Appendix

A. Implementation detail

We implement our model ViPLO on top of the Pytorch implementation [1] of SCG [4] . We also use the Pytorch implementation [2] of CLIP [3] for the backbone network, including modifications related to the proposed MOA module. As mentioned in Sec. 3.3, we follow the training and inference procedure of SCG, such as appending the ground-truth boxes during the training, applying non-maximum suppression (NMS) to the detection results, and computing the final HOI scores for the focal loss. The final HOI scores are computed as in SCG:

$$\mathbf{s}_k = (s_i^h)^\lambda \cdot (s_j^o)^\lambda \cdot \tilde{s_k}, \tag{1}$$

where s_i^h denotes the ith human detection score, s_j^o denotes the jth object detection score, and $\tilde{s_k}$ is the action classification score obtained from the representation of the HOI triplet, including human node encoding, object node encoding, and their edge encoding. These encodings are fused with the MBF module in the SCG. We set λ to 1 in the training process and 2.8 in the inference process [4]. Finally, the focal loss [1] is used as the multi-label classification loss to train the possible interactions for each human-object pair as follows.

$$FL(\hat{y}, y) = \begin{cases} -\alpha(1-\hat{y})^\gamma \log(\hat{y}), & y = 1 \\ -(1-\alpha)\hat{y}^\gamma \log(1-\hat{y}), & y = 0 \end{cases} \tag{2}$$

where y is the ground-truth label, $\hat{y}$ is the final score for the human-object pair, and α and γ are balancing parameters. For focal loss, we set α to 0.5 and γ to 0.2 [4].

When using the Vision Transformer backbone, CLS tokens in which the MOA module is applied are mapped to initialization of each node encoding with a two-layer MLP. We use a three-layer MLP to construct an edge encoding from human pose and spatial information. For extracting the local feature of human, we draw the local region box for each joint as 0.3 times the size of the human bbox height. For the message function using human local node encodings and object node encodings (Eq. 4 in the paper), we concatenate two node encodings for the appearance feature in the MBF module.

We use the AdamW [2] optimizer for training with an initial learning rate of 10^{-4}. For HICO-DET, we train the VIPLO for 8 epochs with flip data augmentation and the learning rate decay by a factor of 0.1. For V-COCO, we train the model for 20 epochs with additional data augmentations including color jittering, and decay the learning rate at the 10^{th} epochs. For the convenience of the experiment, we did not use the pose information for the V-COCO dataset. We perform all experiments with 3 NVIDIA RTX A6000 GPUs using the Pytorch 1.9.0. framework. We use batch size 11 per GPU for ViPLO$_s$ and 8 per GPU for ViPLO$_l$.

B. Efficient computation for MOA

The MOA module leads to a large performance increment in HOI detection, as shown in ablation studies in Sec. 4.3. But to use the MOA module, overlapped area S has to be computed for each bounding box, which may be a computational burden under the CPU operation. So we design the entire process of computing S to be possible through GPU operations. In specific, we compute the overlapped area of each row patch and column patch, then obtain the total overlapped area efficiently by multiplying these two. Details can be found in Algorithm 1.

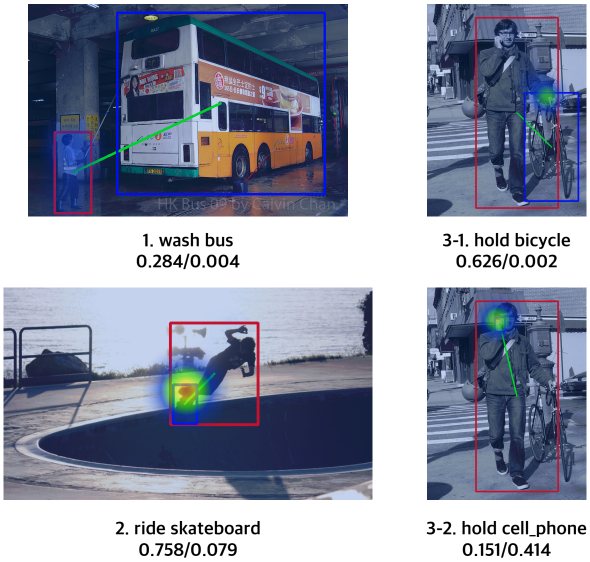
Algorithm 1 Torch-like pseudo-code for the MOA module

```
Input: box coordinate $\mathbf{b}$, patch size $p$, attention map length $L$
Output: attention mask $A$
width = int($\sqrt{L}$)
$A$ = zeros($1, L$)
$\mathbf{b} = \mathbf{b}/p$
$\mathbf{b}_{int}$ = [floor($\mathbf{b}$[0:2]), ceil($\mathbf{b}$[2:4])]
$\mathbf{b}_{wh}$ = 1 - abs($\mathbf{b}_{int} - \mathbf{b}$)
$a, b, c, d = \mathbf{b}_{int}$
$x, y, z, w = \mathbf{b}_{wh}$
row = arange (width * $b$ + $a$ + 1, width * $b$ + $c$ + 1)
mask_index = row.repeat($d$ - $b$) + arange($d$ - $b$).repeat_interleave($c$ - $a$) * width
area_row = [$x$, ones($c$ - $a$ - 2), $w$]
area_column = [$y$, ones($d$ - $b$ - 2), $z$]
mask_area = area_row * area_column
$A$[0, mask_index] = mask_area
```

Another issue is that simply applying the MOA module increases the amount of computation in proportion to the number of regions in the given image. Hence, we propose three methods for reducing computation: 1) we apply the MOA module only in the last layer of ViT, which is sufficient for a feature to be conditioned to a given region; 2) we compute an attention score only for the CLS token, as the CLS token serves as an extracted feature; and 3) we calculate the dot product of the query and key only once, then add $\log(S)$ for each copied attention map in Eq. 1 in the paper. We apply three methods together to reduce the computational complexity of MOA. The computational complexity of the original ViT layer is $O(L^2 \cdot C)$, and that of MOA applied the ViT layer is $O(M \cdot L \cdot C + M \cdot C^2)$, where L, C, M denotes the number of patches, hidden dimension, and the

[1] https://github.com/fredzzhang/spatially-conditioned-graphs

[2] https://github.com/openai/CLIP



1. wash bus
0.284/0.004

3-1. hold bicycle
0.626/0.002

2. ride skateboard
0.758/0.079

3-2. hold cell_phone
0.151/0.414

Figure 1. Qualitative results of ViPLO$_l$ compared to the baseline SCG. For each image, the prediction scores of ViPLO$_l$ and SCG are shown (left: ViPLO$_l$, right: SCG). The joint attention in Eq.2 in the paper is also visualized as a heatmap.

number of regions, respectively. The latter is linear to the number of patches since the number of regions is limited by non-maximum suppression (NMS), showing the efficiency of the MOA module.

C. Qualitative Results

We show qualitative results of ViPLO compared to the SCG in Fig. 1. We find that ViPLO can successfully detect difficult interactions where the human and object are far away (case 1). ViPLO also effectively detects interactions focusing on specific human joint, such as ankle or wrist in case of riding skateboard (case 2). Surprisingly, we find that our model focuses on different joints when detecting interaction for different objects, even in the same image (case 3). These results prove the effectiveness of ViPLO.

References

[1] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 1

[2] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1

[4] Frederic Z Zhang, Dylan Campbell, and Stephen Gould. Spatially conditioned graphs for detecting human-object interactions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13319–13327, 2021. 1