbf{k}_{local}$, $\mathbf{v}_{local}$. In GLC module, we calculate both global-local correlation ($\mathbf{q}_{local}^T\mathbf{k}_{global}$) and self-correlation ($\mathbf{q}_{local}^T\mathbf{k}_{local}$). The weights after softmax are denoted as w_{global} and w_{local}, respectively. Then the output is written as $w_{global} \cdot \mathbf{v}_{global} + w_{local} \cdot \mathbf{v}_{local}$. If we remove self-correlation, the output simply becomes $w_{global} \cdot \mathbf{v}_{global}$, which omits valuable local information and limits the model performance. We validated this assertion by conducting an additional experiment with self-correlation removed. The resulting model achieves F1 score of 43.8% and 41.7% on EGTEA Gaze+ and Ego4D, respectively. The drop in performance supports our design choice of calculating both global-local correlation and self-correlation.

F Performance Evaluation with AUC

AUC is easy to become saturated because of the long-tailed nature of the distribution of gaze in a single frame. However, we still calculated the AUC metric to our evaluation to make a thorough comparison. We obtained an AUC score for our model of 0.935 on EGTEA Gaze+ and 0.938 on Ego4D, compared to [6] which had 0.924 on EGTEA Gaze+ and 0.927 on Ego4D. Thus, we improved the performance by +1.1% and +0.8%, demonstrating the superiority of our model in terms of AUC. Note that the improvement on AUC is much smaller than F1 (+7.6% on EGTEA Gaze+ and +5.6% on Ego4D) due to its saturation in

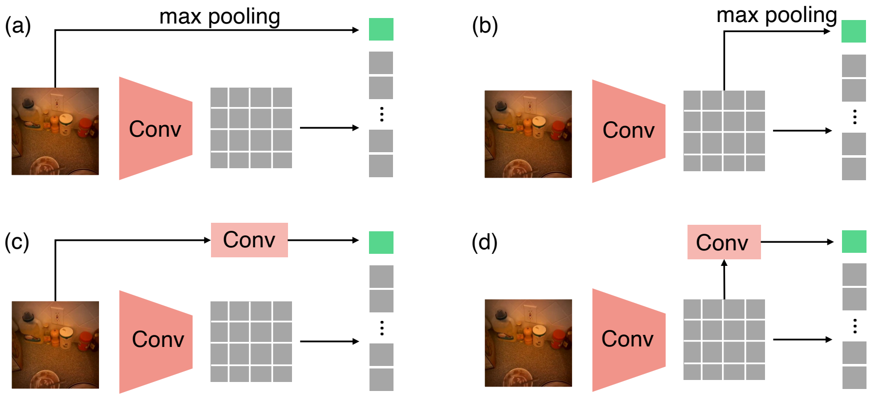

Figure 2: Four different approaches of global visual token embedding.

egocentric gaze estimation task.

G More Visualization Examples of Gaze Estimation

More visualizations of gaze prediction of both our model and previous state-of-the-art approaches are presented in Fig. 3. Our proposed model can accurately predict the gaze distribution even when the scene context is very complicated, while the other three approaches may be misled by background objects or produce predictions with too much uncertainty.

We provide more examples of GLC visualizations in Fig. 4. The 8 heads capture features of different areas which is consistent with the examples in the main paper. On the EGTEA Gaze+ dataset, the maps produced by heads 1, 4, 5, and 8 highlight pixels around the gaze point with different uncertainty (which is illustrated by the size of highlighted area). The other four heads focus on surrounding objects and leave gaze areas unattended. As for the Ego4D data, only head 3 captures the wearers' attention, while the other heads fully focus on the backgrounds in different aspects. This supports our key conclusion in the main paper that our GLC module learns to model human attention by setting different weights from local to global tokens, capturing many facets of scene information (both around the gaze target and in the background) in the multi-headed attention mechanism.

H Future Work

In this paper, we studied the explicit integration of global scene context for egocentric gaze estimation and proposed a novel modeling approach for this problem. We also showed the results of our proposed architecture on egocentric action recognition in this supplementary material to demonstrate our model's generalization ability. Our findings also point to several exciting future research directions:

- Our proposed GLC module has the potential to address other video understanding tasks including visual saliency prediction in third-person video, active object detection, and future forecasting. We plan to study the effect of our method on those tasks in our future work.

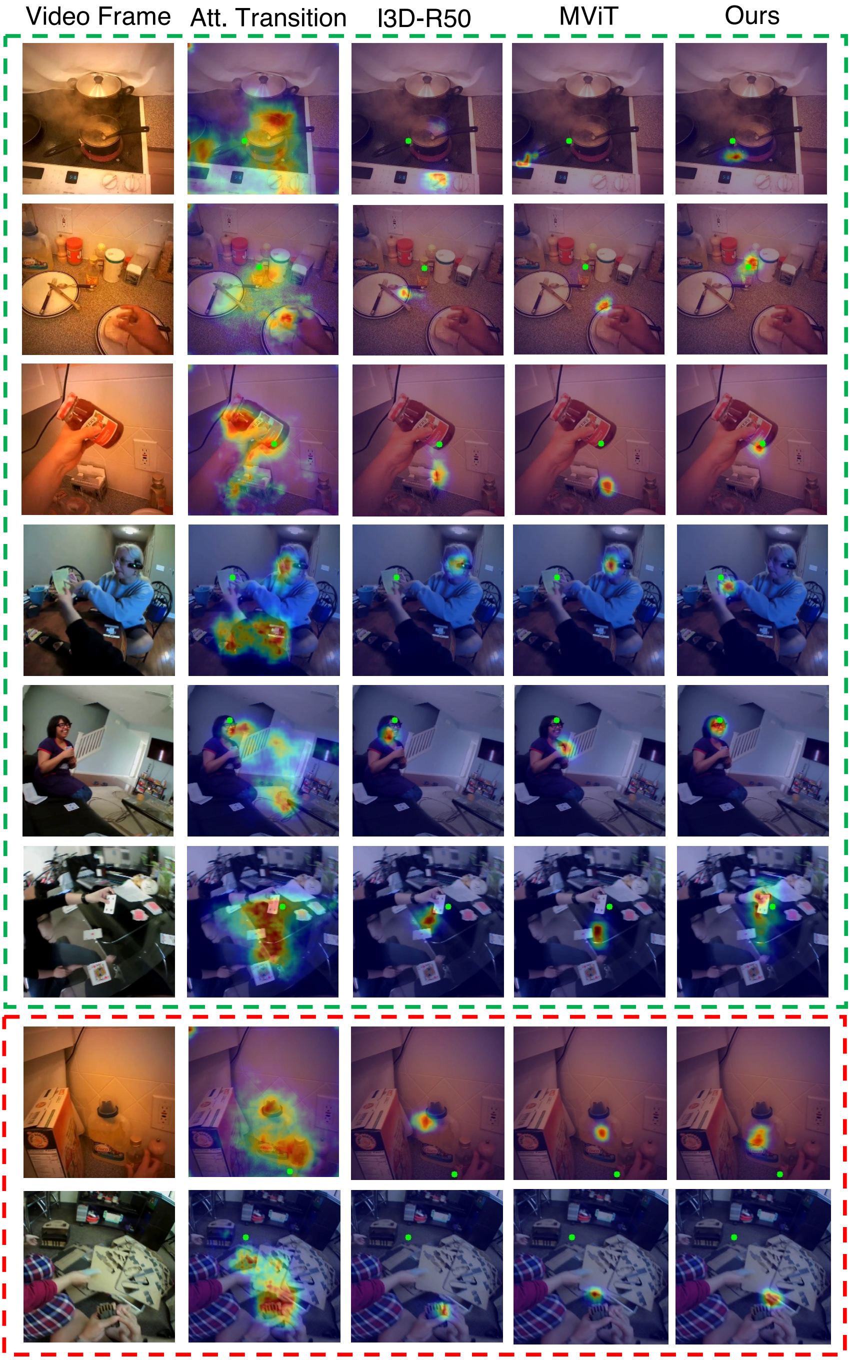

Figure 3: Visualization of gaze estimation. Both successful cases (in green box) and failure cases (in red box) of our model are demonstrated. Green dots present ground truth.

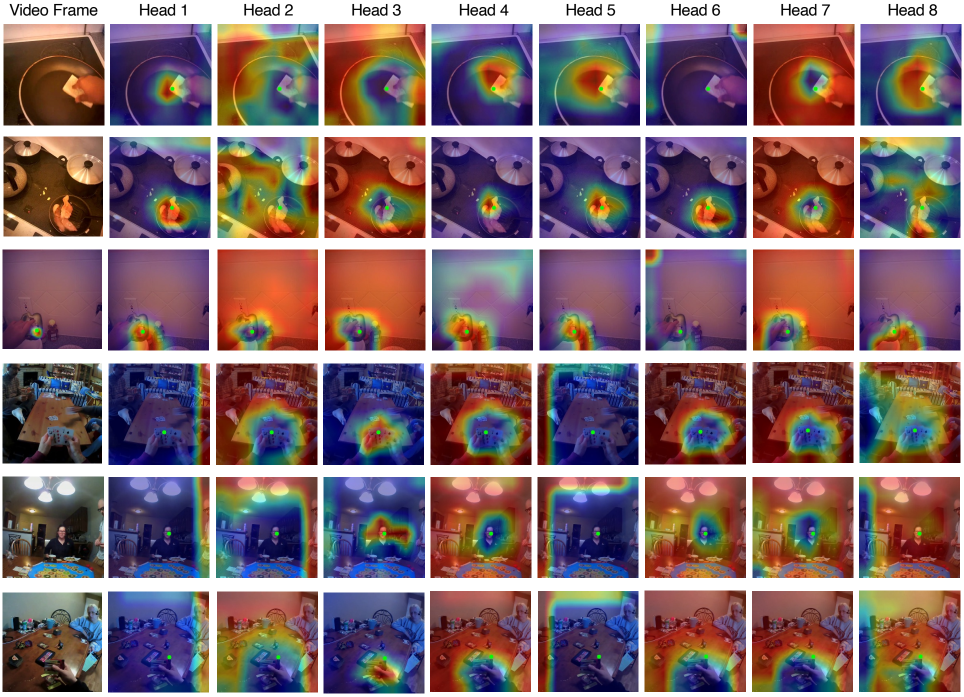

Figure 4: Visualization of the eight heads in the Global-Local Correlation module. Green dots represent the ground truth.

- Our modeling work can be expanded to understanding human gaze behavior associated with multiple sensing modalities, especially in the social conversation setting. An exciting future direction is incorporating audio signals into egocentric gaze estimation.

- Our proposed GLC fails to learn the gaze distribution when the model is trained to predict the action labels. How to design a weakly supervised model for egocentric gaze estimation using action labels is an interesting problem.

- Our transformer based model requires larger computational cost, and therefore may not be feasible for on-device computing (e.g. AR/VR). We will continue to study how to combine it with some recent works on network architecture research [1] and knowledge distillation [8] to reduce the computational cost of transformer architecture.

References

[1] Boyu Chen, Peixia Li, Chuming Li, Baopu Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Glit: Neural architecture search for global and local image transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–21, 2021.

[2] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. Detecting attended visual targets in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5396–5406, 2020.

[3] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[4] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[5] Yanbin Hao, Hao Zhang, Chong-Wah Ngo, and Xiangnan He. Group contextualization for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 928–938, 2022.

[6] Yifei Huang, Minjie Cai, Zhenqiang Li, and Yoichi Sato. Predicting gaze in egocentric video by learning task-dependent attention transition. In *Proceedings of the European conference on computer vision (ECCV)*, pages 754–769, 2018.

[7] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635, 2018.

[8] Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and Gang Wang. Knowledge distillation via the target-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10915–10924, 2022.

[9] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Symbiotic attention with privileged information for egocentric action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12249–12256, 2020.

Stages	Operators	Output Size
data	-	$8 \times 256 \times 256$
local token embedding	$Conv(3 \times 7 \times 7,\ 96)$ $stride\ 2 \times 4 \times 4$	$96 \times 4 \times 64 \times 64$
global token embedding	$Conv(3 \times 3 \times 3,\ 96)$ $Conv(3 \times 3 \times 3,\ 96)$ $Conv(3 \times 3 \times 3,\ 96)$ $Linear(24576)$ $stride\ of\ each\ conv\ 1 \times 2 \times 2$	96×1
tokenization	flattening and concatenation	$96 \times (1+4 \times 64 \times 64)$
encoder block1	$\begin{bmatrix} MSA(96) \\ MLP(384) \end{bmatrix} \times 1$	$192 \times (1+4 \times 64 \times 64)$
encoder block2	$\begin{bmatrix} MSA(192) \\ MLP(768) \end{bmatrix} \times 2$	$384 \times (1+4 \times 32 \times 32)$
encoder block3	$\begin{bmatrix} MSA(384) \\ MLP(1536) \end{bmatrix} \times 11$	$768 \times (1+4 \times 16 \times 16)$
encoder block4	$\begin{bmatrix} MSA(768) \\ MLP(3072) \end{bmatrix} \times 2$	$768 \times (1+4 \times 8 \times 8)$
global-local correlation	$\begin{bmatrix} GLC(768) \\ MLP(3072) \\ concatenation\ in\ channel \end{bmatrix} \times 1$	$1536 \times (1+4 \times 8 \times 8)$
decoder block1	$\begin{bmatrix} MSA(1536) \\ MLP(3072) \end{bmatrix} \times 1$	$768 \times (1+4 \times 16 \times 16)$
decoder block2	$\begin{bmatrix} MSA(768) \\ MLP(1536) \end{bmatrix} \times 1$	$384 \times (1+4 \times 32 \times 32)$
decoder block3	$\begin{bmatrix} MSA(384) \\ MLP(768) \end{bmatrix} \times 1$	$192 \times (1+4 \times 64 \times 64)$
decoder block4	$\begin{bmatrix} MSA(192) \\ MLP(384) \end{bmatrix} \times 1$	$96 \times (1+8 \times 64 \times 64)$
head	$Conv(1 \times 1 \times 1,\ 1)$ $stride\ 1 \times 1 \times 1$	$8 \times 64 \times 64$

Table 2: Architecture of the proposed model. Convolutional layers are denoted as $Conv(kernel\ size,\ output\ channels)$. Numbers of input channels of multi-head self-attention are shown in the parenthesis of *MSA*. Dimensions of the hidden layer in multi-layer perceptrons are listed in parenthesis of *MLP*. In tokenization, local and global tokens are reshaped and concatenated. In global-local correlation, the output is concatenated with its input in the channel dimension. Head only takes local tokens as input.