Efficient Two-Stage Detection of Human–Object Interactions with a Novel Unary–Pairwise Transformer

Frederic Z. Zhang\textsuperscript{1,3} Dylan Campbell\textsuperscript{2,3} Stephen Gould\textsuperscript{1,3}
\textsuperscript{1}The Australian National University \textsuperscript{2}University of Oxford \textsuperscript{3}Australian Centre for Robotic Vision

Abstract

Recent developments in transformer models for visual data have led to significant improvements in recognition and detection tasks. In particular, using learnable queries in place of region proposals has given rise to a new class of one-stage detection models, spearheaded by the Detection Transformer (DETR). Variations on this one-stage approach have since dominated human–object interaction (HOI) detection. However, the success of such one-stage HOI detectors can largely be attributed to the representation power of transformers. We discovered that when equipped with the same transformer, their two-stage counterparts can be more performant and memory-efficient, while taking a fraction of the time to train. In this work, we propose the Unary–Pairwise Transformer, a two-stage detector that exploits unary and pairwise representations for HOIs. We observe that the unary and pairwise parts of our transformer network specialise, with the former preferentially increasing the scores of positive examples and the latter decreasing the scores of negative examples. We evaluate our method on the HICO-DET and V-COCO datasets, and significantly outperform state-of-the-art approaches. At inference time, our model with ResNet50 approaches real-time performance on a single GPU.

1. Introduction

Human–object interaction (HOI) detectors localise interactive human–object pairs in an image and classify the actions. They can be categorised as one- or two-stage, mirroring the grouping of object detectors. Exemplified by Faster R-CNN [24], two-stage object detectors typically include a region proposal network, which explicitly encodes potential regions of interest in the form of bounding boxes. These bounding boxes can then be classified and further refined via regression in a downstream network. In contrast, one-stage detectors, such as RetinaNet [18], retain the abstract feature representations of objects throughout the network, and decode them into bounding boxes and classification scores at the end of the pipeline.

In addition to the same categorisation convention, HOI detectors need to localise two bounding boxes per instance instead of one. Early works [2, 8, 16, 23] employ a pre-trained object detector to obtain a set of human and object boxes, which are paired up exhaustively and processed by a downstream network for interaction classification. This methodology coincides with that of two-stage detectors and quickly became the mainstream approach due to the accessibility of high-quality pre-trained object detectors. The first instance of one-stage HOI detectors was introduced by
in Fig. 2. In contrast, two-stage HOI detectors do not re-
tain consumes large amounts of memory and time as shown 
has to be adapted to attend to regions of high interactive-
jectness by default. The heavy-weight decoder stack then 
trained weights, the decoder attends to regions of high ob-
features. In particular, when initialised with DETR’s pre-
bone networks for all methods have been initialised with the same 
notoriously difficult to train [20], to produce discriminative 
was greatly advanced. However, HOI detectors that adapt 
interaction points, represented as the midpoint of the human 
Table 1. The performance discrepancy between existing state-
of-the-art one-stage and two-stage HOI detectors is largely 
QC [25] one-stage DETR R-50 29.07
QPIC [25] one-stage DETR R-101 29.90
Ours two-stage DETR R-50 31.66
Ours two-stage DETR R-101 32.21

Liao et al. [17]. They characterised human–object pairs as 
summarises large amounts of memory and time as shown 
in a significant improvement, putting it on par with a state-
of-the-art one-stage detector (QPIC), as shown in Tab. 1. 
Liao et al. [17]. They characterised human–object pairs as 
leakage due to their representation power, flexibility, 
and object box centres. Recently, due to the great success 
in using learnable queries in transformer decoders for local- 
and global receptive field via the attention mechanism. The 
computer vision due to their representation power, flexibility, 
Transformer networks [27], initially developed for ma-
framed from the downstream interaction head during train-
ings, its weights can be frozen, and a lighter-weight network 
preferentially increases the predicted interaction scores for 
positive examples, while the pairwise encoder layer sup-
the issue, resulting in procedural misalignment.

We propose a two-stage model that refines the output 
features from DETR with additional transformer layers for 
HOI classification. As shown in Fig. 1, we encode the 
iinstance information in two ways: a unary representation 
where individual human and object instances are encoded 
separately, and a pairwise representation where human– 
object pairs are encoded jointly. These representations 
provide orthogonal information, and we observe different 
behaviours in their associated layers. The unary encoder layer 
preferentially increases the predicted interaction scores for 
positive examples, while the pairwise encoder layer sup-
presses the negative examples. As a result, this comple-
mentary behaviour widens the gap between scores of posi-
tive and negative examples, particularly benefiting ranking 
metrics such as mean average precision (mAP).

Our primary contribution is a novel and efficient two-
stage HOI detector with unary and pairwise encodings. Our 
secondary contribution is demonstrating how pairwise box 
positions—critical for HOI detection—can be incorpo-
ated into a transformer architecture, enabling it to 
jointly reason about unary appearance and pairwise spa-
tial information. We further provide a detailed analysis 
on the behaviour of the two encoder layers, showing that 
they have complementary properties. Our proposed model 
not only outperforms state-of-the-art methods, but also con-
sumes much less time and memory to train. The latter al-
low us to employ more memory-intensive backbone net-
works, further improving the performance.

2. Related work

Transformer networks [27], initially developed for ma-
chine translation, have recently become ubiquitous in com-
puter vision due to their representation power, flexibility, 
and global receptive field via the attention mechanism. The 
image transformer ViT [4] represented an image as a set 
of spatial patches, each of which was encoded as a to-
kton through simple linear transformations. This approach 
for tokenising images rapidly gained traction and inspired
to a new class of one-stage detection models that formulate the interaction decoder to be used alongside the DETR instance decoder. It is specifically responsible for predicting the action while also matching the interactive human–object pairs. These aforementioned one-stage detectors have achieved tremendous success in pushing the state-of-the-art performance. However, they all require significant resources to train the models. In contrast, this work focuses on exploiting novel ideas to produce equally discriminative features while preserving the memory efficiency and low training time of two-stage detectors.

Two-stage HOI detectors have also undergone significant development recently. Li et al. [15] studied the integration and decomposition of HOIs in an analogy to the superposition of waves in harmonic analysis. Hou et al. explored few-shot learning by fabricating object representations in feature space [12] and learning to transfer object affordance [11]. Finally, Zhang et al. [28] proposed to fuse features of different modalities within a graphical model to produce more discriminative features. We make use of this modality fusion in our transformer model and show that it leads to significant improvements.

3. Unary–pairwise transformers

To leverage the success of transformer-based detectors, we use DETR [1] as our backbone object detector and focus on designing an effective and efficient interaction head for HOI detection, as shown in Fig. 3. The interaction head consists of two types of transformer encoder layers, with the first layer modified to accommodate additional pairwise input. The first layer operates on unary tokens, i.e., individual human and object instances, while the second layer operates on pairwise tokens, i.e., human–object pairs. Based on our analysis and experimental observations in Sec. 4.3 and Sec. 4.4, self-attention in the unary layer preferentially increases the interaction scores for positive HOI pairs, whereas self-attention in the pairwise layer decreases the scores for negative pairs. As such, we refer to these layers as cooperative and competitive layers respectively.

3.1. Cooperative layer

A standard transformer encoder layer takes as input a set of tokens and performs self-attention. Positional encodings are usually indispensable to compensate for the lack of order in the token set. Typically, sinusoidal functions of the position [27] or learnable embeddings [1] are used for this purpose. It is possible to extend sinusoidal encodings to bounding box coordinates, however, our unary tokens already contain positional information, since they were decoded into bounding boxes. Instead, we take this as an opportunity to inject pairwise spatial information into the transformer, something that has been shown to be helpful for the task of HOI detection [28]. Specifically, we com-
compute the unary and pairwise spatial features used by Zhang et al. [28] from the bounding boxes, including the unary box centre, width and height, and pairwise intersection-over-union, relative area, and direction, and pass this through an MLP to obtain the pairwise positional encodings. We defer the full details to the appendix. We also found that the usual additive approach did not perform as well for our positional encodings. So we slightly modified the attention operation in the transformer encoder layer to allow directly injecting the pairwise positional encodings into the computation of values and attention weights.

More formally, given the detections returned by DETR, we first apply non-maximum suppression and thresholding. This leaves a smaller set \( \{d_i\}_{i=1}^n \), where a detection \( d_i = (b_i, s_i, c_i, x_i) \) consists of the box coordinates \( b_i \in \mathbb{R}^4 \), the confidence score \( s_i \in [0, 1] \), the object class \( c_i \in K \) for a set of object categories \( K \), and the object query or feature \( x_i \in \mathbb{R}^m \). We compute the pairwise box positional encodings \( \{y_{i,j} \in \mathbb{R}^m\}_{i,j=1}^n \) as outlined above. We denote the collection of unary tokens by \( X \in \mathbb{R}^{n \times m} \) and the pairwise positional encodings by \( Y \in \mathbb{R}^{n \times n \times m} \). The complete structure of the modified transformer encoder layer is shown in Fig. 4. For brevity of exposition, let us assume that the number of heads \( h \) is 1, and define

\[
\hat{X} \in \mathbb{R}^{n \times n \times m}, \quad \hat{X}_i \triangleq X_i \in \mathbb{R}^{n \times m},
\]

\[
\tilde{X} \in \mathbb{R}^{n \times n \times 2m}, \quad \tilde{x}_{i,j} \triangleq (x_i \oplus x_j) \in \mathbb{R}^{2m},
\]

where \( \oplus \) denotes elementwise product and \( w \in \mathbb{R}^{3m} \) and \( b \in \mathbb{R} \) are the parameters of the linear layer. The output of the attention layer is then computed as \( W \odot V \). Additional details can be found in the appendix.

### 3.2. Competitive layer

To compute the set of pairwise tokens, we form all pairs of distinct unary tokens and remove those where the first token is not human, as object–object pairs are beyond the scope of HOI detection. We denote the resulting set as \( \{p_k = (x_i, x_j, y_{i,j}) \mid i \neq j, c_i = \text{“human”}\} \). We then compute the pairwise tokens from the unary tokens and positional encodings via multi-branch fusion (MBF) [28] as

\[
z_k = \text{MBF}(x_i \oplus x_j, y_{i,j}).
\]

Specifically, the MBF module fuses two modalities in multiple homogeneous branches and return a unified feature representation. For completeness, full details are provided in the appendix. Last, the set of pairwise tokens are fed into an additional transformer encoder layer, allowing the network to compare the HOI candidates, before an MLP predicts each HOI pair’s action classification logits \( \bar{s} \).

### 3.3. Training and inference

To make full use of the pre-trained object detector, we incorporate the object confidence scores into the final scores of each human–object pair. Denoting the action logits of the \( k \)th pair \( p_k \) as \( \bar{s}_k \), the final scores are computed as

\[
s_k = (s_i)^{\lambda} \cdot (s_j)^{\lambda} \cdot \sigma(\bar{s}_k),
\]

where \( \lambda > 1 \) is a constant used during inference to suppress overconfident objects [28] and \( \sigma \) is the sigmoid function. We use focal loss\(^1\) [18] for action classification to counter the imbalance between positive and negative examples. Following previous practice [8, 28], we only compute the loss on valid action classes for each object type, specified by the dataset. During inference, scores for invalid combinations of actions and objects (e.g., eating a car) are zeroed out.

### 4. Experiments

In this section, we first demonstrate that the proposed unary–pairwise transformer achieves state-of-the-art performance on both the HICO-DET [2] and V-COCO [7] datasets, outperforming the next best method by a significant margin. We then provide a thorough analysis on the effects of the cooperative and competitive layers. In particular, we show that the cooperative layer increases the scores of positive examples while the competitive layer suppresses

\(^1\)Final scores in Eq. (6) are normalised to the interval \([0, 1]\). In training, we instead recover the scale prior to normalisation and use the corresponding loss with logits for numerical stability. See details in the appendix.
Table 2. Comparison of HOI detection performance (mAP × 100) on the HICO-DET [2] and V-COCO [7] test sets. The highest result in each section is highlighted in bold.

| Method       | Backbone     | Default Setting | Known Objects Setting | V-COCO          |
|--------------|--------------|-----------------|-----------------------|-----------------|
|              |              | Full  Rare Non-rare | Full  Rare Non-rare | AP_{role}^{S1}  | AP_{role}^{S2}  |
| HO-RCNN [2]  | CaffeNet     | 7.81  5.37 8.54 | 10.41 8.94 10.85     | -               | -               |
| InteractNet [6] | ResNet-50-FPN | 9.94  7.16 10.77 | - - - | -               | -               |
| GPNN [23]    | ResNet-101   | 13.11 9.34 14.23 | - - - | 40.0 -          | 44.0 -          |
| TIN [16]     | ResNet-50    | 17.03 13.42 18.11 | 19.17 15.51 20.26   | 47.8 54.2       |
| Gupta et al. [8] | ResNet-152 | 17.18 12.17 18.68 | - - - | - -             | - -             |
| VSGNet [26]  | ResNet-152   | 19.80 16.05 20.91 | - - - | 51.8 57.0       |
| DJ-RN [14]   | ResNet-50    | 21.34 18.53 22.18 | 23.69 20.64 24.60   | - -             |
| PPDM [17]    | Hourglass-104| 21.94 13.97 24.32 | 24.81 17.09 27.12   | - -             |
| VCL [10]     | ResNet-50    | 23.63 17.21 25.55 | 25.98 19.12 28.03   | 48.3 -          |
| ATL [11]     | ResNet-50    | 23.81 17.43 27.42 | 27.38 22.09 28.96   | 51.8 57.0       |
| DRG [5]      | ResNet-50-FPN| 24.53 19.47 26.04 | 27.98 23.11 29.43   | 51.0 -          |
| IDN [15]     | ResNet-50    | 24.58 20.33 25.86 | 27.89 23.64 29.16   | 53.3 60.3       |
| HOTR [13]    | ResNet-50    | 25.10 17.34 27.42 | - - - | 55.2 64.4       |
| FCL [12]     | ResNet-50    | 25.27 20.57 26.67 | 27.71 22.34 28.93   | 52.4 -          |
| AS-Net [3]   | ResNet-50    | 26.61 19.15 28.84 | 29.13 20.98 31.57   | 52.9 -          |
| SCG [28]     | ResNet-50-FPN| 28.87 24.25 30.25 | 31.74 27.07 33.14   | 53.9 -          |
| QPIC [25]    | ResNet-101   | 29.26 24.61 30.65 | 32.87 27.89 34.35   | 54.2 60.9       |
| Ours (UPT)   | ResNet-50    | 31.66 25.94 33.36 | 35.05 29.27 36.77   | 59.0 64.5       |
| Ours (UPT)   | ResNet-101   | 32.31 28.55 33.44 | 35.65 31.60 36.86   | 60.7 66.2       |
| Ours (UPT)   | ResNet-101-DC5| 32.62 28.62 33.81 | 36.08 31.41 37.47   | 61.3 67.1       |

Datasets: HICO-DET [2] is a large-scale HOI detection dataset with 37,633 training images, 9,546 test images, 80 object types, 117 actions, and 600 interaction types. The dataset has 117,871 human–object pairs with annotated bounding boxes in the training set and 33,405 in the test set. V-COCO [7] is much smaller in scale, with 2,533 training images, 2,867 validation images, 4,946 test images, and only 24 different actions.

4.1. Implementation details

We fine-tune the DETR model on the HICO-DET and V-COCO datasets prior to training and then freeze its weights. For HICO-DET, we use the publicly accessible DETR models pre-trained on MS COCO [19]. However, for V-COCO, as its test set is contained in the COCO val2017 subset, we first pre-train DETR models from scratch on MS COCO, excluding those images in the V-COCO test set. For the interaction head, we filter out detections with scores lower than $0.2$, and sample at least $3$ and up to $15$ humans and objects each, prioritising high scoring ones. For the hidden dimension of the transformer, we use $m = 256$, the same as DETR. Additionally, we set $\lambda$ to 1 during training and 2.8 during inference [28]. For the hyperparameters used in the focal loss, we use the same values as SCG [28].

We apply a few data augmentation techniques used in other detectors [1, 25]. Inputs images are scaled such that the shortest side is at least 480 and at most 800 pixels. The longest side is limited at 1333 pixels. Additionally, each image is cropped with a probability of 0.5 to a random rectangle with each side being at least 384 pixels and at most 600 pixels before being scaled. We also apply colour jittering, where the brightness, contrast and saturation values are adjusted by a random factor between 0.6 to 1.4. We use AdamW [22] as the optimiser with an initial learning rate of $10^{-4}$. All models are trained for 20 epochs with a learning rate reduction at the 10th epoch by a factor of 10. Training is conducted on 8 GeForce GTX TITAN X devices, with a batch size of 2 per GPU—an effective batch size of 16.

4.2. Comparison with state-of-the-art methods

The performance of our model is compared to existing methods on the HICO-DET [2] and V-COCO [7] datasets in Tab. 2. There are two different settings for evaluation on HICO-DET. Default Setting: A detected human–object
Table 3. Comparing the effect of the cooperative (coop.) and competitive (comp.) layers on the interaction scores. We report the change in the interaction scores as the layer in the $\Delta$ Architecture column is added to the reference network, for positives, easy negatives and hard negatives, with the number of examples in parentheses. As indicated by the bold numbers, the cooperative layer significantly increases the scores of positive examples while the competitive layer suppresses hard negative examples. Together, these layers widen the gap between scores of positive and negative examples, improving the detection mAP.

| Reference                  | $\Delta$ Architecture | $\Delta$ Positives (25,391) | $\Delta$ Easy Negatives (3,903,416) | $\Delta$ Hard Negatives (510,991) |
|----------------------------|------------------------|------------------------------|-------------------------------------|-----------------------------------|
|                            |                        | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| Ours w/o coop. layer       | + coop. layer          | +0.1487 | +0.1078 | +0.0001 | +0.0000 | +0.0071 | +0.0000 |
| Ours w/o comp. layer       | + comp. layer          | -0.0463 | -0.0310 | -0.0096 | -0.0024 | -0.1080 | -0.0922 |
| Ours w/o both layers       | + both layers          | +0.0799 | +0.0390 | -0.0076 | -0.0018 | -0.0814 | -0.0748 |

Figure 5. Change in the interaction score (delta) with respect to the reference score. (a) The distribution of score deltas when adding the cooperative layer (first row of Tab. 3). (b) Adding the competitive layer to the model (second row). (c) Adding both layers (last row). For visualisation purposes, only 20% of the negatives are sampled and displayed.

4.3. Macroscopic effects of the interaction head

In this section, we compare the effects of the unary (cooperative) and pairwise (competitive) layers on the HICO-DET test set, with ResNet50 [9] as the CNN backbone. Since the parameters in the object detector are kept frozen for our model, the set of detections processed by the downstream network remains the same, regardless of any architectural changes in the interaction head. This allows us to compare how different variants of our model perform on the same human–object pairs. To this end, we collected the predicted interaction scores for all human–object pairs over the test set and compare how adding certain layers influence them. In Tab. 3, we show some statistics on the change of scores upon an architectural modification. In particular, note that the vast majority of collected pairs are easy negatives with scores close to zero. For analysis, we divide the negative examples into easy and hard, where we define an easy negative as one with a score lower than 0.05 as predicted by the “Ours w/o both layers” model, which accounts for a small memory footprint. This allows us to use a higher-resolution feature map by removing the stride in the 5th convolutional block (C5) of ResNet [9], which has been shown to improve detection performance on small objects [1]. We denote this as dilated C5 (DC5).
Table 4. Effect of the cooperative and competitive layers on the HICO-DET test set under the default settings.

| Model                                      | Full     | Rare    | Non-rare |
|--------------------------------------------|----------|---------|----------|
| Ours w/o both layers                       | 29.22    | 23.09   | 31.05    |
| Ours w/o comp. layer                       | 30.78    | 24.92   | 32.53    |
| Ours w/o coop. layer                       | 30.68    | 24.69   | 32.47    |
| Ours w/o pairwise pos. enc.                | 29.98    | 23.72   | 31.64    |
| Ours (1× coop., 1× comp.)                  | 31.33    | 26.02   | 32.91    |
| Ours (1× coop., 2× comp.)                  | 31.62    | **26.18** | 33.24    |
| Ours (2× coop., 1× comp.)                  | **31.66** | 25.94   | **33.36**|

for 90% of the negative examples. In addition, we also show the distribution of the change in score with respect to the reference score as scatter plots in Fig. 5. The points are naturally bounded by the half-spaces $0 \leq x + y \leq 1$.

Notably, adding the cooperative layer results in a significant average increase (+0.15) in the scores of positive examples, with little effect on the negative examples. This can be seen in Fig. 5a as well, where the score changes for almost all positive examples are larger than zero. In contrast, adding the competitive layer leads to a significant average decrease (−0.11) in the scores of hard negative examples, albeit with a small decrease in the score of positive examples as well. This minor decrease is compensated by the cooperative layer as shown in the last row of Tab. 3. Furthermore, looking at Fig. 5b, we can see a dense mass near the line $y = -x$, which indicates that many negative examples have had their scores suppressed to zero.

Ablation study: In Tab. 4, we ablate the effect of different design decisions on performance. Adding the cooperative and competitive layers individually improves the performance by around 1.5 mAP, while adding both layers jointly improves by over 2 mAP. We also demonstrate the significance of the pairwise position encodings by removing them from the modified encoder and the multi-branch fusion module. This results in a 1.3 mAP decrease. Finally, we observe a slight improvement (0.3 mAP) when adding an additional cooperative or competitive layer, but no further improvements with more layers. As the competitive layer is more costly, we use two cooperative layers.

4.4. Microscopic effects of the interaction head

In this section, we focus on a specific image and visualise the effect of attention in our cooperative and competitive layers. In Fig. 6, we display a detection-annotated image and its associated attention map from the unary (cooperative) layer. The human–object pairs (1, 4), (2, 5) and (3, 6) are engaged in the interaction *riding a horse*. Excluding attention weights along the diagonal, we see that the corresponding human and horse instances attend to each other.

We hypothesise that attention between pairs of unary tokens (e.g., 1 and 4) helps increase the interaction scores for the corresponding pairs. To validate this hypothesis, we manually set the attention logits between the three positive pairs to minus infinity, thus zeroing out the corresponding attention weights. The effect of this was an average decrease of 0.06 (8%) in the interaction scores for the three pairs, supporting the hypothesis.

In Fig. 7, we visualise the attention map of the pairwise (competitive) layer. Notably, all human–object pairs attend to the interactive pairs (1, 4), (2, 5) and (3, 6) in decreasing order, except for the interactive pairs themselves. We hypothesise that attention is acting here to have the dominant pairs suppress the other pairs. To investigate, we manually set the weights such that the three interactive pairs all attend to (1, 4) as well, with a weight of 1. This resulted in a decrease of their interaction scores by 0.08 (11%). We then instead zeroed out the attention weights between the rest of the pairs and (1, 4), which resulted in a small increase in the scores of negative pairs. These results together suggest that attention in the competitive layer is acting as a soft version of non-maximum suppression, where pairs less likely to foster interactions attend to, and are suppressed by, the most dominant pairs. See appendix for more examples.
4.5. Qualitative results and limitations

In Fig. 8, we present several qualitative examples of successful HOI detections, where our model accurately localises the human and object instances and assigns high scores to the interactive pairs. For example, in Fig. 8b, our model correctly identifies the subject of an interaction (the lady in red) despite her proximity to a non-interactive human (the lady in black). We also observe in Fig. 8a that our model becomes less confident when there is overlap and occlusion. This stems from the use of object detection scores in our model. Confusion in the object detector often translates to confusion in action classification. We also show five representative failure cases for our model, illustrating its limitations. In Fig. 9a, due to the indefinite position of drivers in the training set (and real life), the model struggled to identify the driver. For Fig. 9d, the model failed to recognise the interaction due to a lack of training data (1 training example), even though the action is well-defined. Overall, ambiguity in the actions and insufficient data are the biggest challenges for our model. Another limitation, specific to our model, is that the computation and memory requirements of our pairwise layer scale quadratically with the number of unary tokens. For scenes involving many interactive humans and objects, this becomes quite costly. Moreover, since the datasets we used are limited, we may expect poorer performance on data in the wild, where image resolution, lighting condition, etc. may be less controlled.

5. Conclusion

In this paper, we have proposed a two-stage detector of human–object interactions using a novel transformer architecture that exploits both unary and pairwise representations of the human and object instances. Our model not only outperforms the current state-of-the-art—a one-stage detector—but also consumes much less time and memory to train. Through extensive analysis, we demonstrate that attention between unary tokens acts to increase the scores of positive examples, while attention between pairwise tokens acts like non-maximum suppression, reducing the scores of negative examples. We show that these two effects are complementary, and together boost performance significantly.

Potential negative societal impact: Transformer models are large and computationally-expensive, and so have a significant negative environmental impact. To mitigate this, we use pre-trained models and a two-stage architecture, since fine-tuning an existing model requires less resources, as does training a single stage with the other stage fixed. There is also the potential for HOI detection models to be misused, such as for unauthorised surveillance, which disproportionately affects minority and marginalised communities.

Acknowledgments: We are grateful for support from Continental AG (D.C.). We would also like to thank Jia-Bin Huang and Yuliang Zou for their help with the reproduction of some experiment results.
References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Eur. Conf. Comput. Vis., 2020. 2, 3, 5, 6

[2] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2018. 1, 2, 4, 5

[3] Mingfei Chen, Yue Liao, Si Liu, Zhiyuan Chen, Fei Wang, and Chen Qian. Reformulating hoi detection as adaptive set prediction. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021. 3, 5

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In Int. Conf. Learn. Represent., 2021. 2

[5] Chen Gao, Jiawu X, Yuliang Zou, and Jia-Bin Huang. DRG: Dual relation graph for human-object interaction detection. In Eur. Conf. Comput. Vis., 2020. 5

[6] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In IEEE Conf. Comput. Vis. Pattern Recogn., 2018. 5

[7] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. arXiv preprint arXiv:1505.04474, 2015. 4, 5

[8] Tanmay Gupta, Alexander Schwing, and Derek Hoiem. No-frills human-object interaction detection: Factorization, layout encodings, and training techniques. In Int. Conf. Comput. Vis., 2019. 1, 4, 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conf. Comput. Vis. Pattern Recogn., pages 770–778, 2016. 5, 6

[10] Zhi Hou, Xiaojing Peng, Yu Qiao, and Dacheng Tao. Visual compositional learning for human-object interaction detection. In Eur. Conf. Comput. Vis., 2020. 5

[11] Zhi Hou, Baosheng Yu, Yu Qiao, Xiaojing Peng, and Dacheng Tao. Affordance transfer learning for human-object interaction detection. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021. 3, 5

[12] Zhi Hou, Baosheng Yu, Yu Qiao, Xiaojing Peng, and Dacheng Tao. Detecting human-object interaction via fabricated compositional learning. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021. 3, 5

[13] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J. Kim. Hotr: End-to-end human-object interaction detection with transformers. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021. 3, 5

[14] Yong-Lu Li, Xinpeng Liu, Han Lu, Shiyi Wang, Junqi Liu, Jiefeng Li, and Cewu Lu. Detailed 2d-3d joint representation for human-object interaction. In IEEE Conf. Comput. Vis. Pattern Recogn., 2020. 5

[15] Yong-Lu Li, Xinpeng Liu, Xiaqian Wu, Yizhuo Li, and Cewu Lu. Hoi analysis: Integrating and decomposing human-object interaction. In Adv. Neural Inform. Process. Syst., 2020. 3, 5

[16] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yanfeng Wang, and Cewu Lu. Transferable interactivity knowledge for human-object interaction detection. In IEEE Conf. Comput. Vis. Pattern Recogn., 2019. 1, 5

[17] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. PPD: Parallel point detection and matching for real-time human-object interaction detection. In IEEE Conf. Comput. Vis. Pattern Recogn., 2020. 2, 5

[18] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Int. Conf. Comput. Vis., 2017. 1, 4

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In Eur. Conf. Comput. Vis., 2014. 5

[20] Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020. 2

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In Int. Conf. Comput. Vis., pages 10012–10022, October 2021. 3

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In Int. Conf. Learn. Represent., 2018. 5

[23] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In Eur. Conf. Comput. Vis., 2018. 1, 5

[24] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Adv. Neural Inform. Process. Syst., pages 91–99, 2015. 1, 2

[25] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. QPIC: Query-based pairwise human-object interaction detection with image-wide contextual information. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021. 2, 3, 5

[26] Oytun Ulutan, A S M Iftekhar, and B. S. Manjunath. VSGNet: Spatial attention network for detecting human object interactions using graph convolutions. In IEEE Conf. Comput. Vis. Pattern Recogn., 2020. 5

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Adv. Neural Inform. Process. Syst., volume 30, 2017. 2, 3, 4

[28] Frederic Z. Zhang, Dylan Campbell, and Stephen Gould. Spatially conditioned graphs for detecting human-object interactions. In Int. Conf. Comput. Vis., pages 13319–13327, October 2021. 2, 3, 4, 5

[29] Cheng Zhou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, and Jian Sun. End-to-end human object interaction detection with hoi transformer. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021. 3, 5