Supplementary Materials: Visual Relationship Detection Using Part-and-Sum Transformers with Composite Queries

Qi Dong Zhuowen Tu Haofu Liao Yuting Zhang Vijay Mahadevan Stefano Soatto
Amazon Web Services
{qdon,ztu,liahaofu,yutingzh,vmahad,soattos}@amazon.com

1. Part-and-Sum Transformers with Composite Queries

In this work, we focus on end-to-end structured data detection by Part-and-Sum Transformers for tasks like visual relationship detection and Human Object Interaction detection. We provide a more detailed discussion for the designs of **vanilla decoder, tensor-based decoder, and composite (part-and-sum) decoder**. These three alternatives differ in the form of queries and how attention is implemented.

Figure 1(a) in the main submission gives an illustration; for an input image, we use a convolutional neural network (CNN) model to extract image features, which are fed into a standard [?]/ deformable [?] transformer encoder. The encoder is composed of multiple self-attention layers to tokenize the visual features. After that, a transformer decoder takes the visual tokens together with a set of learnable queries as input to detect a composite set (visual relationships or human object interactions). We denote the tokenized features of the Transformer Encoder as $\boldsymbol{I}$, and the learnable queries as $\boldsymbol{Q}$; and the embedding of the outputs of a decoder as $\boldsymbol{E} = \text{Decoder}(\boldsymbol{Q}, \boldsymbol{I})$. For embeddings $\boldsymbol{E}$, the structural prediction $\boldsymbol{O}$ is inferred by a prediction module, denoted as $\boldsymbol{O} = \text{Prediction}(\boldsymbol{E})$.

1.1. Vanilla decoder with vector-based query

Our vanilla Transformer decoder contains M query embeddings, and each query is a vector, representing a relationship:

$$\boldsymbol{Q} = \{\boldsymbol{q}_1, ..., \boldsymbol{q}_M\}, \tag{1}$$

where $\boldsymbol{q}_i$ is a vector of a size $1 \times D$, and the overall query $\boldsymbol{Q}$ is $M \times D$. The queries are feed into multiple decoder layers of a same design. Specifically, each decoder layer contains a Multi-head self-attention layer [?], learning the cross-relationship context; and a multi-head cross-attention layer, to learn the representations by attending various image positions; and a feed forward network (FFN) to further embed each query. All query embeddings are feed into these three components one by one, and the last outputs are feed into the following decoder blocks. The decoding process in each decoder block is written as:

$$\begin{aligned} f(\boldsymbol{Q}) &= \text{SA}(\boldsymbol{q}_1, ..., \boldsymbol{q}_M) \\ \varphi(\boldsymbol{Q}, \boldsymbol{I}) &= \text{CA}([\boldsymbol{q}_1, ..., \boldsymbol{q}_M], \boldsymbol{I}), \end{aligned} \tag{2}$$

where f is Self-attention layer (SA), and φ is the Cross-attention layer (CA). Note that in vanilla Transformer, each query represents a relationship, i.e. containing multiple components.

1.2. Tensor-based decoder with tensor-based Query

Unlike vector based query, tensor-based query represents a relationship by a tensor which contains multiple sub-queries to represent each part individually, such as Subject, Predicate and Object parts. The tensor based query can be written as:

$$\boldsymbol{Q} = \{\boldsymbol{Q}_1, ..., \boldsymbol{Q}_M\} = \{\{\boldsymbol{q}_1^s, \boldsymbol{q}_1^p, \boldsymbol{q}_1^o\}, ..., \{\boldsymbol{q}_M^s, \boldsymbol{q}_M^p, \boldsymbol{q}_M^o\}\}, \tag{3}$$

where each query $\boldsymbol{Q}_i$ includes three sub-queries $\boldsymbol{q}_i^s, \boldsymbol{q}_i^p, \boldsymbol{q}_i^o$ to represent subject, predicate, and object, respectively. By doing so, all parts are learnt individually, reducing the ambiguity in similarity computation in attention schemes. It is important for learning the relationships sharing the same subject or object entity. In decoding, attention layers handle all sub-queries, written as:

$$\begin{aligned} f(\boldsymbol{Q}) &= \text{SA}(\boldsymbol{q}_1^s, \boldsymbol{q}_1^p, \boldsymbol{q}_1^o, ..., \boldsymbol{q}_M^s, \boldsymbol{q}_M^p, \boldsymbol{q}_M^o) \\ \varphi(\boldsymbol{Q}, \boldsymbol{I}) &= \text{CA}([\boldsymbol{q}_1, ..., \boldsymbol{q}_M], \boldsymbol{I}). \end{aligned} \tag{4}$$

Vector query and tensor-based query are conceptually different, and the former learns each two-level/structure data as a whole/Sum, while the latter learns each two-level/structure data by part learning. Furthermore, self-attention layers are functionally different in these two designs: self-attention among all sum queries is to mine inter-relation context, while self-attention layer among part queries is to mine the context of entities, which indirectly benefits relationship learning.

1.3. Composite (part-and-sum) decoder with composite Query

Composite query models each relationship in a structural manner, and learns a relationship in both part and sum levels. Each composite query contains three part sub-queries for Subject, Predicate and Object entities, and one sum sub-query for a whole relationship. The composite query can be written as:

$$\begin{aligned} \boldsymbol{Q} &= \{\boldsymbol{Q}_1, ..., \boldsymbol{Q}_M\} \\ \boldsymbol{Q}_i &= \{\boldsymbol{q}_i^s, \boldsymbol{q}_i^p, \boldsymbol{q}_i^o, \boldsymbol{q}_i^{\mathrm{G}}\}, \end{aligned} \tag{5}$$

where $\{\boldsymbol{q}_i^s, \boldsymbol{q}_i^p, \boldsymbol{q}_i^o$ are part queries, and $\boldsymbol{q}_i^{\mathrm{G}}$ is a sum query for relationship i. In the decoding, part query $\boldsymbol{Q}^P$ and sum query $\boldsymbol{Q}^G$ are separately decoded by different self-attention layers f_{Part} and f_{Sum}, and cross-attention layers φ_{Part} and φ_{Sum}, written as:

$$\begin{aligned} f_{\text{Part}}(\boldsymbol{Q}^P) &= \mathrm{SA}(\boldsymbol{Q}_1^P, ..., \boldsymbol{Q}_M^P) \\ \varphi_{\text{Part}}(\boldsymbol{Q}^P, \boldsymbol{I}) &= \mathrm{CA}([\boldsymbol{Q}_1^P, ..., \boldsymbol{Q}_M^P], \boldsymbol{I}), \end{aligned} \tag{6}$$

$$\begin{aligned} f_{\text{Sum}}(\boldsymbol{Q}^G) &= \mathrm{SA}(q_1^G, ..., q_M^G) \\ \varphi_{\text{Sum}}(\boldsymbol{Q}^G, \boldsymbol{I}) &= \mathrm{CA}([q_1^G, ..., q_M^G], \boldsymbol{I}) \end{aligned} \tag{7}$$

Factorized self-attention. To enhance part-based relationship learning, we designs a Factorized self-attention layer, which firstly conducts intra-relationship self-attention, and conducts inter-relationship self-attention. The intra-relationship self attention layer leverages the parts context to benefit relationship prediction, for example, subject query and object query are "person" and "horse" helps predict predicate "Ride". The inter-relationship self-attention layer leverages the inter-relationship context, to enhance the holistic relationship prediction per image. For example, the existence of "Person read book" helps infer the relationship "Person sit", rather than "Person run", which is particularly important for multiple interactions detection for same person entity. The Factorized self-attention is written as:

$$\begin{aligned} f_{\text{Part}}(\boldsymbol{Q}^P) &= \mathrm{FactorizedSA}(\boldsymbol{Q}_1^P, ..., \boldsymbol{Q}_M^P) \\ &= \text{Inter-relationSA}(\text{Intra-relationSA}(\boldsymbol{Q}^P)), \end{aligned} \tag{8}$$

where Intra-relation self-attention and Inter-relation self-attention layers are written as:

$$\begin{aligned} \text{Intra-relationSA}(\boldsymbol{Q}_i^P) &= \mathrm{SA}(q_i^s, q_i^p, q_i^o) \\ \text{Inter-relationSA}(\boldsymbol{Q}^P) &= \mathrm{SA}(\boldsymbol{Q}_1^P, ..., \boldsymbol{Q}_M^P) \end{aligned} \tag{9}$$

Note that the Factorized self-attention design also can be used for Tensor based query to enhance the inter part-query learning.

2. Visualisations of VRD and HOI detection

PST directly predicts all relationships in a set. Figure 1 shows exampled relationship detection results and human object interaction detection results by PST in (a) and (b). Each sub-image visualizes one predicted relationship. It shows that there exist multiple relationships between one entity-pair. For example, in Figure 1 (a), PST detects "Tower-has-clock" and "Clock-on-tower"; and "Road-under-tower" and "Tower-on-road".

3. More Qualitative Illustrations

In addition to the results shown in the main paper, we provide more qualitative results and analysis for visual relationship detection and human-object interaction detection by the proposed PST.

3.1. Small-entity relationship detection

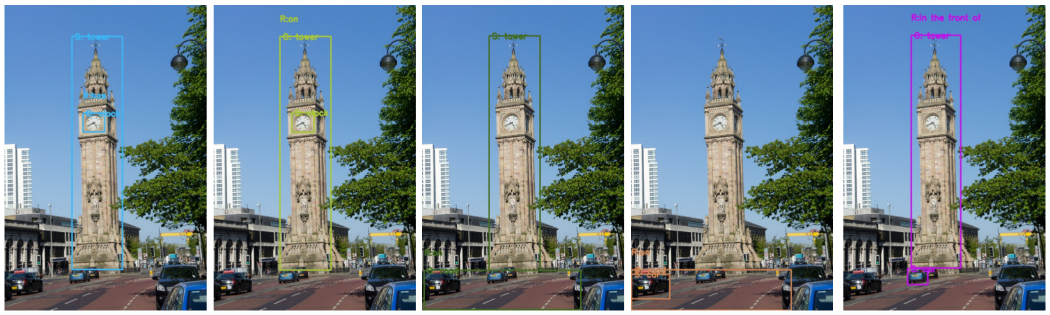
In the VRD task, relationships are composed of multiple types, some of which pose particular challenges, such as small-entity and spatial relationships. We show some examples in Figure 2 for small-entity relationship detection. PST is able to detect small subjects and objects well, such as "ball" in (a), "person" and "jacket" in (b), "clock" in (c) and "bag" in (d). This happens because the part queries are able to mine the subject-predicate-object context, and the sum queries leverage the inter-relation context; the two types of contextual information provide effective information for detecting small entities in a relationship.

3.2. Instance ambiguity in relationship detection

Instance ambiguity in relationship detection causes a detection failure, where the predicted relationships wrongly associate subject and object instances, although the categories of relationships are predicted correctly. For instance, in Figure 3 (a) and (b), there exist two same type relationships "Person-has-phone", and they are visually close. Relationship instance ambiguity makes it hard to associate each "phone" instances with the surrounding "Person" instances. This ambiguity is caused by that multiple relationship instances of the same relationship type are too close, and the visual clues for associating the subject-object instances are subtle. We examine PST in these challenging cases and show some examples in Figure 3. It shows that PST is able to associate subject-object instances correctly in this hard situation, thanks to the effective intra-relation and inter-relation attention for context modeling.

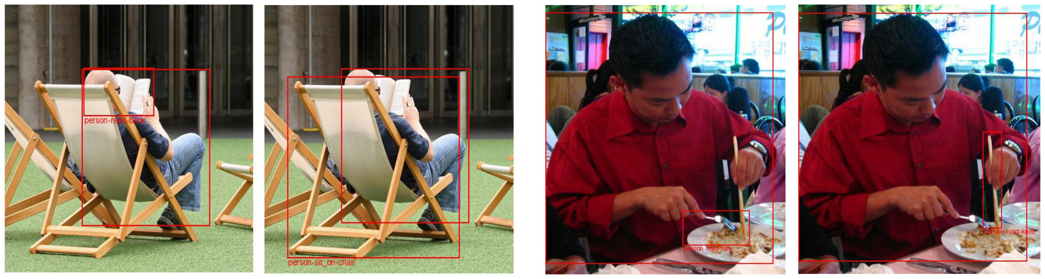
3.3. Composite attention visualisation

To better understand how the model works and what input information it uses to perform relationship detection, we visualize the cross-attention maps of the decoder layers of PST, since cross-attention measures the correlation between



Tower-has-clock *Clock-on-tower* *Tower-on-road* *Car-on-road* *Car-in the front of-tower*

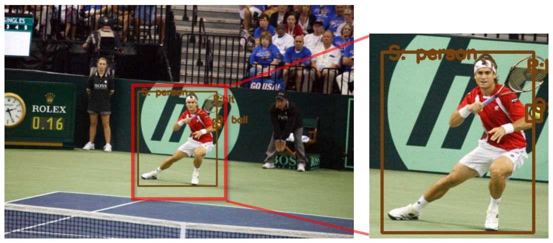
(a) VRD detection results



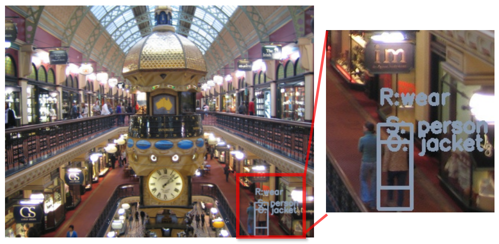
Person-read-book *Person-sit_on-chair* *Person-hold-fork* *Person-hold-knife*

(b) HOI detection

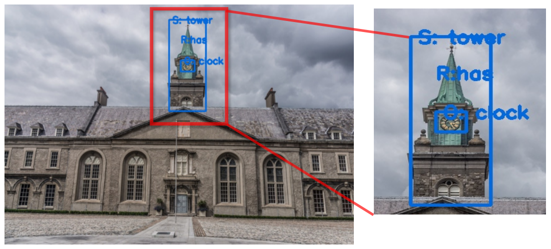
Figure 1: Qualitative results on (a) VRD and (b) HOI by PST. Each sub-image shows one predicted relationship/interaction. "R" refers to predicate; "S" refers to subject; and "O" refers to object.



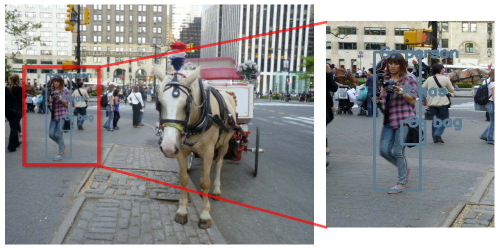
(a) *Person-hit-ball*



(b) *Person-wear-jacket*



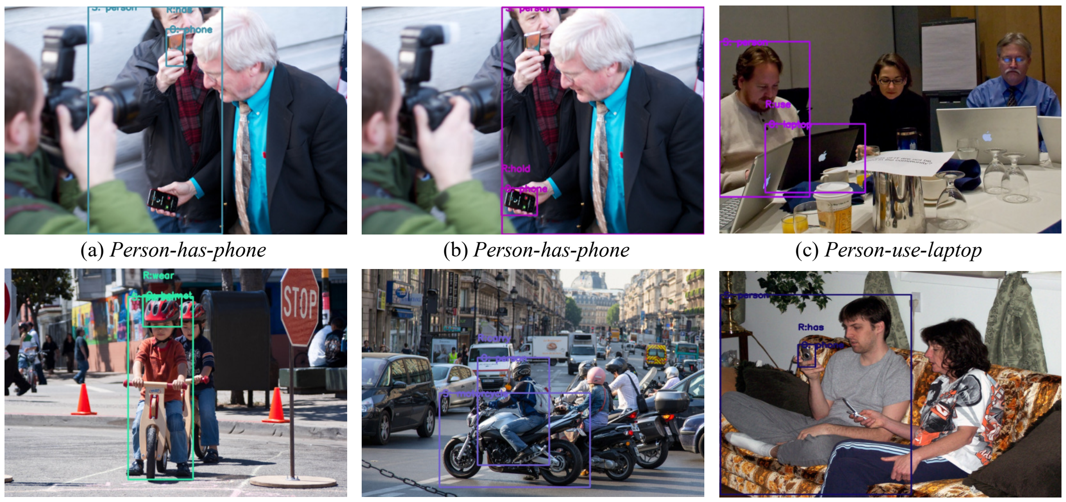
(c) *Tower-has-clock*



(d) *Person-has-bag*

Figure 2: Visualization of small-entity relationship detection. The exampled relationship predictions are (a) Person-hit-ball; (b) Person-wear-jacket; (c) Tower-has-clock; and (d) Person-has-bag. From it, PST is able to detect small subjects or objects in relationships and further detects the overall relationships properly. "R" refers to predicate; "S" refers to subject; and "O" refers to object.

the query embedding and image feature tokens. Given a test image, we extract cross-attention maps from all decoder layers for the subject, object and predicate query embedding individually. We visualize the attention maps of one query in Figure 4 (c), and include results of more queries in the supplementary material. In Figure 4, the relationship query embedding is decoded to *"person-wear-shoes"* semantically, and according to the attention maps, we can see that (1) the transformer decoder *incrementally* focuses on the "person" and "shoes" area in the image, to infer the


(a) *Person-has-phone* (b) *Person-has-phone* (c) *Person-use-laptop*

(d) *Person-wear-helmet* (e) *Motorcycle-carry-person* (f) *Person-has-phone*

Figure 3: Visualization of relationship detection with instance ambiguity. There exist multiple spatially close relationship instances of the same type. Specifically, there exists multiple same type and close relationships, for example, "person-has-phone" in (a) and (b); "person-use-laptop" in (c); "Person-wear-helmet" in (d), "Motorcycle-carry-person" in (e), and "person-has-phone" in (f). "R" refers to predicate; "S" refers to subject; and "O" refers to object.

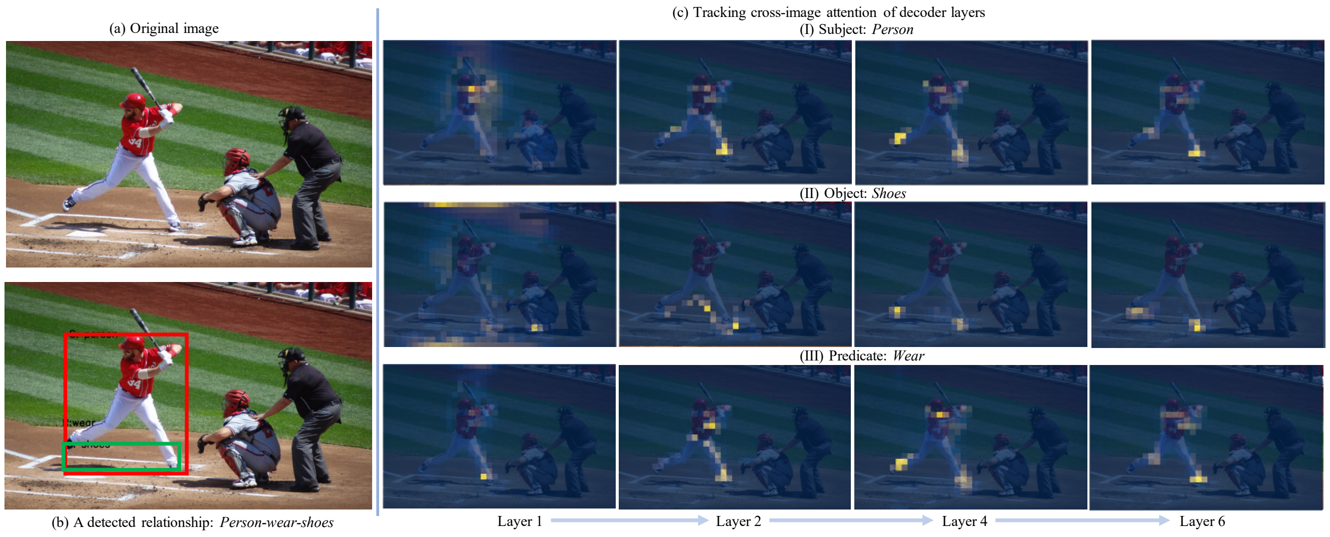
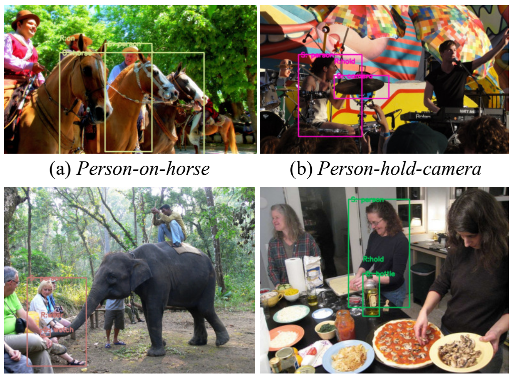


Figure 4: Visualization of the attention maps of decoder layers in PST. (a) is the input image, (b) shows a detected relationship from one query, and (c) visualizes this relationship's attention maps in various decoder layers for subject (I), object (II) and predicate(III) query, respectively. Due to the space limit, we just show attention maps of four decoder layers.

subject and object entities; (2) the predicate ("*Wear*") is mostly predicted from the union area of the subject and object, which suggests that the attention scheme is capable of automatically modeling the subject-object context for predicate detection.

3.4. Failure cases by PST

We visualize the typical errors of relationship detection by PST in Figure 5. There are four typical errors: (1) PST localizes entities inaccurately, when the entity instances are crowed. For instance, in Figure 5 (a), there are multiple horses close to each other, and PST localizes multiple horses as one Object entity of a relationship "Person-on-horse". (2) Object detection mistakes cause the failure in relationship detection, such as wrong entity "camera" detected in Figure 5 (b). (3) Relationship instance ambiguity challenges PST. For instance, in Figure 5 (c), the watch is associated with a wrong person instance which is very close



(a) *Person-on-horse* (b) *Person-hold-camera*

(c) *Person-wear-watch* (d) *Person-hold-bottle*

Figure 5: Visualization of a few failure cases by PST. There are four main relation detection types: (a) Inaccurate entity detection caused by crowed instances; (b) Wrong object detection; (c) Wrong association between subject and object instances; (d) Wrong predicate classification. “R” refers to predicate; “S” refers to subject; and “O” refers to object.

to the right person instance. (d) Predicate is wrongly predicted, for instance, PST classifies the relationship between “Person” and “bottle” as “hold”.