HOI as Embeddings: Advancements of Model Representation Capability in Human-Object Interaction Detection

Junwen Chen Yingcheng Wang Keiji Yanai
Department of Informatics
The University of Electro-Communications
Tokyo, Japan
{chen-j, wang-y, yanai}@mm.inf.uec.ac.jp

***Abstract*—In recent years, human-object interaction detection (HOID) has attracted increasing attention in the computer vision community and has been greatly advanced by the introduction of transformer-based models. However, the representation capability of the pre-trained object detection modelis is insufficient for capturing the complex interactions between humans and objects, which limits the performance of HOID methods. In this paper, we introduce three methods to progressively enhance the representation capability. (1) We propose QAHOI to take advantage of multi-scale feature maps with different spatial scales. (2) We propose PQNet to speed up training convergence with parallel queries. (3) We propose SOV-STG to combine the merits of QAHOI and PQNet and introduce the denoising learning strategy and vision language model to further improve training convergence and performance. Our proposed method SOV-STG achieves state-of-the-art performance on the HICO-Det dataset with one-third of the training epochs compared to previous SOTA methods.**

***Index Terms*—Human-Object Interaction, Transformer**

I. Introduction

Human-object interaction detection (HOID) task as a downstream task of object detection is largely dependent on the pre-trained object detection model. HOID models are required to predict HOI instances by detecting pairs of humans and objects and recognizing the interactions between them. From first of the beginning, CNN-based two-stage HOID methods [1]–[10] leverage the off-the-shelf object detector [11], [12] to detect humans and objects, and then introduce additional branches or modules to recognize the interactions between each pair of them. To improve the efficiency of HOID models, CNN-based one-stage methods [13]–[15] are proposed to detect human-object pairs and recognize their interactions at the same time.

Recently, vision transformer [16] has been successfully applied to various computer vision tasks. The self-attention mechanism in the transformer can capture long-range dependencies and global context information, which is crucial for the HOID task. Thus, transformer-based one-stage HOID methods [17]–[20] built upon the transformer-based object detector [21] are proposed to extract more context information of HOI pairs from the image. However, there are problems such as the slow training convergence and the high computational cost in DETR, which limits the performance of transformer-based HOID methods. Recently, various methods have been proposed to improve DETR [22]–[24]. Similarly, HOI detection methods can achieve higher accuracy by improving the detection architecture. In this paper, we start from the most basic transformer-based HOID model [17], [18] and reduce the learning cost and improve the accuracy in the three following works: QAHOI [25], PQNet [26] and SOV-STG [27].

II. Related Work

Transformer-based HOID methods view the HOID as a set prediction problem, which does not need a matching process between human and object proposals. QPIC [17] and HOITrans [18] add the human box FFN (Feed-forward Network) head and the interaction class FFN head to DETR. Thus, each query embedding can represent an entire HOI instance <*Human Box, Object Box with Category, Interaction Category*> during the decoding process. However, the direct adaptation of DETR to the HOID task has some limitations. The decoder's training burden is heavy, and the performance is limited. Subsequent works [28]–[31] propose additional decoders or modules to facilitate the interaction recognition process. CDN [28] adopts a cascade architecture with an additional interaction decoder to disentangle the interaction recognition process from the object detection process. GEN-VLKT [32] improves the decoding process of CDN and introduces a learning strategy to transfer the prior knowledge of CLIP [33] to the HOID model. In this paper, we focus on improving the representation capability from both the detection framework and the decoding process from the input query to the output prediction heads.

Recently, DN-DETR [24] proposes a denoising learning method with ground-truth information for the DETR-based method and improves the efficiency of learning. Similarly, in the HOID task, DOQ [34] and HQM [35] propose to use the ground-truth label or box information to guide the learning process. However, without a specific denoising target, the learning process is still inefficient. In this paper, we introduce an HOI-specific denoising learning strategy and efficiently reduce the training burden of the HOID model.

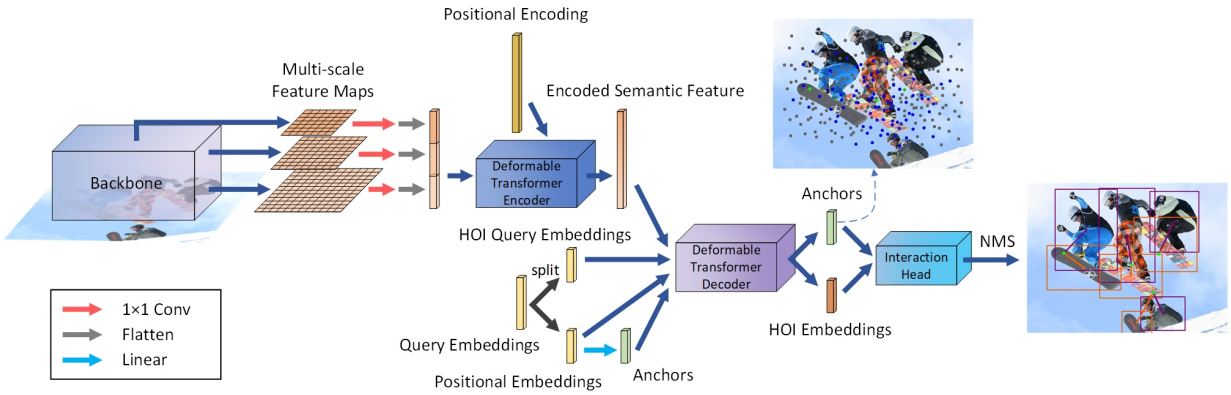

Fig. 1: Overview architecture of QAHOI.

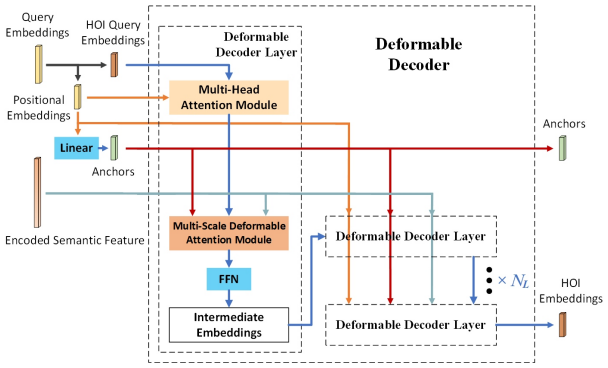

Fig. 2: Decoding process of QAHOI.

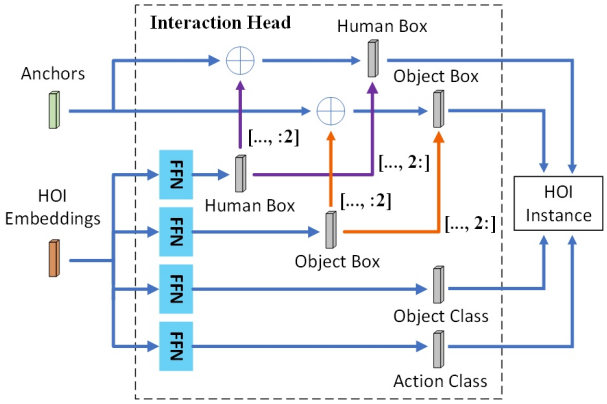

Fig. 3: Anchor-based HOI heads.

III. Method

In this section, we will introduce the motivations, the designs, and the advantages of our proposed methods. We first explore the detection framework and the representation of query embeddings (in Section III-A, QAHOI [25]), and then show the importance of the disentanglement of the human-object detection with parallel queries (in Section III-B, PQNet [26]), and finally, we extend the idea of QAHOI and PQNet to a more advanced method with optimized architecture and learning strategy (in Section III-C, SOV-STG [27]).

A. QAHOI

Transformer-based HOID method QPIC [17] surpasses the previous CNN-based HOID method [13], [15] by a large margin. However, due to the transformer's quadratic computational complexity, it only uses low-resolution feature maps from the last layer of a CNN backbone. The small and overlapping objects commonly exist in HOI images, and the low-resolution feature maps are not sufficient to capture the detailed information about these objects. Deformable DETR [22] (DDETR) designs a Deformable Multi-Scale Attention Module to reduce the complexity of attention in DETR according to the spatial size and achieves multi-scale Transformer-based object detection. Following the idea of DDETR, we propose a multi-scale anchor-based HOI detection method **Q**uery-based **A**nchors for **H**uman-**O**bject **I**nteraction Detection (QAHOI) [25].

Multi-scale feature extractor. QPIC [17] follows DETR [21] to construct a feature extractor consisting of a CNN backbone and a transformer encoder, and uses low-resolution feature maps from the backbone, making it difficult to extract small-scale spatial information. The query embeddings used to represent the HOI instances are refined during the decoding process through cross-attention with the global context feature map. Improving the representation of the global context feature map can effectively improve the performance of the HOID model, thus, QAHOI combines a hierarchical backbone and a deformable transformer encoder [22] to build a multi-scale feature extractor, as shown in Fig. 1. It can use CNN-based (ResNet [36]) or transformer-based backbones (Swin-Transformer [37]). Specifically, QAHOI uses the feature maps of the last three stages of the backbone, $x_1 \in \mathbb{R}^{2C_s \times \frac{H}{8} \times \frac{W}{8}}$, $x_2 \in \mathbb{R}^{4C_s \times \frac{H}{16} \times \frac{W}{16}}$, and $x_3 \in \mathbb{R}^{8C_s \times \frac{H}{32} \times \frac{W}{32}}$. The feature maps x_1, x_2, and x_3 are projected from C_s dimensions to $C_d = 256$ dimensions using 1×1 convolutional layers.

Anchor-based decoding. The decoding process of the deformable transformer decoder is shown in Fig. 2. The query embedding of QAHOI is divided into the HOI query embed-

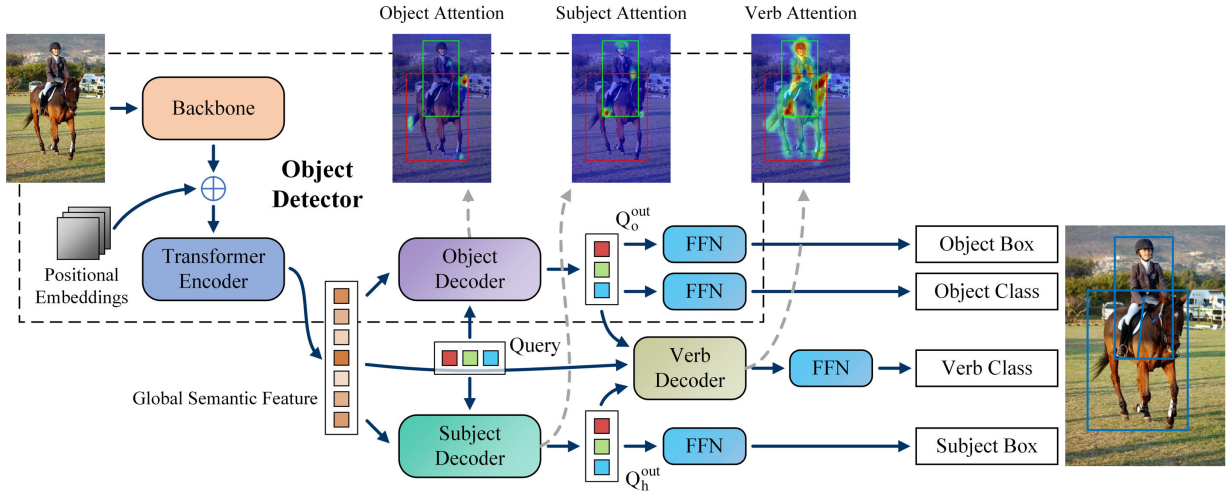

Fig. 4: Overview architecture of PQNet.

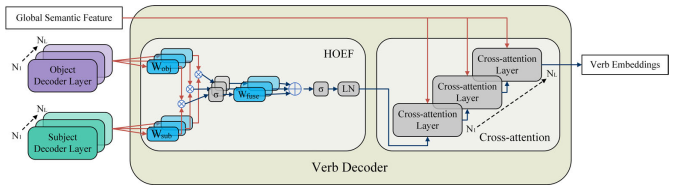

Fig. 5: Verb decoder of PQNet.

ding $Q_{HOI} \in \mathbb{R}^{N_q \times C_d}$ and the position embedding $Q_{Pos} \in \mathbb{R}^{N_q \times C_d}$, where N_q is the number of queries. The anchor $P \in \mathbb{R}^{N_q \times 2}$ is generated from the position embedding Q_{Pos} through a linear layer. With the encoded semantic feature from the deformable transformer encoder, the HOI query embedding Q_{HOI}, and the anchor P, the HOI embedding $E \in \mathbb{R}^{N_q \times C_d}$ is decoded by the attention mechanism of the deformable transformer decoder. Self-attention and multi-scale deformable attention [22] are calculated N_L times in N_L decoder layers, and the HOI embedding for predicting HOI instances is output in the last layer. For the HOID task, with explicit guidance from the anchor, the HOI embedding can capture more detailed information from the multi-scale feature maps, and the representation capability is enhanced.

Anchor-based HOI detection heads. Similar to QPIC, QAHOI uses additional heads to predict the human box and the interaction class. However, different from QPIC, the prediction burden of localizing is shared by the anchor. As shown in Fig. 3, each anchor (p_x, p_y) of the anchor set $P \in \mathbb{R}^{N_q \times 2}$ is used as a reference point for the human-object pair box. In the interaction head, the box elements B^h and $B^o \in \mathbb{R}^{N_q \times 4}$ of the human and object predicted by the Feed-forward Network (FFN) are composed of $\{d_x, d_y, w, h\}$. The final bounding boxes $\hat{B}^h$ and $\hat{B}^o$ are composed of $\{d_x + p_x,\ d_y + p_y,\ w,\ h\}$. Finally, the object class $O \in \mathbb{R}^{N_q \times K_o}$ of the object box and the interaction class $A \in \mathbb{R}^{N_q \times K_a}$ of the HOI instance are combined with the human and object bounding boxes $\hat{B}^h$, $\hat{B}^o$ to construct the output of the HOI instance.

B. PQNet

QPIC [17], HOITrans [18], and subsequent work CDN [28] use the same query embeddings to represent the entire HOI instance. The decoded target embeddings are used to represent both object and human features simultaneously. However, using highly integrated embeddings to simultaneously locate humans and objects is not optimal. By separating human and object detection, higher accuracy can be achieved. To this end, we propose the **P**arallel **Q**uery Network (PQNet) [26], which divides the decoding process into human decoding and object decoding and detects humans and objects in parallel using parallel queries. As shown in Fig. 4, PQNet uses two transformer decoders to decode human embeddings and object embeddings in parallel. PQNet uses two Transformer decoders to decode human embeddings and object embeddings in parallel. In addition, we introduce a verb decoder to fuse human embeddings and object embeddings and predict interactions. By optimizing the detection part and advancements of verb embeddings, PQNet outperforms QPIC in less than half the number of training epochs.

Object detector and feature extractor. As shown in Fig. 4, the object detector of PQNet is the same as DETR. The CNN backbone and the transformer encoder constitute the feature extractor. Given an input image $I \in \mathbb{R}^{3 \times H \times W}$, the CNN backbone extracts the appearance feature map $f \in \mathbb{R}^{8C_s \times \frac{H}{32} \times \frac{W}{32}}$. The feature extractor outputs the global semantic feature $S \in \mathbb{R}^{N_s \times C_d}$ ($N_s = \frac{H}{32} \times \frac{W}{32}$). In the object decoder, the global semantic feature S is used as the source input, and the query is used by the decoder to extract object embeddings.

Detection with parallel queries. The basic idea of PQNet is to use parallel queries to split the detection process. Specifically, we introduce a subject (human) decoder with the same architecture as the object decoder. The object decoder and the subject decoder share the HOI query embedding $Q \in \mathbb{R}^{N_q \times C_d}$

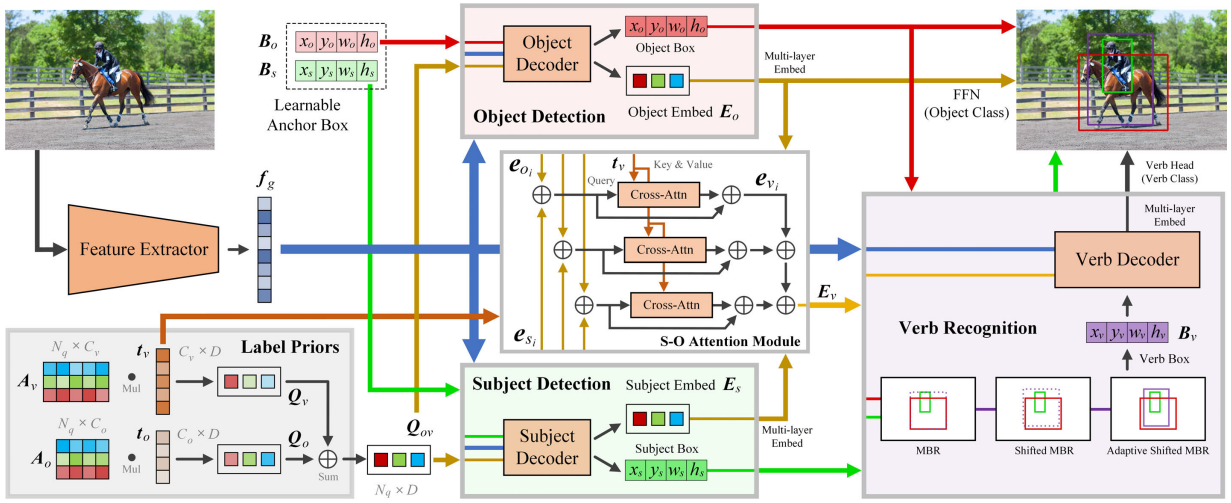

Fig. 6: Overview architecture of SOV-STG.

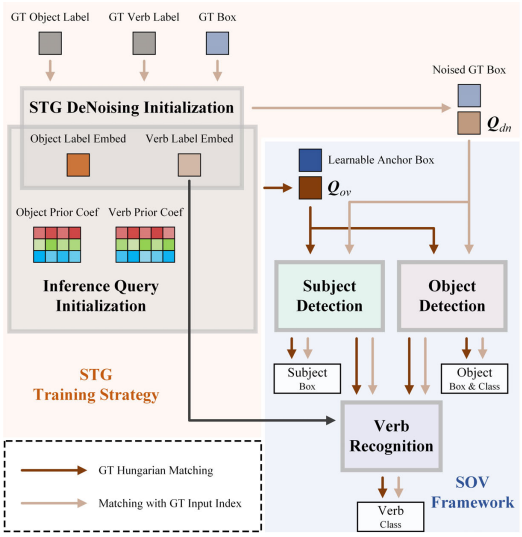

Fig. 7: End-to-end training pipeline of SOV-STG.

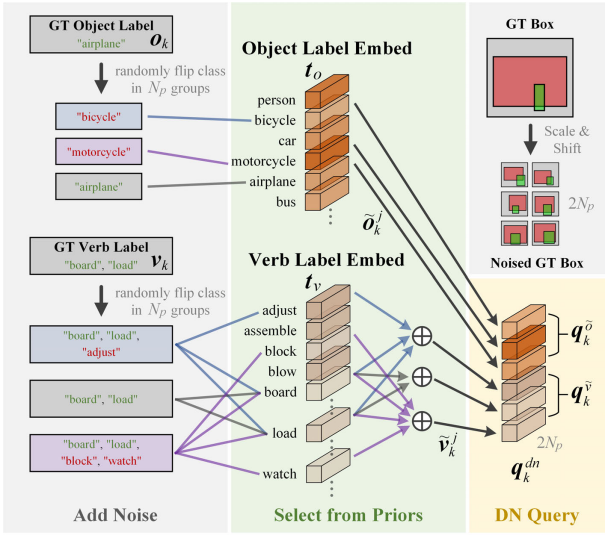

Fig. 8: DN query initialization.

and output the object embedding $Q_o \in \mathbb{R}^{N_q \times C_d}$ and the subject embedding $Q_h \in \mathbb{R}^{N_q \times C_d}$, respectively. Next, the object embedding is used to predict the object box $B^{(o)} \in \mathbb{R}^{N_q \times 4}$ and the object category $C_o \in \mathbb{R}^{N_q \times K_o}$, where K_o is the number of object categories. The subject embedding is used to predict the human box $B^{(h)} \in \mathbb{R}^{N_q \times 4}$. Since the same pipeline is used to predict object instances and subject instances, the part of subject detection with a feature extractor can be considered as a subject detector.

Verb decoder. With the additional semantic information extracted by our subject decoder, we introduce a verb decoder to fuse the object embeddings and the subject embeddings and integrate the global semantic features. As shown in Fig. 5, our verb decoder computes two types of attention mechanisms sequentially using two modules. The Human-Object Embedding Fusion (HOEF) module fuses the embeddings of the last layers of the object decoder and the subject decoder to output the verb embedding. Specifically, given the object embedding Q_o and subject embedding Q_h obtained, the verb embedding Q_v fused by HOEF is calculated as follows:

$$\begin{aligned} Q_v &= \text{HOEF}(Q_o, Q_h) = \text{LN}(\sigma(\text{Sum}(\text{Head}^{(1)}, ..., \text{Head}^{(n)}))), \\ \text{Head}^{(i)}(Q_o, Q_h) &= W_{fuse}^{(i)} \sigma(W_o^{(i)} Q_o \circ W_h^{(i)} Q_h) \end{aligned} \tag{1}$$

where $W_o, W_h \in \mathbb{R}^{K_d \times K_s}$, and $W_{fuse} \in \mathbb{R}^{C_d \times C_m}$ ($C_m = \frac{C_d}{n}$) are projection weight matrices, n is the number of attention heads, LN is the layer normalization, σ is the activation function, and $\circ$ is the element-wise product. Then, the cross-attention module computes the attention between the verb embedding Q_v and the global semantic feature S. Finally, the refined verb embeddings in the last layer of the cross-attention module is used to predict the verb (interaction) class of the HOI instance.

Method	Epoch	Backbone	Default			Known Object		
			Full	*Rare*	*Non-Rare*	*Full*	*Rare*	*Non-Rare*
QPIC [17]	150	ResNet-50	29.07	21.85	31.23	31.68	24.14	33.93
CDN-S [28]	100	ResNet-50	31.44	27.39	32.64	34.09	29.63	35.42
CDN-B [28]	100	ResNet-50	31.78	27.55	33.05	34.53	29.73	35.96
CDN-L [28]	100	ResNet-101	32.07	27.19	33.53	34.79	29.48	36.38
PQNet-S [26]	70	ResNet-50	31.92	28.06	33.08	34.58	30.71	35.74
PQNet-B [26]	100	ResNet-50	32.13	29.43	32.93	34.68	32.06	35.47
PQNet-L [26]	100	ResNet-50	32.45	27.80	33.84	35.28	30.72	36.64
HQM (CDN-S) [35]	80	ResNet-50	32.47	28.15	33.76	35.17	30.73	36.50
RLIP-ParSe [38]	90	ResNet-50	32.84	34.63	26.85	-	-	-
MUREN [39]	100	ResNet-50	32.87	28.67	34.12	35.52	30.88	36.91
DOQ (CDN-S) [34]	80	ResNet-50	33.28	29.19	34.50	-	-	-
GEN-VLKT-S [32]	90	ResNet-50	33.75	29.25	35.10	36.78	32.75	37.99
HOICLIP [40]	90	ResNet-50	34.69	31.12	35.74	37.61	34.47	38.54
GEN-VLKT-M [32]	90	ResNet-101	34.78	31.50	35.77	38.07	34.94	39.01
GEN-VLKT-L [32]	90	ResNet-101	34.95	31.18	36.08	38.22	34.36	39.37
QAHOI-Swin-L [25]	150	Swin-Large-22K	35.78	29.80	37.56	37.59	31.36	39.36
FGAHOI-Swin-L [41]	190	Swin-Large-22K	37.18	30.71	39.11	38.93	31.93	41.02
DiffHOI-Swin-L [42]	90	Swin-Large-22K	41.50	39.96	41.96	43.62	41.41	44.28
SOV-STG-S	30	ResNet-50	33.80	29.28	35.15	36.22	30.99	37.78
SOV-STG-M	30	ResNet-101	34.87	30.41	36.20	37.35	32.46	38.81
SOV-STG-L	30	ResNet-101	35.01	30.63	36.32	37.60	32.77	39.05
SOV-STG-Swin-L	30	Swin-Large-22K	43.35	42.25	43.69	45.53	43.62	46.11

TABLE I: Comparison to the state-of-the-art on the HICO-DET.

C. SOV-STG

Our QAHOI improves the detection framework and the representation of query embeddings, and PQNet promotes the use of query embeddings and the fusion of verb embeddings. To combine these two advancements and further improve the learning efficiency, we propose **S**ubject **O**bject **V**erb (SOV) framework. To improve learning efficiency, we propose a denoise learning method **S**pecific **T**arget **G**uilded (STG) that introduces prior knowledge using learnable object and verb label embeddings.

Fig. 6 shows the architecture of the proposed method SOV-STG [27]. The design of the SOV framework is an extension of the multi-scale architecture in Section III-A and the parallel query idea in Section III-B. Specifically, we construct the architecture of object decoder, subject decoder, and verb decoder using anchor boxes to represent HOIs. We introduce a new **A**daptive **S**hifted **M**inimum **B**ounding **R**ectangle (ASMBR) to generate verb boxes from the output boxes of the object decoder and subject decoder. As shown in Fig. 7, with the explicit object and subject anchor boxes, the object and subject decoders obtain clear position denoising targets directly from the input by adding noise to the ground truth anchor boxes. We define two types of learnable label embeddings, object label prior and verb label prior. By defining label embeddings, the model can obtain label-specific information from the ground truth labels in both the training and inference stages. Besides, according to PQNet, we introduce a novel **S**ubject-**O**bject (S-O) attention module to fuse object and subject information and improve the verb representation learning ability. As a result, SOV-STG achieves SOTA performance with one-third of the training epochs compared to previous SOTA methods.

Prediction of HOI instances using anchor boxes. To clarify the decoding target of the query embedding, the SOV framework utilizes the attention mechanism of DAB-Deformable-DETR [23] and directly uses learnable anchor boxes for predicting human and object boxes. With the object and subject boxes, our proposed ASMBR generates verb boxes to provide the spatial relationship between human boxes and object boxes for our verb decoder. As shown in Fig. 7, given the predicted human box $\boldsymbol{B}_s = (x_s, y_s, w_s, h_s)$ and object box $\boldsymbol{B}_o = (x_o, y_o, w_o, h_o)$ (where (x, y) is the center of the box) from the final layer of the decoder, the verb box is defined as follows:

$$\boldsymbol{B}_v = \left(\frac{x_s + x_o}{2}, \frac{y_s + y_o}{2}, w_v, h_v \right) \tag{2}$$

$$w_v = \frac{w_s + w_o}{2} + |x_s - x_o|, h_v = \frac{h_s + h_o}{2} + |y_s - y_o| \tag{3}$$

ASMBR is a method to shrink and move the **M**inimum **B**ounding **R**ectangle (MBR). The purpose of the shrinkage (adaptive) and movement (shift) is to remove information with low relevance far from the interaction area and to cover more context information around the interaction area.

SOV decoders. It is important to design separate decoders to clarify the decoding targets. The same as QAHOI, a mult-scale feature extractor is adopted. Similar to PQNet, split decoders are used to decode object and subject embeddings and update

object and subject anchor boxes in parallel. Our SOV decoders share the same architecture and extract semantic features with the guidance of corresponding anchor boxes.

S-O attention module. As shown in Fig. 7, to integrate the prior knowledge of verb labels when fusing features, we fuse the verb label embedding with S-O attention. Furthermore, we design a bottom-up path for S-O attention to enhance information from lower layers to upper layers. Givin the i-th layer ($i > 1$) subject embedding $\boldsymbol{e}_{s_i} \in \mathbb{R}^{N_q \times C_d}$ and the object embedding $\boldsymbol{e}_{o_i} \in \mathbb{R}^{N_q \times C_d}$. The verb embedding $\boldsymbol{e}_{v_i}$ can be defined as follows:

$$\begin{aligned} \boldsymbol{e}_{v_i} =&((\text{CrossAttn}(\boldsymbol{e}_{so_{i-1}}, \boldsymbol{t}_v) + \boldsymbol{e}_{so_{i-1}}) \\ &+ (\text{CrossAttn}(\boldsymbol{e}_{so_i}, \boldsymbol{t}_v) + \boldsymbol{e}_{so_i}))/2 \end{aligned} \tag{4}$$

$$\boldsymbol{e}_{so_i} = (\boldsymbol{e}_{o_i} + \boldsymbol{e}_{s_i})/2 \tag{5}$$

Split label priors. We use two learnable label embeddings to initialize the query embedding of the SOV decoder. The object label embedding $\boldsymbol{t}_o \in \mathbb{R}^{C_o \times C_d}$, which consists of C_o C_d-dimensional vectors, is defined as prior knowledge of object labels. Similarly, the verb label embedding $\boldsymbol{t}_v \in \mathbb{R}^{C_v \times C_d}$ is defined as prior knowledge of verb labels. Using the prior knowledge of object and verb labels, we initialize the object label embedding $\boldsymbol{q}_o \in \mathbb{R}^{N_q \times C_d}$ and the verb label embedding $\boldsymbol{q}_v \in \mathbb{R}^{N_q \times C_d}$ by linearly combining them. Then, we add the object and verb label embeddings to obtain the inference query embedding $\boldsymbol{q}_{ov} \in \mathbb{R}^{N_q \times C_d}$. The linear combination is defined using two learnable matrices $\boldsymbol{A}_o \in \mathbb{R}^{N_q \times C_o}$ and $\boldsymbol{A}_v \in \mathbb{R}^{N_q \times C_v}$ as follows:

$$\boldsymbol{q}_o = \boldsymbol{A}_o \boldsymbol{t}_o, \quad \boldsymbol{q}_v = \boldsymbol{A}_v \boldsymbol{t}_v \tag{6}$$

$$\boldsymbol{q}_{ov} = \boldsymbol{q}_o + \boldsymbol{q}_v \tag{7}$$

Specific Target Guided Denoising. In Fig. 8, we illustrate the process of DN (DeNoising) query initialization and adding noise to the ground-truth HOI instances. Given the ground-truth object label set $\boldsymbol{O}_{gt} = \{\boldsymbol{o}_i\}_{i=1}^{K}$ and verb label set $\boldsymbol{V}_{gt} = \{\boldsymbol{v}_i\}_{i=1}^{K}$, two types of label DN queries are initialized. Here, $\boldsymbol{o}_i$ and $\boldsymbol{v}_i$ are one-hot labels of object class and verb class, and k is the number of ground-truth HOI instances. For the k-th ground-truth HOI instance, the ground-truth index of the object label $\boldsymbol{o}_k$ is randomly flipped to the other object class indices to obtain the noisy object label $\boldsymbol{o}'_k$, and N_p groups of noisy labels are generated. Next, the object DN query $\boldsymbol{q}_{dn}^{(o)} \in \mathbb{R}^{N_p \cdot K \times C_d}$ is collected from the object label embedding $\boldsymbol{t}_o$ according to the indices of the noisy object label $\boldsymbol{O}'_{gt}$. Since the verb label has co-occurrence ground-truth classes, the other indices of the ground-truth verb label are randomly flipped to generate the noisy verb label $\boldsymbol{v}'_k$ so that the co-occurrence ground-truth index appears in the noisy verb label. Similar to the object DN query, the verb label DN query $\boldsymbol{q}_{dn}^{(v)} \in \mathbb{R}^{N_p \cdot K \times C_d}$ is the sum of the verb label DN embeddings selected from the verb label embedding $\boldsymbol{t}_v$ according to the indices of the noisy verb label $\boldsymbol{V}'_{gt}$. Finally, the object DN query and verb DN query are concatenated to form the DN query $\boldsymbol{q}_{dn} \in \mathbb{R}^{2N_p \cdot K \times C_d}$ for noise removal learning.

IV. Experiments

Dataset and Metric. We evaluate our proposed methods on the HICO-Det dataset [1]. The HICO-Det dataset contains 38,118 images for training and 9,658 images for the test with 117 verb and 80 object categories and 600 HOI categories. The mean Average Precision (mAP) is used as the evaluation metric. Specifically, for a true positive result, the intersection over union (IoU) between the predicted human and object bounding boxes and the ground-truth human and object bounding boxes need to be higher than 0.5, and the predicted object class and verb class need to be right. According to the evaluation protocol of HICO-Det [1], the mAP is calculated in two settings: the default setting and the known object setting, and three categories *Full* (all of 600 HOI classes), *Rare* (138 HOI classes with less than 10 instances), *Non-Rare* (462 HOI classes with 10 or more than 10 instances) for each setting.

Implementation Details. Our QAHOI and PQNet follow the setting of QPIC [17]. QAHOI is trained for 150 epochs which is the same as QPIC. We use Swin Transformer [37] as the backbone of QAHOI. Three variants of PQNet are implemented: PQNet-S with ResNet-50 backbone and 3-layer decoders, PQNet-B with ResNet-50 backbone and 5-layer decoders, and PQNet-L with ResNet-101 backbone and 5-layer decoders. For PQNet-S, we train the model for 70 epochs while PQNet-B and PQNet-L for 100 epochs. Our SOV-STG uses the same setting as QAHOI and is trained for 30 epochs. Similarly, for SOV-STG, we implement three variants: SOV-STG-S with ResNet-50 backbone and 3-layer decoders, SOV-STG-M with ResNet-101 backbone and 3-layer decoders, and SOV-STG-L with ResNet-101 backbone and 6-layer decoders. For QAHOI and SOV-STG, we also use Swin Transformer [37] as the backbone to achieve SOTA performance.

Comparison with SOTA. Table I shows the comparison of our proposed methods with the state-of-the-art methods on the HICO-Det dataset. Compared to the previous SOTA method, CDN [28], our PQNet-S achieves better performance with fewer training epochs. Compared to the recent SOTA method, GEN-VLKT [32], our SOV-STG achieves better performance with only one-third of the training epochs. For the best performance, our SOV-STG-Swin-L achieves 43.35 mAP on the *Full* category in the Default setting and is 4.5% higher than DiffHOI-Swin-L [42] which uses large-scale synthetic images.

V. Conclusion

In this paper, we focus on improving the model representation capability and introduce three novel methods for HOI detection: QAHOI [25], PQNet [26] and SOV-STG [27]. QAHOI improves the detection framework with the multi-scale architecture and query-based anchors. PQNet improves the query embedding and the fusion of verb embeddings. SOV-STG combines the advantages of QAHOI and PQNet and introduces the prior knowledge from the ground-truth with a denoising learning strategy. With the above advancements, our SOV-STG achieves SOTA performance with one-third of training epochs compared to the previous SOTA.

REFERENCES

[1] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, "Learning to detect human-object interactions," in *WACV*, 2018.

[2] G. Gkioxari, R. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *CVPR*, 2018.

[3] C. Gao, Y. Zou, and J.-B. Huang, "iCAN: Instance-centric attention network for human-object interaction detection," in *BMVC*, 2018.

[4] P. Zhou and M. Chi, "Relation parsing neural network for human-object interaction detection," in *ICCV*, 2019.

[5] O. Ulutan, A. Iftekhar, and B. S. Manjunath, "VSGNet: Spatial attention network for detecting human object interactions using graph convolutions," in *CVPR*, 2020.

[6] C. Gao, J. Xu, Y. Zou, and J.-B. Huang, "DRG: Dual relation graph for human-object interaction detection," in *ECCV*, 2020.

[7] F. Z. Zhang, D. Campbell, and S. Gould, "Spatially conditioned graphs for detecting human-object interactions," in *ICCV*, 2021.

[8] Y.-L. Li, S. Zhou, X. Huang, L. Xu, Z. Ma, H.-S. Fang, Y. Wang, and C. Lu, "Transferable interactiveness knowledge for human-object interaction detection," in *CVPR*, 2019.

[9] X. Zhong, C. Ding, X. Qu, and D. Tao, "Polysemy deciphering network for robust human-object interaction detection," in *IJCV*, 2021.

[10] T. Gupta, A. Schwing, and D. Hoiem, "No-frills human-object interaction detection: Factorization, layout encodings, and training techniques," in *ICCV*, 2019.

[11] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *IEEE TPAMI*, 2016.

[13] Y. Liao, S. Liu, F. Wang, Y. Chen, C. Qian, and J. Feng, "PPDM: Parallel point detection and matching for real-time human-object interaction detection," in *CVPR*, 2020.

[14] X. Zhong, X. Qu, C. Ding, and D. Tao, "Glance and Gaze: Inferring action-aware points for one-stage human-object interaction detection," in *CVPR*, 2021.

[15] T. Wang, T. Yang, M. Danelljan, F. S. Khan, X. Zhang, and J. Sun, "Learning human-object interaction detection using interaction points," in *CVPR*, 2020.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[17] M. Tamura, H. Ohashi, and T. Yoshinaga, "QPIC: Query-based pairwise human-object interaction detection with image-wide contextual information," in *CVPR*, 2021.

[18] C. Zou, B. Wang, Y. Hu, J. Liu, Q. Wu, Y. Zhao, B. Li, C. Zhang, C. Zhang, Y. Wei *et al.*, "End-to-end human object interaction detection with hoi transformer," in *CVPR*, 2021.

[19] M. Chen, Y. Liao, S. Liu, Z. Chen, F. Wang, and C. Qian, "Reformulating hoi detection as adaptive set prediction," in *CVPR*, 2021.

[20] B. Kim, J. Lee, J. Kang, E.-S. Kim, and H. J. Kim, "HOTR: End-to-end human-object interaction detection with transformers," in *CVPR*, 2021.

[21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[22] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *ICLR*, 2020.

[23] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "DAB-DETR: Dynamic anchor boxes are better queries for DETR," in *ICLR*, 2022.

[24] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate detr training by introducing query denoising," in *CVPR*, 2022.

[25] J. Chen and K. Yanai, "QAHOI: Query-based anchors for human-object interaction detection," in *18th International Conference on Machine Vision and Applications (MVA)*, 2023.

[26] ——, "Parallel queries for human-object interaction detection," in *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, 2022.

[27] J. Chen, Y. Wang, and K. Yanai, "Focusing on what to decode and what to train: Efficient training with hoi split decoders and specific target guided denoising," *arXiv preprint arXiv:2307.02291*, 2023.

[28] A. Zhang, Y. Liao, S. Liu, M. Lu, Y. Wang, C. Gao, and X. Li, "Mining the benefits of two-stage and one-stage hoi detection," in *NeurIPS*, 2021.

[29] A. Iftekhar, H. Chen, K. Kundu, X. Li, J. Tighe, and D. Modolo, "What to look at and where: Semantic and spatial refined transformer for detecting human-object interactions," in *CVPR*, 2022.

[30] H. Yuan, M. Wang, D. Ni, and L. Xu, "Detecting human-object interactions with object-guided cross-modal calibrated semantics," in *AAAI*, 2022.

[31] L. Dong, Z. Li, K. Xu, Z. Zhang, L. Yan, S. Zhong, and X. Zou, "Category-aware transformer network for better human-object interaction detection," in *CVPR*, 2022.

[32] Y. Liao, A. Zhang, M. Lu, Y. Wang, X. Li, and S. Liu, "GEN-VLKT: Simplify association and enhance interaction understanding for hoi detection," in *CVPR*, 2022.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[34] X. Qu, C. Ding, X. Li, X. Zhong, and D. Tao, "Distillation using oracle queries for transformer-based human-object interaction detection," in *CVPR*, 2022.

[35] X. Zhong, C. Ding, Z. Li, and S. Huang, "Towards hard-positive query mining for detr-based human-object interaction detection," in *ECCV*, 2022.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[37] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.

[38] H. Yuan, J. Jiang, S. Albanie, T. Feng, Z. Huang, D. Ni, and M. Tang, "RLIP: Relational language-image pre-training for human-object interaction detection," in *NeurIPS*, 2022.

[39] S. Kim, D. Jung, and M. Cho, "Relational context learning for human-object interaction detection," in *CVPR*, 2023.

[40] S. Ning, L. Qiu, Y. Liu, and X. He, "HOICLIP: Efficient knowledge transfer for hoi detection with vision-language models," in *CVPR*, 2023.

[41] S. Ma, Y. Wang, S. Wang, and Y. Wei, "FGAHOI: Fine-grained anchors for human-object interaction detection," *arXiv preprint arXiv:2301.04019*, 2023.

[42] J. Yang, B. Li, F. Yang, A. Zeng, L. Zhang, and R. Zhang, "Boosting human-object interaction detection with text-to-image diffusion model," *arXiv preprint arXiv:2305.12252*, 2023.