—Supplementary Material— Detecting Human-Object Interactions with Action Co-occurrence Priors

Anonymous ECCV submission

Paper ID 3850

The contents of this supplementary material include additional experimental results, implementation details of our model, more qualitative results and analysis which have not been shown in the main paper due to the space limitation.

1 Additional Experimental Results

In this section, we provide additional qualitative results.We also analyze the effect of using different numbers of anchor actions introduced in Section 3.2 of the main paper, performance across different classes, and analyze false predictions in our model's predictions.

1.1 Qualitative Results

Fig. 1 shows examples of HOI detection results that our model predicts correctly with high probability. We show each image with the predicted HOI class followed by the probability computed by our model.

1.2 Number of Anchor Actions

In addition, we investigate the effect of using different numbers of anchor actions $|\mathcal{D}|$ in Fig. 2. We measure the relative performance improvement from the ***+Hierarchical*** model to the ***Modified Baseline*** model by changing the number of anchor actions $|\mathcal{D}|$ at intervals of five.

In principle, the more anchor actions we use, the better performance that can be attained. On one hand, the selected anchor actions can be more distinguishable from one another with more anchor action categories and training samples. On the other hand, the remaining regular actions can also benefit from stronger co-occurrence priors. However, more anchor action will also result in more sub-networks to optimize, and this will cause over-fitting to a certain extent. Through observations, we found that an increase in the number of parameters of the HOI detector often causes a severe performance decrease. Thus, there is a trade-off between a large and small number of anchors, which requires us to empirically select the best anchor number. As depicted in Fig. 2, the hierarchical architecture shows the best overall mAP score (Full) with 15 anchors and the best mAP score on rare classes with 10 anchors. We finally use an experimentally overall best-performing choice of 15 anchor actions (maximum anchor action number is 54).

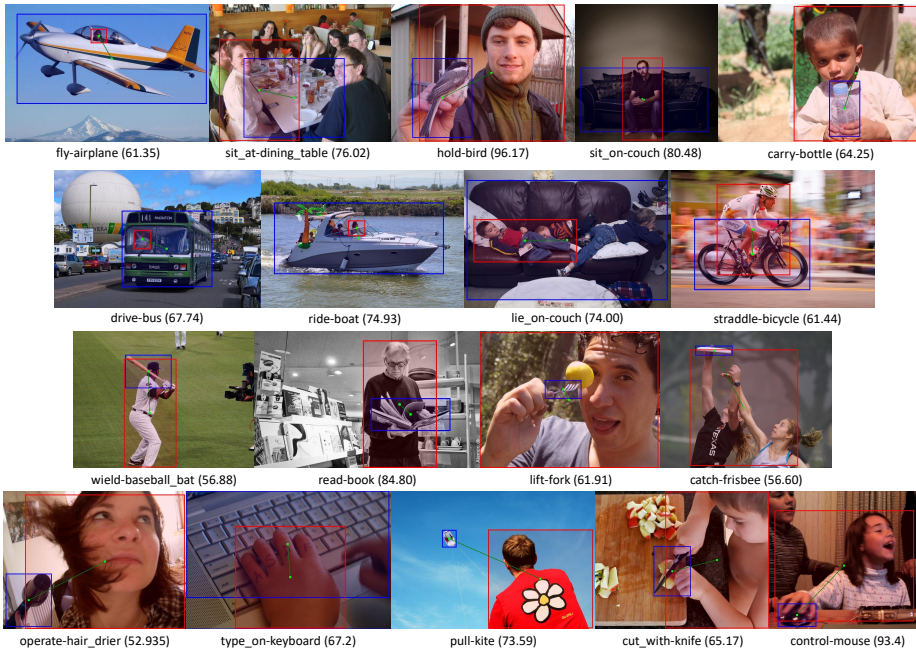


Fig. 1. Examples of bounding boxes and HOI detection scores from our model. Bounding boxes for humans are colored red, and bounding boxes for objects are coloreds blue. Each image is displayed with the predicted action+object class followed by the probability computed by our model.

1.3 Performance across Different Classes

Fig. 3 shows the improvements in mAP scores across different object and action classes. We compare our model with the baseline model.In order to visualize on which classes our method improves performance, the figure is organized in the order of the improvements of the scores of each class. It is found that our method improves mAP scores in 64 out of the 80 object classes (80%) and 90 out of the 117 actions (77%).

1.4 Analysis of False Predictions

Fig. 4 shows examples of false predictions by our model. We found three common reasons that the prediction is evaluated as false. First is the wrong prediction of object label from the object detector (*e.g.*,'couch' as 'chair' or 'sheep' as 'cow') which is shown in the left column of Fig. 4. Second, the prediction is correct but the ground truth label for the prediction in a test image is missing (*e.g.*,'ride-bus' or 'hold-horse') which is shown in the middle column of Fig. 4. Last, a HOI detector could have been predicted correctly if there were a sophisticated way to take context (the third object or the background) into account (*e.g.*,the

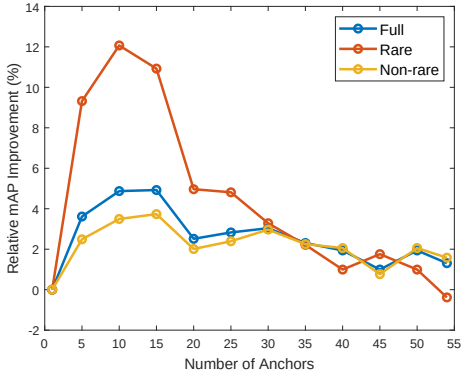


Fig. 2. Performance of the hierarchical architecture with different number of anchors at intervals of five. The model with 15 and 10 anchors show the best performance overall and on rare classes, respectively.

background for predicting 'repair-bicycle' or an object that person carries for predicting 'load-bus') which is shown in the right column of Fig. 4. The last issue can be solved by devising a better network architecture for effectively encoding context, which is a direction orthogonal to our work. All of these issues (the errors in the object detector, missing labels in datasets, and encoding context) are fundamental issues in HOI detection, which can be interesting topics for future work.

2 Implementation Details

In this section, we provide more implementation details of our method which are not discussed in the main paper due to the space limitation.

2.1 Knowledge Distillation in Object-agnostic Level

In Knowledge Distillation via ACP Projection (Section 3.4 in the main paper), the action co-occurrence matrices C_o and $C_o^{'}$ used for ACP Projection are object-aware (described in Eq. (17) and Eq. (18) in the main paper). We also use the object-agnostic action co-occurrence matrix C and $C^{'}$.

We thus can generate two more object-aware teacher labels when constructing teacher objectives:

$$\hat{Y}_{proj} = joint(\hat{H}, \hat{O}, project(\hat{A}, C, C^{'})) \in \mathbb{R}^{M}, \tag{1}$$

$$Y^{gt}_{proj} = joint(H^{gt}, O^{gt}, project(A^{gt}, C, C^{'})) \in \mathbb{R}^{M} \tag{2}$$

Finally, the total loss function we used to train the ACP model is expressed as:

$$\begin{aligned}\mathcal{L}_{tot} =& \lambda_1 \mathcal{L}(\hat{Y}, Y^{gt}) + \lambda_2 \mathcal{L}(\hat{Y}, \hat{Y}_{projO}) + \lambda_3 \mathcal{L}(\hat{Y}, Y^{gt}_{projO}) \\ &+ \lambda_4 \mathcal{L}(\hat{Y}, \hat{Y}_{proj}) + \lambda_5 \mathcal{L}(\hat{Y}, Y^{gt}_{proj}),\end{aligned} \tag{3}$$

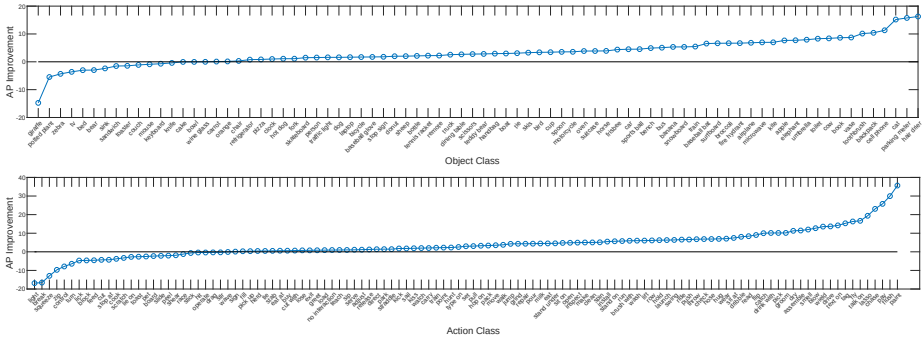


Fig. 3. The AP score improvements across different object and action classes sorted by the amount of improvement compared to the baseline model [3]. Our method improves mAP scores in 64 out of the 80 object classes (80%) and 90 out of the 117 actions (77%).

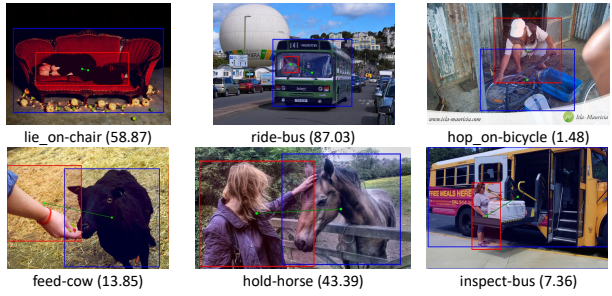


Fig. 4. Examples of the false predictions from our model. The false cases include wrong prediction of object label from the object detector (left column), correct prediction but missing ground truth labels (middle column), and predictions that could have been correct if the context were taken into account (right column).

by introducing two more loss balancing weights λ_4, λ_5.

2.2 Details of Multiple Information and Fusion Networks

In Section 3.3 of the main paper, we introduced the baseline network "no-frills" [3]. We now give detailed definitions of the multiple information X and the fusion networks $F(\cdot)$ used in this baseline and our modified baseline.

The CNN features used for HOI classification are denoted as $\hat{x} = \{\hat{x}_h, \hat{x}_o\}$, where $\hat{x}_h$ and $\hat{x}_o$ are for human and object appearance, respectively. These CNN features are directly extracted from the final fully connected (FC) layer of the detector. The multiple information consists of human appearance $\hat{x}_h$, object appearance $\hat{x}_o$, as well as human pose ($\hat{k}$), object category ($\hat{o}$) and bounding boxes ($\hat{b}$ including both object and human bounding boxes). All of these input cues $X = \{\hat{x}, \hat{k}, \hat{o}, \hat{b}\}$ are fed to our target model. There are four separate network streams: human appearance ($f_h(\cdot)$), object appearance ($f_o(\cdot)$), bounding box

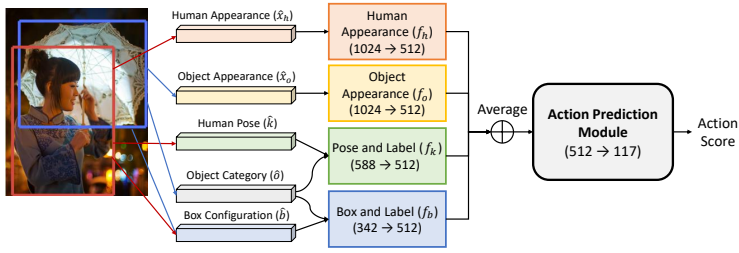


Fig. 5. Overall illustration of the basic components of our network architecture (More symbols added). The design choices for the hierarchical *action prediction module* are detailed in Fig. 7.

and object category ($f_b(\cdot)$), and human pose and object category ($f_k(\cdot)$). Each individual type of information is first fed through a separate network of two FC layers to generate a fixed dimension (number of actions N) feature. Then, all the features are added together and sent through a *sigmoid* activation to get the probability prediction for the action a:

$$\hat{A}(a) = p(a|X) = sigmoid(F(X)). \tag{4}$$

The multi-information fusion procedure $F(X)$ is expressed as

$$F(X) = f_h(\hat{x}_h) + f_o(\hat{x}_o) + f_k(\hat{k}||\hat{o}) + f_b(\hat{b}||\hat{o}), \tag{5}$$

where the operation $||$ denotes concatenation.

However, instead of directly adding up the multiple information, we average them and forward this through another *action prediction module* f_{sub} of a few FC layers to obtain the final action probability prediction as illustrated in Fig. 5. For a naive approach (denoted as the ***Modified Baseline***), we simply use a sub-network f_{sub} of one FC layer as the *action prediction module*. Then Eq. (4) and Eq. (5) are modified to

$$\hat{A} = p(a|X) = sigmoid(f_{sub}(F(X))), \tag{6}$$

$$F(X) = \frac{f_h(\hat{x_h}) + f_o(\hat{x_o}) + f_k(\hat{k}||\hat{o}) + f_b(\hat{b}||\hat{o})}{n_{stream}}. \tag{7}$$

In this case, $n_{stream} = 4$.

2.3 Training Details

We employ Faster R-CNN [7] with a Feature Pyramid Network (FPN) [5] and ResNet-101 [4] backbone as an object detector and freeze it while training an HOI classifier. This detector was originally trained on the MS COCO dataset [6], which has the same 80 object categories as the HICO-Det dataset.The HOI classifier consists of four streams similar to [1,2], extracting features from instance

Table 1. The list of 54 anchors made out of all 117 actions from HICO-Det dataset.

flush	tag	stab	wave	brush with	hunt
toast	pay	move	teach	no interaction	eat at
milk	squeeze	greet	stop at	stir	install
point	sign	paint	shear	release	control
break	light	lose	zip	lift	pack
cut with	type on	talk on	set	slide	operate
spin	assemble	pour	lie on	turn	chase
herd	flip	buy	hose	kick	row
peel	dry	hop on	direct	adjust	kiss

appearance, spatial location, and human pose. The pose and spatial streams are composed of two FC layers whereas the human and object appearance streams consist of one FC layer of size 512. The dimension of the two hidden layers in the pose and spatial streams is 512. The outputs of the four streams are consolidated via average pooling and passed through an FC layer of size 512 to perform HOI classification. We consider all detections for which the detection confidence is greater than 0.01 and create human-object pairs for each image. The image-centric training strategy [7] is also applied. In other words, pair candidates from one image make up each mini-batch. We adopt SGD and set the initial learning rate to 1e-3, weight decay to 1e-4, and momentum to 0.9. For the ratio of positive to negative samples in training, while [3] uses 1:1000, we suspect 1:600 is more reasonable because intuitively the ratio of the positive and the negative sample is likely to be 1:(Number of classes). We empirically found out 1:600 gives better performance than 1:1000. We train the framework for 100000 iterations. All experiments are conducted on a single Nvidia K40 GPU.

3 More Illustrations for Our Methods

3.1 Illustration for NES

Fig. 6 shows a step-by-step example of the anchor selection process among eight action classes (from 'A' to 'H'). The color of each node denotes whether or not its exclusiveness value e_i is included in the exclusiveness pool $\mathcal{E}$. The i-th node is blue if $e_i \in \mathcal{E}$, orange if selected as an anchor action, gray if $e_i \notin \mathcal{E}$) The color of the edge denotes whether or not the co-occurrence value c_{ij} is included in the co-occurrence pool $\mathcal{C}$.

3.2 Design Choices for Hierarchical Architecture

Fig. 7 illustrates the design choices for our hierarchical architecture (***Modified Baseline***, ***MultiTask***, ***TwoStream***, and ***Hierarchical***).

4 Extras

4.1 List of Anchors

Table 1 shows the list of maximum number of anchor actions from the HICO-Det dataset in the order in which they are selected. For example, when setting the number of anchor actions to be 10, we take from the first action ('flush') to the tenth action ('teach') as the set of anchors, and the rest are associated to 'other.'

4.2 Example of Co-occurrence Matrices for Other Objects

Fig. 8 shows more examples of the co-occurrence matrices C_o for object o, constructed from the HICO-Det dataset.

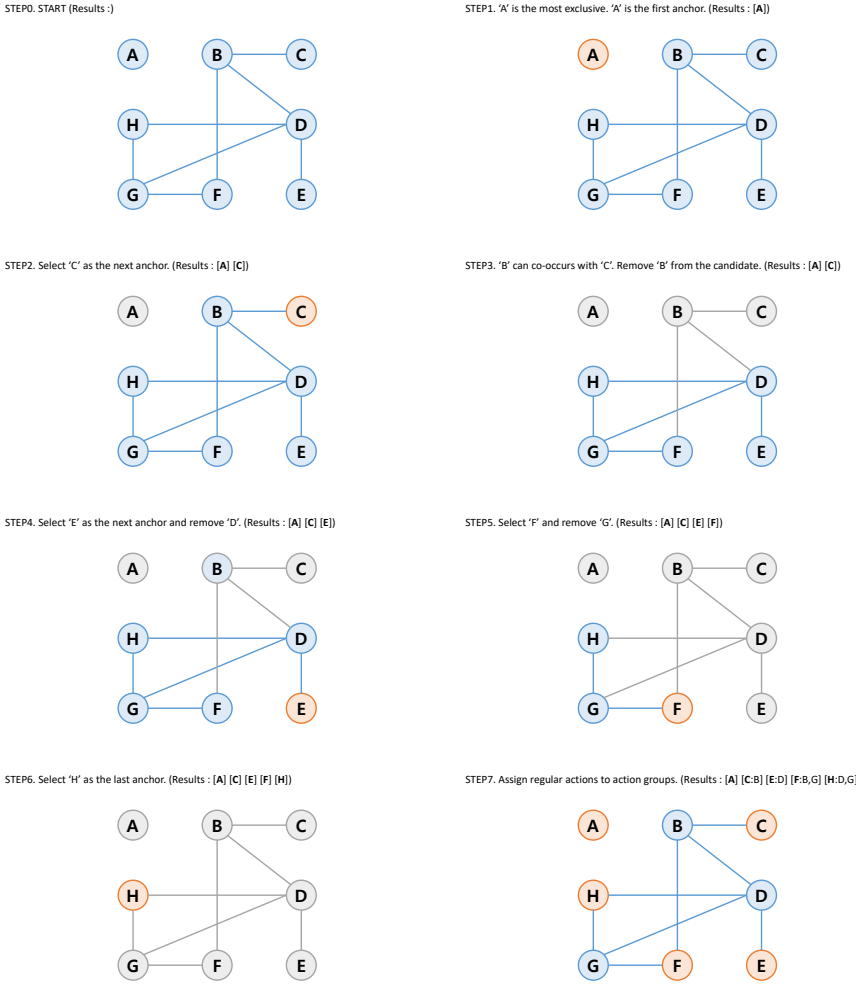
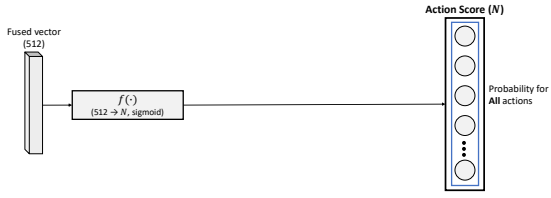
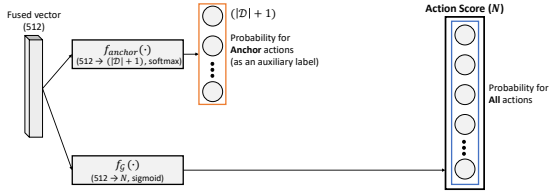


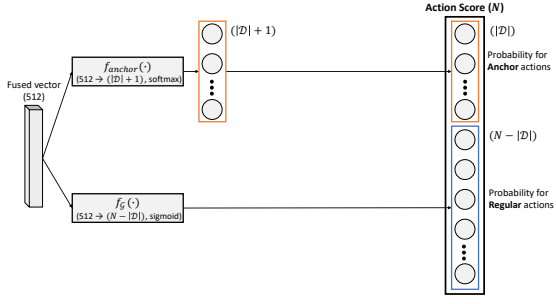
Fig. 6. Illustration of the proposed anchor action selection process.



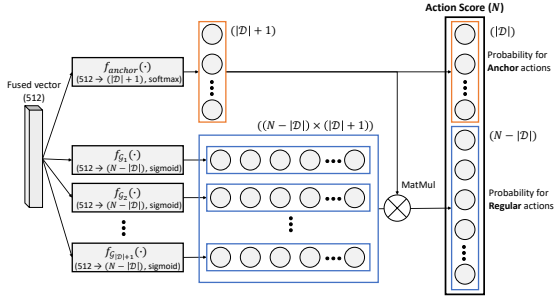
(a) ***Modified Baseline***



(b) ***MultiTask***



(c) ***TwoStream***



(d) ***Hierarchical***

Fig. 7. Design choices for the action prediction module of our network. (a) ***Modified Baseline*** architecture without leveraging the prior of anchor actions. (b) ***MultiTask*** which utilizes the anchor actions as in a multi-task learning manner. (c) ***TwoStream*** which separately predicts the anchor and the regular actions but without using the hierarchical modeling between anchor and regular actions. (d) The proposed hierarchical target (***Hierarchical***). Anchor probability is directly used as a final score, and we exploit multiple conditional sub-networks to further compute the probabilities for the regular actions.

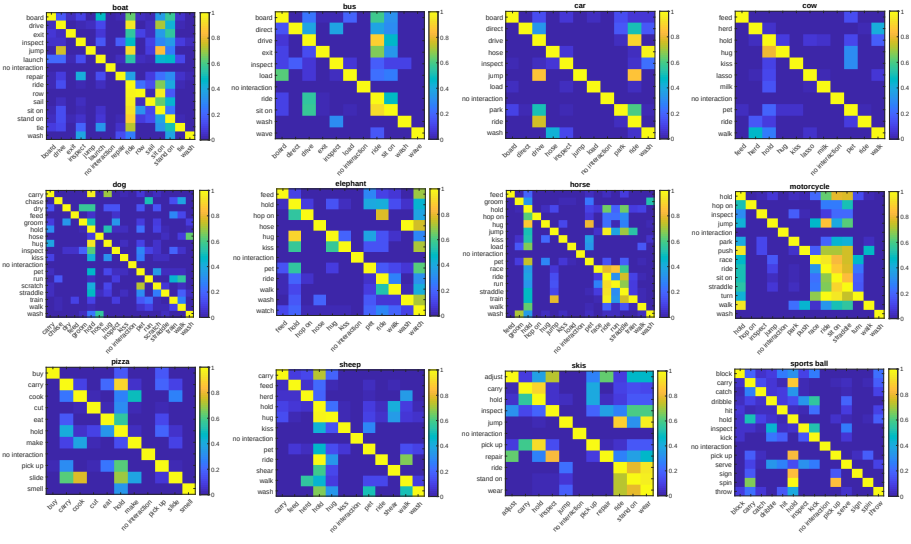


Fig. 8. Example of co-occurrence matrices C_o for each object o constructed from HICO-Det dataset.

References

1. Chao, Y.W., Liu, Y., Liu, X., Zeng, H., Deng, J.: Learning to detect human-object interactions. In: IEEE Winter Conference on Applications of Computer Vision (WACV) (2018)
2. Gao, C., Zou, Y., Huang, J.B.: ican: Instance-centric attention network for human-object interaction detection. In: British Machine Vision Conference (BMVC) (2018)
3. Gupta, T., Schwing, A., Hoiem, D.: No-frills human-object interaction detection: Factorization, layout encodings, and training techniques. In: IEEE International Conference on Computer Vision (ICCV) (2019)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
5. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
6. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV) (2014)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS) (2015)