

Agglomerative Transformer for Human-Object Interaction Detection

Danyang Tu¹, Wei Sun¹, Guangtao Zhai^{1✉}, Wei Shen^{2✉}

¹Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University

²MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University

{danyangtu, sunguwei, zhaiguangtao, wei.shen}@sjtu.edu.cn

Abstract

We propose an **agglomerative Transformer (AGER)** that enables Transformer-based human-object interaction (HOI) detectors to flexibly exploit extra instance-level cues in a single-stage and end-to-end manner for the first time. AGER acquires instance tokens by dynamically clustering patch tokens and aligning cluster centers to instances with textual guidance, thus enjoying two benefits: 1) *Integrality*: each instance token is encouraged to contain all discriminative feature regions of an instance, which demonstrates a significant improvement in the extraction of different instance-level cues and subsequently leads to a new state-of-the-art performance of HOI detection with 36.75 mAP on HICO-Det. 2) *Efficiency*: the dynamical clustering mechanism allows AGER to generate instance tokens jointly with the feature learning of the Transformer encoder, eliminating the need of an additional object detector or instance decoder in prior methods, thus allowing the extraction of desirable extra cues for HOI detection in a single-stage and end-to-end pipeline. Concretely, AGER reduces GFLOPs by 8.5% and improves FPS by 36%, even compared to a vanilla DETR-like pipeline without extra cue extraction. The code will be available at <https://github.com/six6607/AGER.git>.

1. Introduction

Human-object interaction (HOI) detection aims at understanding human activities at a fine-grained level. It involves both the localization of interacted human-object pairs and the recognition of their interactions, where the latter poses the major challenges as a higher-level vision task [9].

Since interactions describe the relations between different instances (*i.e.*, humans and objects), instance-level cues (*e.g.*, human pose and gaze) are unanimously recognized as pivotal to discriminating subtle visual differences between similar relation patterns in interaction recognition. However, extracting these instance-level cues intuitively indicates a multi-stage pipeline, where an off-the-shelf object detector is essential to generate instance proposals firstly [11, 44, 23, 7, 50, 58]. Such a paradigm struggles in proposal generation, yielding less competitive



Figure 1: **Instance queries vs. instance tokens.** Instance queries typically attend to instance parts, while our instance tokens are encouraged to contain integral discriminative regions of instances. More examples are presented in supplementary materials.

performance in model efficiency. In this work, we seek to explore a *single-stage* Transformer-based HOI detector to flexibly and efficiently exploit extra instance-level cues, thus continuing their success in HOI detection.

The challenge stems from task-bias, *i.e.*, different tasks have different preferences of discriminative feature regions [61]. Gaze tracking, for example, prefers the discriminative regions of human heads [40], whereas pose estimation favours holistic human body contexts [22]. Therefore, the crux of building a single-stage pipeline lies in a proper design of information carrier, which need to ensure the integrality of instance-level representations (IRs), *i.e.*, containing all discriminative regions of an instance to satisfy the diverse region preferences of different tasks. However, most popular Transformer-based detectors deal with local patches, neglecting the integrality of different instances.

Some previous methods partially tackled the above challenge. STIP [58] employs an additional DETR detector to generate instance proposals, which yet suffers from the low efficiency of the multistage pipeline. Several works [27, 3, 18] propose to use an additional query-based instance decoder to extract instance queries individually. Despite being ingenious, these queries are task-driven and

learned to highlight only the most distinguishable feature regions preferred by a given task, as verified by the sparsity of learned attention map [65]. As shown in Fig. 1, the object detection driven human queries in existing methods typically contain only instance extremities, which likewise fails to guarantee the integrality of IRs, limiting its adaptability to other tasks (*e.g.*, pose estimation) due to task bias (Sec. 4.2). Although joint multitask learning can partially alleviate the sparsity of instance queries, it introduces unexpected ambiguities and makes the model fitting harder [52].

In this paper, we present **AGER**, short for **AGglomerative Transform**ER****, a new framework that improves prior methods by proposing instance tokens, handling the above-mentioned challenges favorably. Specifically, we formulate tokenization as a text-guided dynamic clustering process, which progressively agglomerates semantic-related patch tokens (*i.e.*, belonging to the same instance) to enable the emergence of instance tokens through feature learning. Being decoupled from downstream tasks, the clustering mechanism encourages instance tokens to ensure the integrality of extracted IRs (Fig. 1) and eliminate task bias, thus allowing a flexible extraction of different instance-level cues for HOI detection. Despite being conceptually simple, instance tokens have some striking impacts. Unlike instance proposals being regular rectangles, the instance tokens are generated over irregularly shaped regions that are aligned to different instances with arbitrary shapes (Fig. 1), thus being more expressive. With this, **AGER** already outperforms **QPIC** [38] by **10.6%** mAP even without involving any extra cues (Sec. 4.3). Additionally, compared to instance queries, instance tokens demonstrate a significant precision improvement in cue extraction (Fig. 3), leading to a new state-of-the-art performance of HOI detection on **HICO-Det** [2] with **36.75** mAP. Of particular interest, the dynamical clustering mechanism can be seamlessly integrated with Transformer encoder, dispensing with additional object detectors or instance decoders and showing an impressive efficiency. Concretely, taking as input an image with size of 640×640 , **AGER** reduces GFLOPs by **8.5%** and improves FPS by **36.0%** even compared to **QPIC** that built on an vanilla DETR-like Transformer pipeline (Sec. 4.3), and the relative efficiency gaps become more evident as the image resolution grows (Fig 3c).

2. Related Work

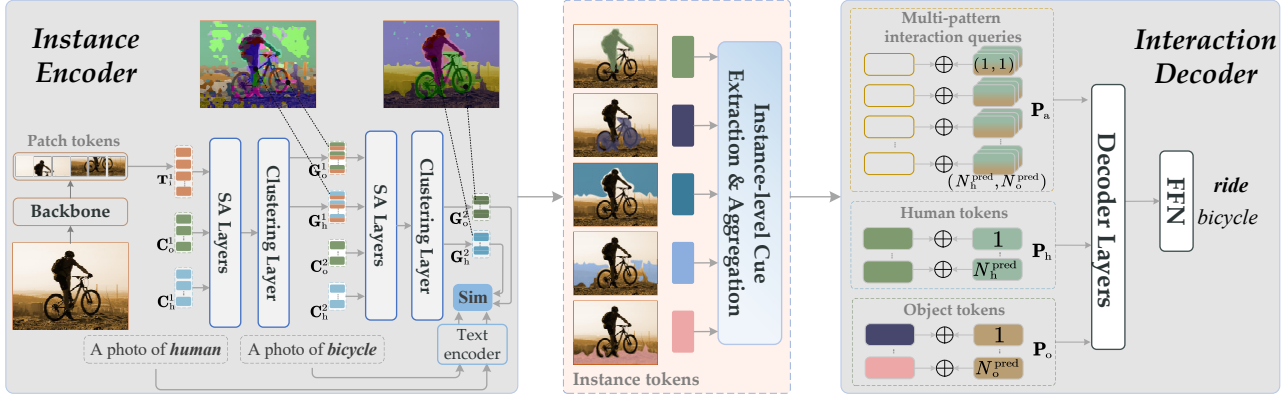
Modern HOI detection methods are built on three different information carriers of IRs, *i.e.*, instance proposals, points and instance queries, which show different effects on the utilization of instance-level cues.

Instance proposals dominated CNN-based HOI detection approaches for almost the entire era [2, 7, 8, 9, 11, 13, 17,

20, 23, 25, 29, 31, 35, 43, 44, 45, 47, 50, 53, 59, 63, 64]. These methods conventionally shared a two-stage pipeline, employing an object detector [37, 12] to generate instance proposals in the first stage. Then, the human and object proposals are processed separately to extract various instance-level cues, such as human pose [11, 44, 23], human parsing [31], spatial configurations [7], human gaze [50], object labels [59], among others. Along with visual features of appearance, these auxiliary cues were leveraged either individually or conjunctively to further reason about the interactions. Although a fine-selected proposal contains integral IRs, thus allowing the extraction of various fine-grained cues, the additional object detector inevitably compromises the efficiency of these methods. Furthermore, the cropped proposals lack global contextual information, leading to lower effectiveness. In contrast, the generation of the instance tokens in **AGER** does not involve an object detector but is optimized as a dynamically clustering process in an end-to-end manner along with the Transformer encoder. Moreover, the clustering mechanism enables instance tokens to be aggregated from a global perception field and potentially eliminates visual redundancy among similar patch tokens, leading to stronger expressiveness of instance tokens.

Points were proposed to represent instances to achieve a one-stage framework for HOI detection. Specifically, [26, 48, 60] represented the interactions as the midpoints of human-object pairs and detected them based on keypoint detection networks [34, 54], dispensing with additional detectors. Thus, they enjoyed a simpler and faster pipeline, but at the expense of the capability to freely extract extra cues due to the lack of integral IRs.

instance queries were first introduced in the Transformer-based detector [1], which interact with patch tokens and aggregate information through several interleaved cross-attention and self-attention modules. Thanks to the impressive global context modeling capability, Transformer rapidly revolutionizes HOI detection methods [66, 4, 18, 3, 38, 56, 19, 14, 62, 58, 41, 30, 27, 42]. Most works [66, 38, 3] focused on designing an end-to-end pipeline and continuing the success of the attention mechanism for HOI detection, dealing with visual appearance features solely and neglecting the potential of extra instance-level cues. Additionally, some methods [18, 27] propose to use additional queries to detect instances individually by stacking more decoders. Nevertheless, instance queries are task-driven and fail to extract integral IRs, weakening their ability to extract extra cues due to task bias. In comparison, our **AGER** introduces clustering mechanism into Transformer to enable the generation of instance tokens that guarantee the integrality of IRs, which continues the success of global attention and meanwhile enjoys the potential of extra instance-level cues.



T: Token **C**: Clustering Center **G**: Grouped Token **Sim**: Similarity **P**: Embedding \square : zero **i**: image **h**: human **o**: object **a**: interaction

Figure 2: **Architecture of AGER.** AGER performs tokenization as a text-guided dynamic clustering process in the instance encoder, dispensing with any additional object detector or instance decoder, which outputs instance tokens that encourage the integrality of instance-level representations. This property enables the extraction of different instance-level cues in a *single-stage* pipeline. Finally, a new interaction decoder leverages these desirable cues to recognize interactions in a multi-pattern manner.

3. Method

In this section, we aim to explore the solution for a *single-stage* pipeline that allows us to leverage extra instance-level cues for HOI detection. We start with a detailed description of our instance encoder, which incorporates the attention mechanism with dynamical clustering to extract instance tokens, in Sec. 3.1. Then, we take three instance-level cues as guidance to explain the scheme of the extraction and aggregation of extra cues in Sec. 3.2. Next, in Sec 3.3, we propose a new interaction decoder that enumerates all human-object pairs to recognize their interactions in a multi-pattern manner. Finally, we design a special loss function that enables the textual guidance in Sec. 3.4.

3.1. Instance Encoder

As shown in Fig. 2, the instance encoder is organized as a backbone followed by a hierarchical Transformer encoder, where the latter incorporates self-attention and clustering mechanism to extract instance tokens iteratively.

Backbone. An input image is first downsampled through a plain CNN backbone and then flattened to add a cosine positional embedding to harvest the initialized and sequenced patch tokens $\mathbf{T}_b \in \mathbb{R}^{N_b \times D_b}$.

Overall architecture. Transformer encoder consists of two stages that share an identical architecture, which comprises of several self-attention layers and a clustering layer.

Concretely, in the s -th stage, we first initialize two sets of learnable clustering centers for humans $\mathbf{C}_h^s \in \mathbb{R}^{N_h^s \times D_h^s}$ and objects $\mathbf{C}_o^s \in \mathbb{R}^{N_o^s \times D_o^s}$ separately, which are then concatenated with image tokens \mathbf{T}_i^s and learned to update representations through several self-attention (SA) layers. Subsequently, at the end of each stage, we assign each image

token to different clustering centers based on feature affinities, and the assigned image tokens are then aggregated in the clustering layer. Formally, each stage is computed as

$$[\hat{\mathbf{C}}_h^s; \hat{\mathbf{C}}_o^s; \hat{\mathbf{T}}_i^s] = \text{SA-Layer}([\mathbf{C}_h^s; \mathbf{C}_o^s; \mathbf{T}_i^s]), \quad (1)$$

$$[\mathbf{G}_h^s; \mathbf{G}_o^s] = \text{ClusteringLayer}([\hat{\mathbf{C}}_h^s; \hat{\mathbf{C}}_o^s; \hat{\mathbf{T}}_i^s). \quad (2)$$

Here, $[\ ; \]$ denotes concatenation operator, $\mathbf{G}_h^s \in \mathbb{R}^{N_h^s \times D_h^s}$ and $\mathbf{G}_o^s \in \mathbb{R}^{N_o^s \times D_o^s}$ are the agglomerated image tokens after the s -th stage. Note that we omit the modules of token projection, residual connection and normalization here. Specifically, $\mathbf{T}_i^1 = \mathbf{T}_b$ and $\mathbf{T}_i^2 = [\mathbf{G}_h^1; \mathbf{G}_o^1]$, *i.e.*, we feed the initialized patch tokens from the backbone to the 1-th stage, and these small local patches are dynamically agglomerated into relatively larger segments, which are subsequently fed into the 2-th stage to generate the final instance tokens. Following [51], we propagate the learned clustering centers in the 1st stage to the 2nd stage through a MLP-Mixer layer [39]. Meanwhile, to make the human and object clustering centers distinct, we add two sets of position embedding to them. Then, for the human centers, they are obtained via

$$\mathbf{P}_h^s = \text{Embedding}(N_h^s, D_h^s), \quad (3)$$

$$\tilde{\mathbf{C}}_h^s = \text{Zeros}(N_h^s, D_h^s), \quad (4)$$

$$\mathbf{C}_h^1 = \tilde{\mathbf{C}}_h^1 + \mathbf{P}_h^1, \quad (5)$$

$$\mathbf{C}_h^2 = \tilde{\mathbf{C}}_h^2 + \mathbf{P}_h^2 + \text{MLP-Mixer}(\tilde{\mathbf{C}}_h^1). \quad (6)$$

$\tilde{\mathbf{C}}_h^s$ indicate the centers that are initialized as zeros and $\tilde{\mathbf{C}}_h^1$ are updated center representations that are calculated by Eq. 7. Object centers share the same process.

Clustering layer. The clustering layer at the end of each stage aims to aggregate local image tokens into a new token

based on their feature affinity, thus the small local patch tokens can be gradually merged into a larger segment and finally into an instance token that covers the integral discriminative feature region of an instance.

In particular, we first employ a cross-attention module to update the representation of clustering centers, which enables information propagation between clustering centers and image tokens via

$$\hat{\mathbf{C}}_{[h,o]}^s = \text{softmax}\left(\frac{\hat{\mathbf{C}}_{[h,o]}^s \cdot (\hat{\mathbf{T}}_i^s)^\top}{\sqrt{D_i^s}}\right) \cdot \hat{\mathbf{T}}_i^s, \quad (7)$$

where $\hat{\mathbf{C}}_{[h,o]}^s = [\hat{\mathbf{C}}_h^s; \hat{\mathbf{C}}_o^s]$ is the concatenation of human and object centers from Eq. 1. D_i^s is the channel dimension of image tokens. Subsequently, we adopt the scheme in [51] to employ a Gumbel-softmax [15] to compute the similarity matrix \mathbf{A}^s between the clustering centers and the image tokens as

$$\mathbf{A}_{(k,j)}^s = \frac{\exp(W_c \check{c}_k^s \cdot W_i \hat{t}_j^s + \gamma_i)}{\sum_{n=1}^{N_c^s} \exp(W_c \check{c}_n^s \cdot W_i \hat{t}_j^s + \gamma_n)}, \quad (8)$$

where \check{c}_k^s stands for the k -th clustering center in $\hat{\mathbf{C}}_{[h,o]}^s$ and \hat{t}_j^s denotes the j -th updated image token in $\hat{\mathbf{T}}_i^s$. $N_c^s = N_h^s + N_o^s$ counts the total number of clustering centers in the s -th stage. W_c and W_i are the weights of the learned linear projections for the clustering centers and the image tokens, respectively. $\{\gamma\}_{n=1}^{N_c^s}$ are *i.i.d* random samples drawn from the Gumbel(0, 1) distribution that enables the Gumbel-softmax distribution to be close to the real categorical distribution. Finally, we merge N_i^s image tokens with corresponding clustering centers to calculate grouped tokens $\mathbf{G}_{[h,o]}^s = [\mathbf{G}_h^s; \mathbf{G}_o^s]$ via

$$\mathbf{g}_k^s = \check{c}_k^s + W_u \frac{\sum_{j=1}^{N_i^s} \mathbf{A}_{(k,j)}^s W_v \hat{t}_j^s}{\sum_{j=1}^{N_i^s} \mathbf{A}_{(k,j)}^s}, \quad (9)$$

where \mathbf{g}_k^s is the k -th grouped token in $\mathbf{G}_{[h,o]}^s$, W_u and W_v are learned weights to project the merged features.

3.2. Cues Extraction & Aggregation

This work realizes three instance-level cues, *i.e.*, human poses (P), spatial locations (S) and object categories (T), as guidance, other valuable cues can be extracted similarly.

Extraction. Unlike prior methods that use different specially customized models to extract different cues, we extract those cues through several lightweight MLPs in parallel, thanks to the excellent expressiveness of the instance tokens. Concretely, we perform a 5-layer MLP to estimate the normalized locations of 17 keypoints for human pose estimation. Note that object tokens do not have a pose representation. Meanwhile, a 3-layer MLP is used to predict

the normalized bounding boxes of all humans and objects as their spatial locations. Additionally, we adopt a 1-layer FFN to predict each category of humans and objects \hat{y} . Specifically, for the i -th human instance, its prediction $\hat{y}_h^i \in [0, 1]^2$, where the 2-th element indicates *no-human*. For object instance, similarly, $\hat{y}_o^i \in [0, 1]^{N_o^c+1}$, where N_o^c is the number of object classes and the $(N_o^c + 1)$ -th element denotes *no-object*.

Aggregation. We first adopt two fully connected layers to project all cues into a united and embedded feature space, leading to four new cue representations $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{N_h^2 \times D_{\text{pos}}}$ (human poses), $\mathbf{E}_{\text{h-spa}} \in \mathbb{R}^{N_h^2 \times D_{\text{spa}}}$ (human spatial locations), $\mathbf{E}_{\text{o-spa}} \in \mathbb{R}^{N_o^2 \times D_{\text{spa}}}$ (object spatial locations) and $\mathbf{E}_{\text{cls}} \in \mathbb{R}^{N_o^2 \times D_{\text{cls}}}$ (object classes). Particularly, the text of the predicted object name is first transformed into a vector using Word2Vec [33]. Since these cues may introduce noise due to misrecognition, we manually set a threshold $\gamma = 0.7$ over the confidence of category prediction to decide their employment. Concretely, if the category (*no-object* and *no-human* are excluded) prediction confidence of an instance is larger than γ , we keep its corresponding cues otherwise reset them as 0. Finally, these cues are concatenated to corresponding instance tokens to obtain the final representations via:

$$\hat{\mathbf{T}}_h = W_h[\mathbf{T}_h; \mathbf{E}_{\text{pos}}; \mathbf{E}_{\text{h-spa}}], \quad (10)$$

$$\hat{\mathbf{T}}_o = W_o[\mathbf{T}_o; \mathbf{E}_{\text{cls}}; \mathbf{E}_{\text{o-spa}}], \quad (11)$$

where $\mathbf{T}_h = \mathbf{G}_h^2$ and $\mathbf{T}_o = \mathbf{G}_o^2$ are human and object tokens generated by the instance encoder. W_h and W_o are learned weights to project the concatenated features.

3.3. Interaction Decoder

We adopt a 3-layer Transformer decoder to recognize interactions, of which each layer consists of a cross-attention module and a self-attention module. As the clustering mechanism in the instance encoder has located different humans and objects, our decoder aims to associate the interacted human-object pairs and recognize their interactions.

Association. Formally, a given image is invariably transformed into $N_h^{\text{pred}} = N_h^2$ human tokens $\hat{\mathbf{T}}_h \in \mathbb{R}^{N_h^{\text{pred}} \times D_h}$ and $N_o^{\text{pred}} = N_o^2$ object tokens $\hat{\mathbf{T}}_o \in \mathbb{R}^{N_o^{\text{pred}} \times D_o}$ after the instance encoder and the cue utilization module. By design, $D_h = D_o$ and we simplify them as D . Then, we add two sets of position embedding to $\hat{\mathbf{T}}_h$ and $\hat{\mathbf{T}}_o$ respectively via:

$$\mathbf{P}_h = \text{Embedding}(N_h^{\text{pred}}, D), \quad \check{\mathbf{T}}_h = \hat{\mathbf{T}}_h + \mathbf{P}_h; \quad (12)$$

$$\mathbf{P}_o = \text{Embedding}(N_o^{\text{pred}}, D), \quad \check{\mathbf{T}}_o = \hat{\mathbf{T}}_o + \mathbf{P}_o. \quad (13)$$

Next, the position embedding for interaction queries is initialized as the one-to-one sum of the human and object position embedding. Concretely, the position of the (ij) -th interaction is the sum of the position of the i -th hu-

man and the position of the j -th object, *i.e.*, $\mathbf{p}_a^{(ij)} = \mathbf{p}_h^i + \mathbf{p}_o^j$, leading to an interaction position embedding $\mathbf{P}_a \in \mathbb{R}^{N_h^{\text{pred}} N_o^{\text{pred}} \times D}$, which actually enumerates total $N_h^{\text{pred}} N_o^{\text{pred}}$ possible human-object pairs.

Moreover, in practical scenarios, one human-object pair may have multiple interaction labels. Thus, we follow [49] to incorporate multiple patterns into each interaction position. Concretely, we use a small set pattern embedding $\mathbf{P}_{\text{pattern}} = \text{Embedding}(N_{\text{pattern}}, D)$ to detect different interactions from each human-object pair. N_{pattern} is the number of patterns that is very small, here $N_{\text{pattern}} = 3$. Next, we share the $\mathbf{P}_{\text{pattern}}$ to each interaction position \mathbf{p}_a to get the multi-pattern interaction position embedding $\hat{\mathbf{P}}_a \in \mathbb{R}^{N_a \times D}$, where $N_a = N_{\text{pattern}} \times N_h^{\text{pred}} \times N_o^{\text{pred}}$. Finally, our interaction queries are initialized as:

$$\mathbf{Q}_a = \text{Zeros}(N_a, D) + \hat{\mathbf{P}}_a. \quad (14)$$

Recognition. Along with the human and object instance tokens from the instance encoder, we feed the interaction queries \mathbf{Q}_a into the interaction decoder. After that, the interactions are recognized through a 1-layer FFN, following QPIC [38].

3.4. Loss Function

The loss function consists of three parts: 1) loss of interaction recognition \mathcal{L}_a , 2) loss of cues extraction \mathcal{L}_e , and 3) loss of instance token generation \mathcal{L}_t . Specifically, \mathcal{L}_e consists of pose estimation and location regression. Category recognition is jointly optimized with \mathcal{L}_t . We use the focal loss [28] as \mathcal{L}_a and adopt L_2 loss as \mathcal{L}_e . The total loss is the weighted sum of them, *i.e.*, $\mathcal{L} = \alpha_1 \mathcal{L}_a + \alpha_2 \mathcal{L}_e + \alpha_3 \mathcal{L}_t$. More details are described in *supplementary materials*. Here, we mainly introduce the design of \mathcal{L}_t , which enables text representations to guide the generation of instance tokens.

3.4.1 Textual Guidance

Actually, some works have tried to incorporate clustering with Transformer for other tasks, such as GroupViT [51] and kMaX [55], and we borrow some ideas from them for model design. However, training the model for HOI detection is not easy. GroupViT use contrastive loss, which demands large training batch size (4096) and kMaX uses heavy decoder and dense annotations. All of these are unaffordable for HOI detection. Thus, we devise a new loss function that uses a textual signal to guide the learning of the instance encoder by enforcing a similarity between the textual representation and the instance token representation. To this end, we first define a similarity metric and then match instance tokens to each ground truth instance with this metric and finally optimize the instance encoder.

Similarity metric. Suppose an input image contains N_h^{gt}

humans and N_o^{gt} objects, in which the j -th object is labeled as \mathbf{y}_o^j . Then, taking objects as examples, our similarity metric $\text{sim}(\cdot, \cdot)$ between the j -th ground truth object and the k -th generated object token \mathbf{t}_o^k is defined as

$$\text{sim}(j, k) = \hat{\mathbf{y}}_o^k(j) \times \text{Cosine}(\mathbf{r}_{\text{vis}}^k, \mathbf{r}_{\text{txt}}^j), \quad (15)$$

where $\hat{\mathbf{y}}_o^k(j) \in [0, 1]$ is the probability of predicting the j -th class and $\text{Cosine}(\cdot, \cdot)$ denotes cosine similarity. $\mathbf{r}_{\text{vis}}^k$ is visual representation vector projected from the k -th object token \mathbf{t}_o^k through two FC layers, and $\mathbf{r}_{\text{txt}}^j$ is a text representation vector from CLIP [36]. Concretely, we prompt the *noun* word of j -th ground truth class with a handcrafted sentence template, *i.e.*, “A photo of a {noun}”. Then, we feed this sentence into a frozen text encoder of CLIP followed by two FC layers as projector to acquire the text representation $\mathbf{r}_{\text{txt}}^j$. The human tokens share the same process. Note that for human, the j ranges from 1 to 2, indicating *human* and *no-human*, while for object, $j = [1, 2, \dots, N_o^c, N_o^c + 1]$, denoting total N_o^c different object categories and a *no-object*.

Instance matching. By design, $N_h^{\text{pred}} > N_h^{\text{gt}}$ and $N_o^{\text{pred}} > N_o^{\text{gt}}$. We first pad $(N_h^{\text{pred}} - N_h^{\text{gt}})$ and $(N_o^{\text{pred}} - N_o^{\text{gt}})$ “*nothing*”s to human and object ground truths respectively, leading to two new ground truth sets $\{\mathbf{y}_h^i\}_{i=1}^{N_h^{\text{pred}}}$ and $\{\mathbf{y}_o^j\}_{j=1}^{N_o^{\text{pred}}}$. Following, in case of the object tokens (same for the human tokens), we search for a permutation of N_o^{pred} elements $\hat{\sigma} \in \mathfrak{S}_{N_o^{\text{pred}}}$ to achieve the maximum total similarity:

$$\hat{\sigma} = \arg \max_{\sigma \in \mathfrak{S}_{N_o^{\text{pred}}}} \sum_{i=1}^{N_o^{\text{pred}}} \text{sim}(i, \sigma(i)). \quad (16)$$

The optimal assignment is calculated with the Hungarian algorithm [21], following DETR [1].

Objective. After finding the optimal assignment $\hat{\sigma}$, we are inspired by [46] to define the objective taking into account both positive predictions (assigned to ground truths that are not *nothing*) and negative (assigned to *nothing*) predictions into account. In case of object instances, the positive loss is calculated as:

$$\begin{aligned} \mathcal{L}_o^{\text{pos}} &= \sum_{i=1}^{N_o^{\text{gt}}} \text{sg}(\hat{\mathbf{y}}_o^{\hat{\sigma}(i)}(i)) \cdot [-\text{Cosine}(\mathbf{r}_{\text{vis}}^{\hat{\sigma}(i)}, \mathbf{r}_{\text{txt}}^i)] \\ &+ \sum_{i=1}^{N_o^{\text{gt}}} \text{sg}(\text{Cosine}(\mathbf{r}_{\text{vis}}^{\hat{\sigma}(i)}, \mathbf{r}_{\text{txt}}^i)) \cdot [-\log \hat{\mathbf{y}}_o^{\hat{\sigma}(i)}(i)]. \end{aligned} \quad (17)$$

Intuitively, $\mathcal{L}_o^{\text{pos}}$ is equivalent to optimizing a cosine loss weighted by the class correctness and optimizing a cross-entropy loss weighted by the cosine similarity. Note that the stop gradient operator $\text{sg}(\cdot)$ ensures constant loss weights. If a token is mis-recognized, we disregard its representation since it is a false negative anyway. The wrong representation also downweights the weight of the recognition loss.

Thus, we enforce the representation and class to be correct at the same time. Besides, we define the negative loss as:

$$\mathcal{L}_o^{\text{neg}} = \sum_{j=N_o^{\text{gt}}+1}^{N_o^{\text{pred}}} [-\log \hat{\mathbf{y}}_o^{\hat{\sigma}(j)}(N_o^c + 1)]. \quad (18)$$

Finally, the objective for the object instances is designed as $\mathcal{L}_t^o = \lambda \mathcal{L}_o^{\text{pos}} + (1-\lambda) \mathcal{L}_o^{\text{neg}}$. The objective of human instances \mathcal{L}_t^h shares the same process. Finally, $\mathcal{L}_t = \mathcal{L}_t^o + \mathcal{L}_t^h$.

4. Experiments

Technical details. Most of our default settings follow QPIC [38], *e.g.*, data augmentation, backbone, etc. Specifically, the channel dimension of all tokens, clustering centers and position embedding are set to 256. $D_{\text{pos}} = 64$, $D_{\text{sqa}} = 16$, $D_{\text{cls}} = 64$. We design $N_h^1 = 16$; $N_h^2 = 4$ and $N_o^1 = 64$; $N_o^2 = 8$. There are 4 and 2 self-attention layers in the first and second stage. For loss calculation, $\alpha_{1,2,3}$ are set to 2.5, 1, 1.5, and $\lambda = 0.75$. For human pose, we use the annotations provided by [6, 24] for HICO-Det [2] and the annotations from [16] for V-COCO [10].

Training. Our batch size is 32, with an initialized learning rate of the backbone 10^{-5} , that of the others $2.5e^{-4}$, and the weight decay 10^{-4} . We adopt the AdamW [32] optimizer for a total of 150 epochs where learning rates are decayed after 80 and 120 epochs.

4.1. Importance of Instance-level Cues

This subsection aims to verify the importance of different instance-level cues and explore why they facilitate interaction recognition. As Tab. 1a verified, all cues contribute a performance gain for HOI detection, especially for the “rare” case (with fewer than 10 training instances), ranging from 3.4% to 10.1%. The transformer shows excellent performance when dealing with a large number of training samples yet an inferior performance with inadequate sample volume due to the lack of inductive bias [5]. However, HOI detection has always been plagued by the long-tail distribution problem, interactions (*e.g.*, *stand on chair*) with a minority of samples are thereby more likely to be misrecognized as an interaction (*e.g.*, *sit on chair*) with similar visual pattern but massive samples. In this case, instance-level cues serve as some explicit priori knowledge that may be prioritised by the Transformer to recognize interactions. We further verify this solution in Tab. 1b. Concretely, we choose 5,000 images of *wheel* bicycle and 5,000 images of *ride* bicycle to retrain the interaction decoder with the instance encoder being frozen. As the table shows, when the sample volumes of two interaction instances differ substantially (*e.g.*, 500 vs. 5,000), additional cues can significantly improve performance, especially for small samples (79.1% vs. 13.7%). However, the gain diminishes as the sample

size tends to equalize (5.3% vs. 6.4% with 5,000/5,000 samples). Additionally, Tab. 1c reports the mean difference between the cues extracted from these two interaction examples. Empirically, a relatively larger mean difference indicates a better recognizability and thus facilitates the process of classification. From this point, the various instance-level cues are more desirable features for interaction recognition.

4.2. Importance of Clustering

As mentioned previously, the integrality of IRs is the cornerstone of extracting different cues in a single-stage framework. Fig. 3a first shows the coverage rate of different instance information carriers over the instance bounding box. Concretely, the proposals extracted by an extra object detector (DETR in here) show best performance, but enforces a two-stage pipeline that compromises the efficiency. Meanwhile, object detection-driven instance queries in GEN [27] attend to instance parts (14.85% coverage rate), which leads to an inferior performance in extracting other cues, as shown in Fig. 3b. In comparison, the instance tokens generated by clustering enable a sufficient coverage over instances, allowing one to flexibly extract different extra cues (3× precision improvement). Interestingly, the clustering mechanism natively eliminates the visual redundancy in similar tokens, promising the instance tokens a capability for increased expressiveness. Therefore, even without using any additional decoder, AGER already shows a competitive result of object detection (57.48@AP50) compared to other more complex methods.

4.3. Analysis of Effectiveness & Efficiency

Effectiveness. Tab. 2 and Tab. 3 verify the effectiveness of AGER on HICO-Det [2] and V-COCO [10], respectively. First, AGER even without involving any additional cues already achieves a competitive result, with a relative 10.6% mAP gain compared to QPIC [38] on HICO-Det. It is ascribable to the CLIP-guided dynamic clustering process, which reduces the visual redundancy in patch tokens and leads to more expressive instance tokens. Secondly, AGER achieves a new state-of-the-art performance (36.75 mAP) based on human poses, spatial distributions and object categories. Note that this result can be further improved by using more valuable cues (37.10/37.77 mAP with gaze/interactiveness) at a negligible cost of additional parameters (+2.36M). However, we are not striving for that, but aim to provide the first paradigm that enables us to use extra cues in a single-stage manner, giving some valuable points to the HOI detection community. Although AGER does not achieve the optimal results on V-COCO, its performance is still very competitive.

Efficiency. In Tab. 4, we compare four different yet typical Transformer-based methods, including: (i) QPIC [38] that

Cues	Full	Rare	Number	w/o	w/	$\Delta \uparrow$	Feature	Diff.
A	32.15	23.81	5000/500	61.42/15.37	69.83/27.52	8.41/ 12.15	A	0.06
A+P	33.79 (+5.1%)	25.63 (+7.6%)	5000/1000	61.38/20.21	69.86/28.64	8.48 /8.43	P	0.13
A+S	32.74 (+2.0%)	24.61 (+3.4%)	5000/3000	58.63/39.07	65.74/43.86	7.11/4.79	S	0.08
A+T	34.08 (+6.0%)	26.21 (+10.1%)	5000/5000	57.45/52.51	61.14/55.28	3.69/2.77	T	-

(a) Extra cues improve HOI detection (b) mAP gain varies with different sample volumes (c) Cues differ in recognizability

Table 1: **Importance of instance-level cues.** (a) The results of incorporating visual appearance features (A) with other cues in Sec. 3.2. (b) The results of using (w/) and not using (w/o) extra cues with different sample volumes. (c) The mean differences of different cues.

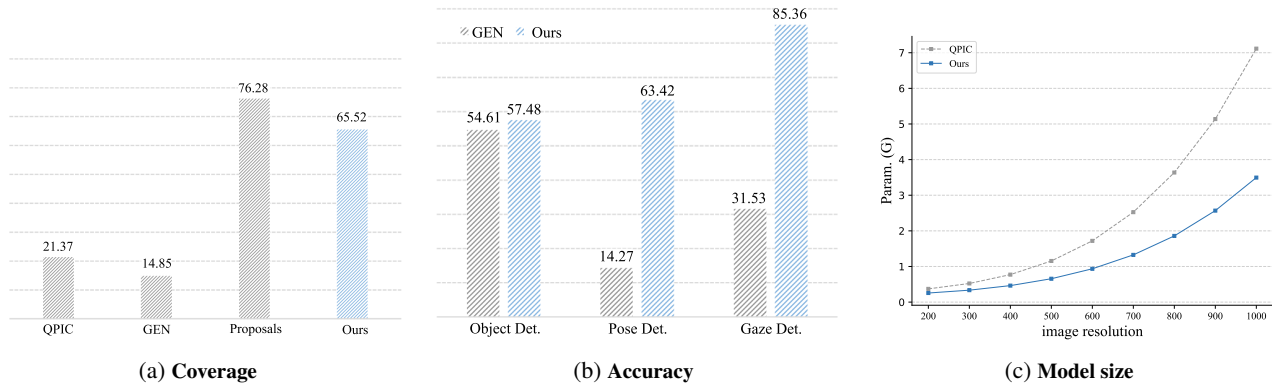


Figure 3: **Importance of clustering.** (a) We report the coverage as the proportion of the area of the feature region highlighted by an information carriers to the area of an instance. (b) Performance of different information carriers for different tasks (both fine-tuned with the same supervisory signals as ours). (c) The model parameters over image resolution.

adopt a vanilla DETR-like Transformer (6-layer encoder and 6-layer decoder); (ii) AS-Net [3] that performs two decoders to detect instances and interactions respectively (6-layer encoder and 2×6 -layer decoder); (iii) STIP [58] that built on a two-stage pipeline where instances are first detected through DETR [1] and (iv) our AGER. As shown in the table, AGER is even more efficient than QPIC that has the most simple architecture in prior Transformer-based HOI detection methods, with a relative 36.0% gain of FPS and a 8.5% reduction of FLOPs. Formally, additional computational costs of AGER are mainly introduced by calculating instance-level cues and clustering centers. However, for the former, thanks to the expressiveness of instance tokens, several lightweight MLPs are adequate to extract different cues, which bring a negligible additional complexity compared to the method using different customized tools. For the latter, although the first stage of the instance encoder takes more computation to update the clustering centers, the second stage starts to process much less tokens after clustering, and the number of tokens is further reduced after the second stage. Thus, the decoder demands a minority of computational complexity. Meanwhile, thanks to the great representation ability of the instance tokens, the decoder of AGER is much shallower than that of QPIC (3 vs. 6). Also, unlike QPIC has a quadratic computational cost *w.r.t* the number of pixels, the size of input image does not intro-

duce additional computations to AGER but the first stage of encoder. This is because except the first stage, AGER deals with a fixed number of tokens regardless of the input size. We visualize the relations between the complexity of different methods and the image resolution in Fig. 3c, and present a detailed validation in *supplementary materials*.

4.4. Ablation Study

Clustering center numbers. In Tab. 5a, we compare different numbers of clustering centers. Overall, increasing centers consistently improves performance, and we find (16,64) for the first stage and (4,8) for the second stage to be optimal. Empirically, an inadequate amount of centers may fail to characterize an image sufficiently, while an excessive amount of centers are likely to introduce unexpected noises.

Pattern numbers. Tab. 5b shows the effect of multi-pattern mechanism in the interaction decoder. Specially, when the number of patterns is one, we adopt the strategy of QPIC [38] to predict a *not-one-hot-like* label, *i.e.*, a label with multiple true values. However, such an intuitive solution brings more ambiguity. In contrast, our multi-pattern strategy explicitly encourages each position embedding to attend to one specific interaction, leading to a relative 5.6% mAP gain.

Strategies. We verify the effectiveness of the proposed strategies in Tab. 5c. Concretely, without explicitly adding

Method	Detector	Backbone	Cues	Single	Full	Default		Know Object		
						Rare	Non-Rare	Full	Rare	Non-Rare
CNN-based Methods:										
InteractNet [9]	COCO	R50-FPN	✗	✗	9.94	7.16	10.77	-	-	-
iCAN [8]	COCO	R50	✗	✗	14.84	10.45	16.15	16.26	11.33	17.73
PMFNet [44]	COCO	R50-FPN	✓	✗	17.46	15.65	18.00	20.34	17.47	21.20
DRG [7]	COCO	R50-FPN	✓	✗	19.26	17.74	19.71	23.40	21.75	23.89
FCMNet [31]	COCO	R50	✓	✗	20.41	17.34	21.56	22.04	18.97	23.12
DJ-RN [23]	COCO	R50	✓	✗	21.34	18.53	22.18	23.69	20.64	24.60
SCG [57]	COCO	R50-FPN	✓	✗	21.85	18.11	22.97	-	-	-
UnionDet [17]	COCO	R50	✗	✓	17.58	11.72	19.33	19.76	14.68	21.27
IP-Net [48]	COCO	Hg-104	✗	✓	19.56	12.79	21.58	22.05	15.77	23.92
PPDM [26]	HICO-Det	Hg-104	✗	✓	21.94	13.97	24.32	24.81	17.09	27.12
GG-Net [60]	HICO-Det	Hg-104	✗	✓	23.47	16.48	25.60	27.36	20.23	29.48
Transformer-based Methods:										
HOI-T [66]	HICO-Det	R50	✗	✓	23.46	16.91	25.41	26.15	19.24	28.22
PST [4]	-	R50	✗	✓	23.93	14.98	26.60	26.42	17.61	29.05
HOTR [18]	HICO-Det	R50	✗	✓	25.10	17.34	27.42	-	-	-
AS-Net [3]	HICO-Det	R50	✗	✓	28.87	24.25	30.25	31.74	27.07	33.14
QPIC [38]	HICO-Det	R101	✗	✓	29.90	23.92	31.69	32.38	26.06	34.27
CDN-L [56]	HICO-Det	R101	✗	✓	32.07	27.19	33.53	34.79	29.48	36.38
MSTR [19]	HICO-Det	R50	✗	✓	31.17	25.31	32.92	34.02	28.83	35.57
SSRT [14]	HICO-Det	R50	✗	✓	31.34	24.31	33.32	-	-	-
DT [62]	HICO-Det	R50	✗	✓	31.75	27.45	33.03	34.50	30.13	35.81
STIP [58]	HICO-Det	R50	✓	✗	32.22	28.15	33.43	35.29	31.43	36.45
Iwin [41]	HICO-Det	R101	✗	✓	32.79	27.84	35.40	35.84	28.74	36.09
IF [30]	HICO-Det	R50	✗	✓	33.51	30.30	34.46	36.28	33.16	37.21
GEN [27]	HICO-Det	R101	✗	✓	34.95	31.18	36.08	38.22	34.36	39.37
Our w/o cues	HICO-Det	R50	✓	✓	33.07	29.87	34.05	35.21	32.04	37.09
Our w/ cues	HICO-Det	R50	✓	✓	36.75	33.53	37.71	39.84	35.58	40.23

Table 2: **Performance comparison on the HICO-Det test set.** We present an additional tag “Cues” to indicate the ability to flexibly use a variety of instance-level cues, as well as “Single” to denote a single-stage pipeline.

Method	Cues	AP ^{S1} _{role}	AP ^{S2} _{role}
iCAN [8]	✗	45.03	52.40
FCMNet [31]	✓	53.10	-
AS-Net [3]	✗	53.90	-
QPIC [38]	✗	58.80	60.00
Iwin [41]	✗	60.85	-
STIP [58]	✓	66.00	70.70
GEN [27]	✗	63.58	65.93
Ours	✓	65.68	69.72

Table 3: **Performance on the V-COCO.** Limited by space, the detailed comparison is listed in *supplementary materials*.

Method	Param.	GFIOPs	FPS
QPIC [38]	42.35M	36.95	20.0
AS-Net [3]	59.14M	52.94	1.6
STIP [58]	54.71M	48.27	1.6
Ours	44.47M	33.81	27.2

Table 4: **Analysis of efficiency.** All models are tested using a single GTX 1080Ti taking as input an image with a size of 640×640 . Here, we adopt ResNet50-FPN as the backbone.

position embedding to human and object centers respectively, the increased ambiguity leads to a 8.3% performance degradation. Besides, we observe a relative 6.8% degradation when invalidating the “cue-switch” strategy in cue aggregation module (Sec. 3.2), *i.e.*, treating all generated

instance tokens as valid without using the threshold γ to invalidate mis-recognized instances. Note that our utilization of CLIP is quite different from other methods. Concretely, other methods (*e.g.*, GEN [27]) perform CLIP to transfer interaction-specific linguistic knowledge to a visual model by using interaction (HOI-specific) labels to customize an interaction classifier, while we use just instance labels to generate general IRs. Actually, the majority of HOI detection methods use such general IRs since they are initialized using a pre-trained object detection or classification network.

Similarity metric. Tab. 5d compares different similarity metrics for our new objective function. When using cross-entropy (CE) solely, *i.e.*, involving no textural guidance, we observe severe performance degradation ($\approx 50\%$), indicating that simple CE loss cannot enable dynamical clustering. We conjecture that using CE loss is more like a recognition task that may introduce unexpected task bias, *i.e.*, highlighting partial features. In comparison, text representation is decoupled from downstream tasks and thus involves no task-bias. However, when adopting cosine similarity individually, we also observe a 18.9% performance degradation. It is because that the frozen text encoder of CLIP cannot differentiate two instances in the same category but with different attributes (*e.g.*, a standing human and a sit-

Stage 1	Stage 2	Full	Rare	Pattern	Full	Rare	Strategy	Full	Rare	metric	Full	Rare
(32, 32)	(4, 4)	30.19	26.24	1	34.81	29.40	Base	36.75	33.53	CE	19.82	14.53
(16, 64)	(4, 8)	36.75	33.53	3	36.75	33.53	- Center pos.	33.71	30.26	Cos	29.80	25.32
(32, 64)	(8, 8)	33.31	31.07	5	35.54	32.89	- Cue-Switch	34.26	29.82	CE+Cos	33.21	29.40
(32, 64)	(8, 16)	34.42	31.93	7	35.10	32.81	- CLIP	19.82	12.53	weighted	36.75	33.53

(a) Centers number.

(b) Patterns number.

(c) Training strategies.

(d) Similarity metric.

Table 5: **Ablations.** In (a), (·, ·) denotes (human,object); In (c) “-” means “w/o”. All experiments are conducted on HICO-Det test set.

ting human) as they are both labeled as “a photo of a human”. If we jointly train the text encoder and provide more fine-grained labels (e.g., a photo of a standing human), the results should be improved, yet introduce much more training complexity and annotation workload. In comparison, our proposed loss is a dynamical fusion of features’ generality (both human) and variability (with different attributes), which eliminates task-bias and also facilitates model training.

5. Discussion & Conclusion

Limitation. We find that clustering demands a relative higher resolution, so AGER struggles to handle small and occluded instances. Besides, our instance decoder enumerates all human-object pairs without considering interactivity. All of these await further exploration.

Conclusion. In this paper, we present AGER, a novel vision Transformer for HOI detection, which provides the first paradigm that enables Transformer-based HOI detector to leverage extra cues in an efficient (single-stage) manner. AGER performs tokenization as a text-guided dynamic clustering process, improving prior methods with instance tokens, which ensures the integrality of IRs. We validate AGER on two challenging HOI benchmarks and achieve a considerable performance boost over SOTA results.

Acknowledgments. This work was supported by NSFC 61831015, 62225112, 62176159, National Key R&D Program of China 2021YFE0206700, Natural Science Foundation of Shanghai 21ZR1432200 and Shanghai Municipal Science and Technology Major Project 2021SHZDZX0102.

References

- [1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.
- [2] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *WACV*, 2018.
- [3] Mingfei Chen, Yue Liao, Si Liu, Zhiyuan Chen, Fei Wang, and Chen Qian. Reformulating hoi detection as adaptive set prediction. In *CVPR*, 2021.
- [4] Qi Dong, Zhuowen Tu, Haofu Liao, Yuting Zhang, Vijay Mahadevan, and Stefano Soatto. Visual relationship detection using part-and-sum transformers with composite queries. In *ICCV*, 2021.
- [5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.
- [6] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.
- [7] Chen Gao, Jiarui Xu, Yuliang Zou, and Jia-Bin Huang. Drg: Dual relation graph for human-object interaction detection. In *ECCV*, 2020.
- [8] Chen Gao, Yuliang Zou, and Jia-Bin Huang. ican: Instance-centric attention network for human-object interaction detection. In *BMVC*, 2018.
- [9] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *CVPR*, 2018.
- [10] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015.
- [11] Tanmay Gupta, Alexander Schwing, and Derek Hoiem. No-frills human-object interaction detection: Factorization, layout encodings, and training techniques. In *ICCV*, 2019.
- [12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.
- [13] Zhi Hou, Xiaojiang Peng, Yu Qiao, and Dacheng Tao. Visual compositional learning for human-object interaction detection. In *ECCV*, 2020.
- [14] ASM Iftekhar, Hao Chen, Kaustav Kundu, Xinyu Li, Joseph Tighe, and Davide Modolo. What to look at and where: Semantic and spatial refined transformer for detecting human-object interactions. In *CVPR*, 2022.
- [15] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *ICLR*, 2017.
- [16] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *ECCV*, 2020.
- [17] Bumsoo Kim, Taeho Choi, Jaewoo Kang, and Hyunwoo J. Kim. UnionDet: Union-level detector towards real-time human-object interaction detection. In *ECCV*, 2020.
- [18] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *CVPR*, 2021.
- [19] Bumsoo Kim, Jonghwan Mun, Kyoung-Woon On, Minchul Shin, Junhyun Lee, and Eun-Sol Kim. Mstr: Multi-scale transformer for end-to-end human-object interaction detection. In *CVPR*, 2022.

- [20] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors. In *ECCV*, 2020.
- [21] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1955.
- [22] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *ICCV*, 2021.
- [23] 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 *CVPR*, 2020.
- [24] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, and Cewu Lu. Pastanet: Toward human activity knowledge engine. In *CVPR*, 2020.
- [25] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yanfeng Wang, and Cewu Lu. Transferable interactiveness knowledge for human-object interaction detection. In *CVPR*, 2019.
- [26] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. In *CVPR*, 2020.
- [27] Yue Liao, Aixi Zhang, Miao Lu, Yongliang Wang, Xiaobo Li, and Si Liu. Gen-vlkt: Simplify association and enhance interaction understanding for hoi detection. In *CVPR*, 2022.
- [28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
- [29] Xue Lin, Qi Zou, and Xixia Xu. Action-guided attention mining and relation reasoning network for human-object interaction detection. In *IJCAI*, 2020.
- [30] Xinpeng Liu, Yong-Lu Li, Xiaoqian Wu, Yu-Wing Tai, Cewu Lu, and Chi-Keung Tang. Interactiveness field in human-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20113–20122, 2022.
- [31] Yang Liu, Qingchao Chen, and Andrew Zisserman. Amplifying key cues for human-object-interaction detection. In *ECCV*, 2020.
- [32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017.
- [33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013.
- [34] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hour-glass networks for human pose estimation. In *ECCV*, 2016.
- [35] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, 2018.
- [36] 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 *ICML*, 2021.
- [37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.
- [38] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. In *CVPR*, 2021.
- [39] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *NIPS*, 2021.
- [40] Danyang Tu, Xiongkuo Min, Huiyu Duan, Guodong Guo, Guangtao Zhai, and Wei Shen. End-to-end human-gaze-target detection with transformers. In *CVPR*, 2022.
- [41] Danyang Tu, Xiongkuo Min, Huiyu Duan, Guodong Guo, Guangtao Zhai, and Wei Shen. Iwin: Human-object interaction detection via transformer with irregular window. In *ECCV*, 2022.
- [42] Danyang Tu, Wei Sun, Xiongkuo Min, Guangtao Zhai, and Wei Shen. Video-based human-object interaction detection from tubelet tokens. In *NeurIPS*, 2022.
- [43] Oytun Ulutan, ASM Iftekhhar, and Bangalore S Manjunath. Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions. In *CVPR*, 2020.
- [44] Bo Wan, Desen Zhou, Yongfei Liu, Rongjie Li, and Xuming He. Pose-aware multi-level feature network for human object interaction detection. In *ICCV*, 2019.
- [45] Hai Wang, Wei-shi Zheng, and Ling Yingbiao. Contextual heterogeneous graph network for human-object interaction detection. In *ECCV*, 2020.
- [46] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021.
- [47] Tiancai Wang, Rao Muhammad Anwer, Muhammad Haris Khan, Fahad Shahbaz Khan, Yanwei Pang, Ling Shao, and Jorma Laaksonen. Deep contextual attention for human-object interaction detection. In *ICCV*, 2019.
- [48] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *CVPR*, 2020.
- [49] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022.
- [50] Bingjie Xu, Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Interact as you intend: Intention-driven human-object interaction detection. *TMM*, 2019.
- [51] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022.
- [52] Xiaogang Xu, Hengshuang Zhao, Vibhav Vineet, Ser-Nam Lim, and Antonio Torralba. Mtfomer: Multi-task learning via transformer and cross-task reasoning. In *ECCV*, 2022.
- [53] Dongming Yang and Yuexian Zou. A graph-based interactive reasoning for human-object interaction detection. *IJCAI*, 2020.
- [54] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018.

- [55] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV*, 2022.
- [56] Aixi Zhang, Yue Liao, Si Liu, Miao Lu, Yongliang Wang, Chen Gao, and Xiaobo Li. Mining the benefits of two-stage and one-stage hoi detection. *NIP*, 34, 2021.
- [57] Frederic Z Zhang, Dylan Campbell, and Stephen Gould. Spatially conditioned graphs for detecting human-object interactions. In *ICCV*, 2021.
- [58] Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang-Wen Chen. Exploring structure-aware transformer over interaction proposals for human-object interaction detection. In *CVPR*, 2022.
- [59] Xubin Zhong, Changxing Ding, Xian Qu, and Dacheng Tao. Polysemy deciphering network for robust human-object interaction detection. In *ICCV*, 2021.
- [60] Xubin Zhong, Xian Qu, Changxing Ding, and Dacheng Tao. Glance and gaze: Inferring action-aware points for one-stage human-object interaction detection. In *CVPR*, 2021.
- [61] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.
- [62] Desen Zhou, Zhichao Liu, Jian Wang, Leshan Wang, Tao Hu, Errui Ding, and Jingdong Wang. Human-object interaction detection via disentangled transformer. In *CVPR*, 2022.
- [63] Penghao Zhou and Mingmin Chi. Relation parsing neural network for human-object interaction detection. In *ICCV*, 2019.
- [64] Tianfei Zhou, Wenguan Wang, Siyuan Qi, Haibin Ling, and Jianbing Shen. Cascaded human-object interaction recognition. In *CVPR*, 2020.
- [65] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.
- [66] Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, and Yichen Wei. End-to-end human object interaction detection with hoi transformer. In *CVPR*, 2021.