RelTransformer: A Transformer-Based Long-Tail Visual Relationship Recognition

Jun Chen, Aniket Agarwal, Sherif Abdelkarim, Deyao Zhu, Mohamed Elhoseiny

1 King Abdullah University of Science and Technology
2 Indian Institute of Technology

{jun.chen,deyao.zhu,mohamed.elhoseiny}@kaust.edu.sa
aagarwal@ma.iitr.ac.in, sherif.abdelkarim91@gmail.com

Abstract

The visual relationship recognition (VRR) task aims at understanding the pairwise visual relationships between interacting objects in an image. These relationships typically have a long-tail distribution due to their compositional nature. This problem gets more severe when the vocabulary becomes large, rendering this task very challenging. This paper shows that modeling an effective message-passing flow through an attention mechanism can be critical to tackling the compositionality and long-tail challenges in VRR. The method, called RelTransformer, represents each image as a fully-connected scene graph and restructures the whole scene into the relation-triplet and global-scene contexts. It directly passes the message from each element in the relation-triplet and global-scene contexts to the target relation via self-attention. We also design a learnable memory to augment the long-tail relation representation learning. Through extensive experiments, we find that our model generalizes well on many VRR benchmarks. Our model outperforms the best-performing models on two large-scale long-tail VRR benchmarks, VG8K-LT (+2.0% overall acc) and GQA-LT (+26.0% overall acc), both having a highly skewed distribution towards the tail. It also achieves strong results on the VG200 relation detection task. Our code is available at https://github.com/Vision-CAIR/RelTransformer.

1. Introduction

The Visual Relationship Recognition (VRR) task goes beyond recognizing individual objects by comprehensively understanding relationships between interacting objects in a visual scene. Owing to the enriched scene understanding provided by VRR, it benefits various other vision tasks such as image captioning (e.g., [10,46,47]), VQA (e.g., [15,40]), image generation (e.g., [16]), and 3D scene synthesis (e.g.,
Many previous approaches characterize the VRR problem under a graph scenario. Popular graph-based methods iteratively pass messages from other direct or indirect nodes to the relation along with the structure of the graph using the long short-term memory [39, 44, 50], or graph attention networks [24, 45]. However, the graph structure may implicitly constraint the relation to focus on its nearby neighbors. This phenomenon has been observed in recent works [2, 43], showing that graph neural networks incline to pay most attention to the local surrounding nodes but could not benefit much from distant nodes [2], and node representations will become indistinguishable if there are many layers [43]. Such problems can also be seen in long short-term memory networks due to their iterative message-passing learning nature. However, the target relation can also benefit a lot from the distant nodes. e.g., when we predict the “holding” relationship between “player” and “bat” in Fig. 1, the distant objects such as “catcher” and “umpire” can provide a context that the “player” is in a baseball game, and those can help the model better predict “holding” relationship.

To alleviate the aforementioned problems, we propose to adapt self-attention mechanisms originally introduced in Transformer [41] to tackle the VRR challenges. Self-attention can be viewed as a non-local mean operation [3], which computes the weighted average of all the inputs. When it applies to the VRR problem, it assumes that the relation has a full connection with all the other nodes in the graph and directly passes the messages among them via attention. In contrast to GNN/LSTM approaches, this strategy can allow the relation to have a larger attention scope and pass the message regardless of the graph structure or spatial constraints. It also avoids the valuable information from distant nodes being suppressed by nearby neighbors. Hence, each relation can selectively attend to its relevant features without spatial constraints and learn a richer-contextualized representation, which can benefit the long-tail visual relationship understanding.

In our approach, dubbed as RelTransformer, we reconstruct the scene graph into the relation-triplet and global-scene context as we demonstrated in the Fig. 1. The relation triplet here refers to the target relation and its referred subject and object, such as (player, holding, bat) in the figure. The global context represents all the relation triplets that are gathered for each appearing relation. We directly connect the target relation “holding” with every element from the relation triplet and global context, and pass their information to the target relation via self-attention. Furthermore, since the long-tail relations tend to be amenable to forgetting, we also propose a novel memory attention module to augment the relation representation with external persistent memory vectors, as we will detail later.

We showcase the effectiveness of our model on VG200 [20] and two recently proposed large-scale long-tail VRR benchmarks, GQA-LT [1] and VG8K-LT [1]. GQA-LT and VG8K-LT scale the number of relation types up to 300 and 2,000 compared to only 50 relation types in VG200. These two benchmarks are highly skewed (e.g., the VG8K-LT benchmark ranges from 14 to 618,687 examples per relation type) and offer us a suitable platform for studying long-tail VRR problems. Our approach achieves the state-of-the-art on those three datasets in our experimental results, demonstrating its effectiveness. We also conducted several ablative experiments and showed the usefulness of each component design in RelTransformer.

2. Background

Visual Relationship Recognition. Correct visual relation prediction requires having a comprehensive understanding of the image contents, which guides many successful works in the literature. Early works employ RNN models to construct a global context by aggregating the node and edge features via iterative message passing such as [14, 22, 39, 44, 50]. e.g., VCTree [39] composes a dynamic tree structure to organize the object orders and apply TreeLSTM [36] to aggregate features. There are also several graph convolutional network (GCN) [18] approaches [5, 21, 30, 45], which attempt to learn different importance weights to the neighborhood nodes. Lin et al. [24] extend the graph attention to also capture the node-specific contextual information and encode the edge direction information. More recently, there is also emerging a Transformer-based approach [19], which models the pairwise interaction among nodes and edges in two separate Transformer networks. Our model differs from it mainly in two aspects: a) we have a different message-passing flow in which we specifically aggregate relation features from the relation-triplet and global-context information. b) We further design an effective memory attention module for augmenting the long-tail relation representation.

Long-Tail Visual Relationship Recognition. Long-tail problem is very severe in visual relation recognition (VRR) [1, 24, 51]. There are mainly two approaching directions to alleviate this problem. The first one is semantically guided visual recognition. In this case, language models [27] are employed for zero-shot or few-shot recognition [11, 29]. There are also several VRR works [1, 45, 48, 51] using the language priors as a guidance to learn relation features, which can derive a better classification on long-tail classes. The second direction is to apply the strategies that have been designed for unbalanced object detection, including various class imbalance loss functions (e.g. weighted cross entropy, focal loss [23], equalization loss [37]), sampling
3. Approach

3.1. Problem Definition

An image can be decomposed into a scene graph \( G = (N, E) \), where each node \( n_i \in N \) represents an object and each edge \( e_i \in E \) represents the spatial or semantic relationships between two interacting objects. We denote the visual relationship between a subject \( n_s \) and an object \( n_o \) as \( r \). In the visual relationship recognition (VRR) task, the goal is to predict \( r \) between the given \( n_s \) and \( n_o \).

\[
y^r = f(b^s, b^o, b^r, I)
\]  

(1)

where \( b^s \), \( b^o \), and \( b^r \) are the subject, object and relationship bounding boxes. \( b^r \) is obtained by the minimum enclosing region of \( b^s \) and \( b^o \). \( y^r \) is the relationship label. \( I \) denotes the arbitrary information such as image raw RGB pixel features. \( f \) is the inference model.

3.2. RelTransformer Architecture

RelTransformer mainly comprises two components, an \( L \)-layer global-context encoder and a \( P \)-layer relational encoder. The overall architecture can be seen in Fig. 2. Given an image, we first extract the object and relation features via a faster R-CNN detection network [32]. Those features are grouped together as a tuple of triplets \((s_1, r_1, o_1), \ldots, (s_N, r_N, o_N)\) according to their spatial or semantic relationships provided in the dataset, where each \( s_i, r_i \) and \( o_i \in \mathbb{R}^{1 \times D} \) and \( N \) is the number of triplets. We first feed all the triplets into a multi-layer global-context encoder to learn a scene-contextualized representation. Then we concentrate on learning the target relation \( r \); representation from the context of \((s_i, r_i, o_i)\) in the relational encoder. An external memory module is also introduced here to augment the long-tail relation representations. Finally, we employ a meshed connection (see Fig. 2) to integrate global-context features into the relation representation learning.

Global-Context Encoder. We employ an \( L \)-layer Transformer [41] to model the global-context information and learn their pairwise relationships. We first concatenate each triplet from \((s_1, r_1, o_1), \ldots, (s_N, r_N, o_N)\) together into a compact representation as \( X^{mo} = (x_1^{mo}, \ldots, x_N^{mo}) \) where
\(X^{wo} \in \mathbb{R}^{N \times 3D}\). We then feed \(X^{wo}\) into the global-context encoder in a permutation-invariant order.

The Transformer \([41]\) is a stack of multi-headed self-attention (MSA) and MLP layers. Its core component is the self-attention as defined in Eq. 2. In each Transformer encoding layer, the multi-head self-attention repeats the self-attention multiple times and concatenate the results together; the result is then projected back to the same dimensionality. After that, the result is fed into an MLP network and produces each layer’s output.

\[
f_{sa}(Q, K, V) = \text{softmax}\left( \frac{(W_q Q)(W_k K)^\top}{\sqrt{d}} \right) W_v V \tag{2}\]

where \(Q, K\) and \(V \in \mathbb{R}^{t \times h}\) are query, key and value vectors, \(t\) is the number of input tokens and \(h\) is the hidden size, \(d\) is a scaling factor. \(W_q, W_k\) and \(W_v\) are the learnable weight parameters.

We feed \(X^{wo}\) into the global-context encoder, each layer \(l\) outputs a contextualized representation \(z_l\). We gather them together as \(Z = (z_1, \ldots, z_L)\) where \(z_l \in \mathbb{R}^{N \times h}\).

**Relational Encoder.** It has multiple functions: 1) it specifically aggregates the relation representation from its referred subject and object via self-attention 2) it incorporates a persistent memory to augment the relations with “out-of-context” information, which could especially benefit the long-tail relations. 3) it also has a “bridging” role to aggregate each global-context encoding layer’s output into the relation representation through a meshed attention module. The detailed operating procedure is described as follows.

We first add a learnable positional embedding \([8]\) to each position of \(s_i, r_i\), and \(o_i\) in order to distinguish their semantic differences in the input sequence, and we denote this sequence as \(X_p\) in the layer \(p\). We apply the self-attention defined in Eq. 2 on \(X_p\) in each layer \(p\) and model their pairwise relationship. Their result will be \(X_p^{att}\) as shown in Eq. 3.

\[
X_p^{att} = f_{sa}(X_{p-1}, X_{p-1}, X_{p-1}) \tag{3}\]

**Memory Augmentation.** The trained model can easily forget long-tail relations because the model training is dominated by the instance-rich (or head) relations, and hence it tends to underperform on lower-frequent relations. Also, the self-attention is limited to attending to the features only from the tokens in an input sequence; hence, each relation only learns a representation based on the immediate context. To alleviate this issue, we propose a novel memory attention module motivated by several successful persistent memory ideas \([35, 53]\) in the literature. We denote a group of persistent and differentiable memory vectors as \(M\). Each time the relation passes its features to \(M\) and retrieves the information from \(M\) via attention. The memory here captures the information not dependent on the immediate context; instead, it is shared across the whole dataset \([35]\). Through this way, long-tail relations are able to access the information (e.g., from other relations or itself in different training steps) with relevance to itself, and they can augment the target relation with the useful “out-of-context” information in a well-trained model.

To compute this memory, we first randomly initialize \(m\) memory vectors in each relational encoding layer \(p\) as \(M_p \in \mathbb{R}^{m \times h}\). We then compute the memory attention between the input feature with \(M_p\), we treat \(X_{p-1}\) as the query, and \(M\) here is the Key and Value. Same self-attention operation is applied here in Eq. 4 and we can obtain \(X_p^{mem}\). The memory is directly updated via SGD.

\[
X_p^{mem} = f_{sa}(X_{p-1}, M_{p-1}, M_{p-1}) \tag{4}\]

To aggregate \(X_p^{mem}\) into \(X_p^{att}\), we design a fusion function as \(g(x, y)\) in Eq. 5. \(g(x, y)\) is an attention gate and determines how to effectively combine two input features. It computes the complementary attention weights to each input and weighted combine them as the output. Through this fusion function in combination with a skip connection, we can get the fused feature as \(\hat{X}_p = g(X_p^{att}, X_p^{mem}) + X_{p-1}\).

\[
g(x, y) = \alpha \odot x + (J - \alpha) \odot y
= \sigma(W[x; y] + b) \tag{5}\]

where \(W\) is a 2D \(\times D\) matrix, \(b\) is a bias term. \([;]\) denotes the concatenation. \(\odot\) is the Hadamard product. \(J\) is an all-one matrix with the same dimensions as \(\alpha\).

**Meshed Attention Fusion.** The features from different global-context encoding layers capture different vision granularity, and leveraging features from all of them has shown to be better than the one only from the last encoding layer\([4, 7]\). Therefore, we adopt a meshed connection in our model and contribute each layer’s output \(z_l\) to the relation representation. To compute the meshed attention, we first compute the cross-attention between \(\hat{X}_p\) and each global-context encoding output \((z_1, \ldots, z_L)\); Its attention output is fused with \(\hat{X}_p\) through Eq. 5. Their results are averaged up for each layer. We then project the averaged output in an MLP network and incorporate a skip connection to compute the final fused relation representation \(X_p\) in Eq. 6 as the layer output.

\[
z^l_p = f_{sa}(\hat{X}_p, z_l, \hat{z}_l)\]

\[
X_p = \text{MLP} \left( \frac{1}{L} \sum_{l=1}^L g(\hat{X}_p, z^l_p) \right) + \hat{X}_p \tag{6}\]

**Final Triplet Prediction.** In the last layer \(P\) of the relational encoder, we extract subject \(x_s\), relation \(x_r\) and object \(x_o\) from \(X_P\) accordingly. In the prediction stage, we leverage the language prior knowledge following previous
4. Experiments

4.1. Datasets

We evaluate our model on the VG200 dataset and two large-scale long-tail VRR datasets, named GQA-LT [1] and VG8K-LT [1].

**GQA-LT.** This dataset contains 72,580 training, 2,573 validation and 7,722 testing images. Overall, it contains 1,703 objects and 310 relationships. The GQA-LT has a heavy “long-tail” distribution where the example numbers per each class range from merely 1 to 1,692,068.

**VG8K-LT.** It is collected from Visual Genome (v1.4) [20] dataset, containing 97,023 training, 1,999 validation and 4,860 testing images. It covers 5,330 objects and 2,000 different relation types in total, in which least frequent objects/relationships only have 14 examples and the most frequent ones have 618,687 examples.

**VG200.** This dataset has been widely studied in the literature [44, 51, 52]. It contains 50 relationships, and the category frequency in this dataset is considerably more balanced than that in GQA-LT and VG8K-LT. We follow the same data split as in [51] in our experiment.

4.2. Experimental Settings

**GQA-LT & VG8K-LT Baselines.** We compare RelTransformer with several state-of-the-art models. The most popular models in this benchmark are implemented based on LSVRU [51] framework. To improve over the long-tail performance, baseline models usually are combined with the following strategies: 1) class-imbalance loss functions such as weighted cross entropy (WCE), equalization loss (EQL) [37], focal loss [23], and ViLHub loss [1]. 2) relation augmentation strategies such as RelMix [1] to augment more examples in long-tail relations. 3) Decoupling [17] which decouples the learning procedure into representation learning and classification. 4) Counterfactual [38], which alleviates the biased scene graph generation via the counterfactual learning. 5) OLTR [25], which has memory module with augmented attention.

**VG200 Baselines.** We compare with several strong baselines including Visual Relationship Detection [26], Message Passing [44], Associative Embedding [28], MotifNet [50], Permutation Invariant Predication [14], LSVRU [51], relationship detection with graph contrastive loss (RelDN) [52], GPS-Net [24], Visual Relationship Detection with Visual-Linguistic Knowledge (RVL-BERT) [6] and Relational Transformer Network (RTN) [19].

**Evaluation Metrics.** For the GQA-LT and VG8K-LT datasets, we report the average per-class accuracy, which is commonly used for long-tail evaluation [1, 17, 37]. Following the same evaluation setting as [1], we split the relationship classes into the many, medium, and few based on the relation frequency in the training dataset as shown in Table 1. For the VG200 dataset, following the previous evaluation setting in [34, 51], we measure the Recall@K and mean Recall@K for predicate classification (PRDCLS), which is to predict the relation labels given the ground truth boxes and labels of the subject and object.

```
<table>
<thead>
<tr>
<th>Architecture</th>
<th>Learning Methods</th>
<th>VG8K-LT</th>
<th>GQA-LT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>many</td>
<td>medium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>LSVRU</td>
<td>ViHub [1]</td>
<td>27.5</td>
<td>17.4</td>
</tr>
<tr>
<td>LSVRU</td>
<td>ViHub + RelMix [1]</td>
<td>24.5</td>
<td>16.5</td>
</tr>
<tr>
<td>LSVRU</td>
<td>OLTR [25]</td>
<td>22.5</td>
<td>15.6</td>
</tr>
<tr>
<td>LSVRU</td>
<td>EQL [37]</td>
<td>22.6</td>
<td>15.6</td>
</tr>
<tr>
<td>LSVRU</td>
<td>Counterfactual² [38]</td>
<td>12.1</td>
<td>25.6</td>
</tr>
<tr>
<td>LSVRU</td>
<td>CE</td>
<td>22.2</td>
<td>15.5</td>
</tr>
<tr>
<td>RelTransformer</td>
<td>CE</td>
<td>26.8</td>
<td>18.6</td>
</tr>
<tr>
<td>RelTransformer</td>
<td>Focal Loss [23]</td>
<td>24.5</td>
<td>16.2</td>
</tr>
<tr>
<td>RelTransformer</td>
<td>Focal Loss</td>
<td>30.5</td>
<td>22.8</td>
</tr>
<tr>
<td>LSVRU</td>
<td>DCPL [17]</td>
<td>34.3</td>
<td>15.4</td>
</tr>
<tr>
<td>RelTransformer</td>
<td>DCPL</td>
<td>37.3</td>
<td>27.6</td>
</tr>
<tr>
<td>LSVRU</td>
<td>WCE</td>
<td>35.5</td>
<td>24.7</td>
</tr>
<tr>
<td>RelTransformer</td>
<td>WCE</td>
<td>36.6</td>
<td>27.4</td>
</tr>
</tbody>
</table>
```

Table 1. Average per-class accuracy in relation prediction on VG8K-LT and GQA-LT datasets. We evaluate the average per-class accuracy for many, medium, few, and all classes. The best performance for each column is underlined. ² denotes our reproduction. Learning methods include various class-imbalance loss functions, augmentation and counterfactual approaches. Our model is marked in gray.
considerable margin of WCE improves the medium and few category in GQA-LT by a few classes. The combination of RelTransformer with the addition of RelTransformer, especially on the medium is a clear performance improvement over all the baselines results for GQA-LT and VG8K-LT datasets in Table 1. There GQA-LT and VG8K-LT Evaluation. We present our re-

4.3. Quantitative Results

GQA-LT and VG8K-LT Evaluation. We present our results for GQA-LT and VG8K-LT datasets in Table 1. There is a clear performance improvement across all the baselines with the addition of RelTransformer, especially on the medium and few classes. The combination of RelTransformer with WCE improves the medium and few category in GQA-LT by a considerable margin of \( \approx 20\% \) compared to all the baselines. This huge gain can be attributed to the weighted assignment of different classes when WCE loss is applied. This further refines the attention weights assigned to different classes in the global context and hence helps the overall performance. While previous works [38] improved the tail performance at the cost of head class accuracy, RelTransformer consistently improves on the tail as well as the head as seen from the table, underlying the effectiveness of our model.

For VG8K-LT, we also see performance gains with the addition of RelTransformer on all baselines across “many”, “medium” and “few” categories. A considerable improvement of \( \approx 5\% \) can be seen when RelTransformer is combined with DCPL [17], performing the best for the VG8K-LT dataset. While we see consistent improvements with the addition of RelTransformer for the VG8K-LT dataset, the improvement margins are certainly lower as compared to GQA-LT as seen in Table 1. This is due to the more challenging nature of VG8K-LT, containing 2000 relation classes compared to 300 classes present in GQA-LT. Some qualitative examples with the addition of RelTransformer can be seen in Fig. 5.

VG200 Evaluation. We also evaluate our model on VG200 dataset in Table 2. We compare with many different message-passing approaches in our baselines including RNN-based [39, 50], GCN-based [24, 45] and Transformer-based [6, 19]. These two Transformer-based approaches either only focus on the relational triplet context [6] or neglect it completely [19]. Our model differs from them with a different message-passing strategy, different context construction and a novel memory attention. The experimental results show that our method can better exploit the relational features, and we improve over the best-performing baselines by 0.2% on R@20, 1.0% on R@50, and 0.9% on R@100.

Mean Recall@K on VG200. We evaluate the mean recall@k performance on the VG200 dataset for relation prediction, and compared RelTransformer with several strong baselines such as VCTREE and Motif with both cross-entropy and EBM losses [34]. The results are summarized in Table 3. We observe that RelTransformer can outperform all the baselines on mR@(20, 50, 100) while only being combined with cross-entropy loss, which shows its robustness on other data-imbalanced datasets.

4.4. Further Analysis

To analyze our results in more depth, we quantify our model’s improvement per each class and visualize them in Fig. 4. We provide the contrast between RelTransformer and LSVRU with cross entropy loss. From the figure, we can observe that RelTransformer can improve the the majority of classes on both datasets. In particular, RelTransformer improves 173 relations while only worsening 32 ones on VG8K-LT dataset with most performance gain from the medium and few classes.

GQA-LT and VG8K-LT Compositional Prediction. The compositional prediction is the correct prediction of the subject, relation, and object together. This could trigger a more skewed long-tail distribution due to its combinatorial nature. To evaluate our model’s compositional behaviors, we follow the work [1] to group the classification results by the pairs of (subject, object), (subject, relation), and (object, relation). The results are provided in Table 4, and we can see noticeable performance improvement on all the classes in contrast to the baselines. But we also observe that both RelTransformer(CE) and LSVRU (CE) perform better than the ones combining with focal loss and WCE in many categories, which differs from the results for only predicting the relations. The main reason is that those class imbalance losses hurt more “head” performance on subject/object..

<table>
<thead>
<tr>
<th>Models</th>
<th>R@20</th>
<th>R@50</th>
<th>R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRD [26]</td>
<td>-</td>
<td>27.9</td>
<td>35.0</td>
</tr>
<tr>
<td>Message Passing [44]</td>
<td>52.7</td>
<td>59.3</td>
<td>61.3</td>
</tr>
<tr>
<td>Associative Embedding [28]</td>
<td>47.9</td>
<td>54.1</td>
<td>55.4</td>
</tr>
<tr>
<td>MotifNet (Left to Right) [50]</td>
<td>58.5</td>
<td>65.2</td>
<td>67.1</td>
</tr>
<tr>
<td>Permutation Invariant [14]</td>
<td>-</td>
<td>65.1</td>
<td>66.9</td>
</tr>
<tr>
<td>LSVRU [51]</td>
<td>66.8</td>
<td>68.4</td>
<td>68.4</td>
</tr>
<tr>
<td>RelDN [52]</td>
<td>66.9</td>
<td>68.4</td>
<td>68.4</td>
</tr>
<tr>
<td>Graph-RCNN [45]</td>
<td>-</td>
<td>54.2</td>
<td>59.1</td>
</tr>
<tr>
<td>VCTREE-SL</td>
<td>59.8</td>
<td>66.2</td>
<td>67.9</td>
</tr>
<tr>
<td>GPS-Net [24]</td>
<td>60.7</td>
<td>66.9</td>
<td>68.8</td>
</tr>
<tr>
<td>RVL-BERT [6]</td>
<td>-</td>
<td>62.9</td>
<td>66.6</td>
</tr>
<tr>
<td>RTN [19]</td>
<td>68.3</td>
<td>68.7</td>
<td>68.7</td>
</tr>
<tr>
<td>RelTransformer (ours)</td>
<td>68.5</td>
<td>69.7</td>
<td>69.7</td>
</tr>
</tbody>
</table>

Table 2. Relation prediction on VG200 dataset.
### Table 4. Relationship triplet performance on GQA-LT dataset. SO = (subject, object), SR = (subject, relation) and OR = (object, relation).

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Learning Methods</th>
<th>many</th>
<th>medium</th>
<th>few</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SO</td>
<td>SR</td>
<td>OR</td>
</tr>
<tr>
<td>LSVRU</td>
<td>VilHub</td>
<td>40.5</td>
<td>32.8</td>
<td>33.7</td>
</tr>
<tr>
<td>LSVRU</td>
<td>CE</td>
<td>38.6</td>
<td>30.3</td>
<td>31.5</td>
</tr>
<tr>
<td>RelTransformer</td>
<td>CE</td>
<td>54.2</td>
<td>46.6</td>
<td>47.2</td>
</tr>
<tr>
<td>LSVRU</td>
<td>Focal Loss</td>
<td>39.2</td>
<td>31.1</td>
<td>32.3</td>
</tr>
<tr>
<td>LSVRU</td>
<td>WCE</td>
<td>18.3</td>
<td>17.3</td>
<td>17.2</td>
</tr>
<tr>
<td>RelTransformer</td>
<td>WCE</td>
<td>19.2</td>
<td>20.0</td>
<td>19.5</td>
</tr>
</tbody>
</table>

4.5. Ablation Studies

To quantify the contributions of each component to the whole model performance, we ablate and evaluate our RelTransformer in different versions on VG8K-LT dataset as shown in Table 5. We choose VG8K-LT instead of GQA-LT since it is more challenging and covers more classes.

The role of global-context encoder. To investigate the effect of the global-context encoder, we ablate our RelTransformer with the version without learning global context. The results in Table 5 indicate that the performance will drop on all the categories if we exclude it. It brings the performance down by 1.45% accuracy (acc) on many, 2.35% acc on medium, and 1.13% acc on few when we average all the combined loss functions’ results. This analytic shows that incorporating global context can benefit all categories with gaining the most performance on medium classes. We also demonstrate its per-class analysis in Fig. 4, from which we can observe that the performance drops in most classes.

The role of memory attention. The persistent memory vectors are aimed to augment the relation representation compared to CE (see supplementary) and they are reflected in the compositional prediction.

<table>
<thead>
<tr>
<th>Models</th>
<th>Losses</th>
<th>many</th>
<th>med</th>
<th>few</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full model CE</td>
<td></td>
<td>26.8</td>
<td>18.6</td>
<td>15.0</td>
<td>16.1</td>
</tr>
<tr>
<td>✗ global CE</td>
<td></td>
<td>24.8</td>
<td>16.9</td>
<td>13.9</td>
<td>14.8</td>
</tr>
<tr>
<td>✗ mem CE</td>
<td></td>
<td>26.0</td>
<td>16.4</td>
<td>14.0</td>
<td>15.0</td>
</tr>
<tr>
<td>Full model</td>
<td>Focal Loss</td>
<td>30.5</td>
<td>22.8</td>
<td>14.8</td>
<td>16.8</td>
</tr>
<tr>
<td>✗ global Focal Loss</td>
<td></td>
<td>28.9</td>
<td>19.8</td>
<td>13.3</td>
<td>15.1</td>
</tr>
<tr>
<td>✗ mem Focal Loss</td>
<td></td>
<td>30.1</td>
<td>20.8</td>
<td>14.2</td>
<td>15.9</td>
</tr>
<tr>
<td>Full model</td>
<td>DCPL</td>
<td>37.3</td>
<td>27.6</td>
<td>16.5</td>
<td>19.2</td>
</tr>
<tr>
<td>✗ global DCPL</td>
<td></td>
<td>36.1</td>
<td>25.4</td>
<td>15.7</td>
<td>18.2</td>
</tr>
<tr>
<td>✗ mem DCPL</td>
<td></td>
<td>37.0</td>
<td>26.8</td>
<td>16.1</td>
<td>18.8</td>
</tr>
<tr>
<td>Full model</td>
<td>WCE</td>
<td>36.6</td>
<td>27.4</td>
<td>16.3</td>
<td>19.0</td>
</tr>
<tr>
<td>✗ global WCE</td>
<td></td>
<td>35.6</td>
<td>24.5</td>
<td>15.2</td>
<td>17.6</td>
</tr>
<tr>
<td>✗ mem WCE</td>
<td></td>
<td>36.5</td>
<td>27.2</td>
<td>16.0</td>
<td>18.7</td>
</tr>
</tbody>
</table>

Table 5. Ablation study of RelTransformer on VG8K-LT dataset. global and mem represent the global-context encoder and memory attention module, respectively. ✗ represents the removal operation. Our default setting is marked in gray.
with more useful “out-of-context” information, especially for the low-frequent relations. In Table 5, we evaluate the version without the memory attention, and we can observe that it brings a performance reduction for all the categories. It drops the performance by 0.4% acc on many, 1.2% acc on medium, and 0.58% acc on few relations when we average the results from all the loss functions. The medium and few are more influenced than many relations, indicating that memory attention can benefit the infrequent relation classes more. We also show the per-class analysis with removing the memory in Fig. 4, where we can find the performances for medium and few relations drop most.

4.6. Qualitative Results

We randomly sample 2 images which include long-tail relationships from the testing dataset. We evaluate RelTransformer and LSVRU [51] in terms of CE and WCE loss functions and contrast them in Fig. 5. It can be observed that LSVRU predicts very trivial relationships such as “behind”, “under,” or “to the left of”. Those trivial relationships convey a vague and inaccurate understanding of the image content. Our RelTransformer instead can describe the relations more accurately. e.g., it predicts “man sniffing wine” for the bottom image. “sniffing” is a long-tail relation type and this prediction exactly matches the ground truth. We also randomly sample 3 images from VG200 dataset and visualize our prediction results in Fig. 6.

5. Conclusion

We presented a Transformer-based long-tail visual relationship recognition model, dubbed RelTransformer, which directly connects the relations with all the visual objects via the attention mechanism. Empirically, we organize the whole scene into the relation-triplet and global-scene context, and attentively aggregate their information to the relation representation under our message-passing flow. We also propose a memory attention module to augment the relation representation with “out-of-context” information, which is shown to be more effective for infrequent relations. With our design, RelTransformer surpasses all previous stat-of-the-art results on GQA-LT, VG8K-JT, and VG200 datasets.
References


[40] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In \textit{Proceedings of the IEEE conference on computer vision and pattern recognition}, pages 1–9, 2017. 1


[51] Ji Zhang, Yannis Kalantidis, Marcus Rohrbach, Manohar Paluri, Ahmed Elgammal, and Mohamed Elhoseiny. Large-scale visual relationship understanding. In \textit{Proceedings of