Segmentation-Guided Transformer Network for Subtle Visual Relationship Detection

Fan Yu[1], Hanxi Cao[1], Ruichao Hou[1*], Tongwei Ren[1]

[1]State Key Laboratory for Novel Software Technology, Nanjing University, 22 Hankou Road, Nanjing, 210093, Jiangsu, China.

*Corresponding author(s). E-mail(s): rchou@nju.edu.cn;
Contributing authors: yf@smail.nju.edu.cn; chx@smail.nju.edu.cn; rentw@nju.edu.cn;

Abstract

In real-world images, low-resolution regions caused by downsampling or compression often exhibit severe pixelation and detail loss, which substantially degrades visual relationship detection (VRD). Most existing VRD approaches primarily rely on fine-grained appearance cues and therefore perform poorly when entities are small or visually ambiguous. To address this limitation, we propose a Segmentation-Guided Transformer Network (SGTN) designed for robust detection of subtle visual relationships under low-resolution conditions. SGTN explicitly incorporates segmentation information and leverages pixel-level semantics, such as entity masks, to compensate for detail loss in low-resolution areas, enabling joint entity and relation prediction within a single-stage Transformer architecture. The framework consists of three modules: a two-stream feature extraction module employing visual and textual encoders, a multimodal feature fusion module based on an encoder-decoder structure that deeply integrates visual and textual features to generate enhanced entity, relation, and mask features, and an entity-relation prediction module that uses the fused features for entity detection and relationship classification. Experiments on downsampled *VG* benchmarks *VG_2*, *VG_5*, and *VG_10* show that SGTN achieves the best performance in most settings, with particularly notable gains under severe downsampling.

Keywords: Subtle Visual Relationship Detection, Segmentation-Guided Transformer, Low-Resolution Images, Multimodal Feature Fusion

1 Introduction

Existing image visual relationship detection methods tend to detect visual relationships relevant to regions characterized by high resolution and clear details [1, 2]. However, in practical applications, images may contain low-resolution regions that are challenging to perceive. These regions may still encompass important visual relationships, such as those between entities within low-resolution regions or between low-resolution entities and others. In large field-of-view images, such as surveillance footage from city squares, transportation hubs, or large warehouses, the physical resolution limitations of imaging devices often result in severe resolution degradation for distant targets or small regions due to insufficient pixel density per unit area. Such low-resolution regions are typically accompanied by severe pixelation, blurred contours, and loss of texture details, which severely limit object recognition and relationship detection in critical local regions. Under low-resolution conditions, the perception and discrimination capabilities of traditional visual relationship detection methods are substantially reduced, making it challenging to accurately identify subtle visual relationships that are hard to discern in images. The results in incomplete entity sets $\mathcal{E}$, and relationship sets $\mathcal{R}$, ultimately affecting the semantic completeness of the image and the system's interpretative capacity. Therefore, how to perform robust and comprehensive visual relationship detection under the condition of limited image information, and achieve a structurally complete and semantically rich representation of image content, has become a critical issue that urgently needs to be addressed.

Fig. 1 Examples of missing visual relationships related to low-resolution regions in an image.

This paper studies subtle visual relationship detection under low-resolution conditions to improve the perception of relationships associated with degraded regions in complex images, thereby enabling more comprehensive and precise semantic parsing of images and enhancing the ability of computer vision systems to interpret complex scenes. Recovering subtle visual relationships in low-resolution regions not only enhances the structural integrity and semantic expressiveness of image representation but also holds significant value in various practical applications. On the one hand, in many real-world scenarios (*e.g.*, urban surveillance, transportation hubs, unmanned operations), images inevitably contain distant or blurry regions. Effectively identifying key relationships associated with these regions can improve the stability and robustness of vision systems operating in complex environments. On the other hand, for the

intelligent management and structured analysis of large-scale image data, the completion of semantic information in low-resolution regions also helps to build a more complete index structure, which in turn enhances semantic retrieval and reasoning capabilities. Furthermore, subtle visual relationship detection can be directly applied to the analysis of low-resolution images. In resource-constrained environments (*e.g.*, mobile devices, edge computing terminals), low-resolution images offer advantages such as smaller data volumes and lower transmission costs. If efficient relationship modeling can be achieved under limited pixel conditions, it will significantly enhance the system's real-time processing capacity and deployment efficiency.

Research on small object detection has been relatively well explored [3–5]. However, subtle visual relationship detection extends beyond this by requiring analysis of relationships between entities. This introduces higher challenges in both perception and reasoning compared to merely localizing and classifying individual entities. As shown in Fig. 1, the cars and bicycles in the background exhibit extremely low resolution, and general visual relationship detection methods tend to overlook the relationships involving these entities. Therefore, subtle visual relationship detection faces the following primary difficulties and challenges: (1) Object blurring and detail loss: In low-resolution images, object boundaries are blurred, and details are missing, leading to a significant decrease in the accuracy of entity detection and relationship recognition. (2) Difficulty in feature extraction: Visual features in low-resolution images are sparse and insignificant, and traditional feature extraction methods have difficulty capturing effective semantic information. (3) Complexity in relationship learning: Under low-resolution conditions, spatial relationships and semantic associations between entities become more difficult to learn.

Recent visual relationship detection methods are generally trained and optimized on high-resolution images, often neglecting the specific characteristics of low-resolution scenarios. The principal limitations of these methods are as follows: (1) Inadaptability of feature extraction mechanisms: The feature extraction networks of existing models are designed for high-resolution images rich in detail. As a result, they operate inefficiently when applied to low-resolution images with sparse and low-quality features, making it difficult to obtain effective discriminative representations. (2) Reliance on high-quality input for relationship learning: Current relationship learning methods heavily depend on accurate object detection boxes and clear appearance features. Under low-resolution conditions, blurred object boundaries and sparse details directly cause detection errors and degraded feature quality, thereby undermining the foundation of relationship reasoning. (3) Lack of effective utilization of low-level information: Existing methods mainly focus on entity-level features while neglecting low-level visual cues at the pixel or region level, which can provide crucial complementary information under low-resolution conditions. For example, the detection results of the VCTree model [6] on the VG dataset downsampled to 1/10 resolution are shown in Fig. 2, which compares the ground-truth annotations of the original images (first row) with the VCTree visual relationship detection outputs (second row). In Fig. 2(a), the object "bus" is clearly identifiable in the original image, but when downsampled to a low resolution, it is misclassified as "train" due to its long vehicle body. In Fig. 2(b), since "sign" and "building" occupy relatively large regions in the image, the correct entity

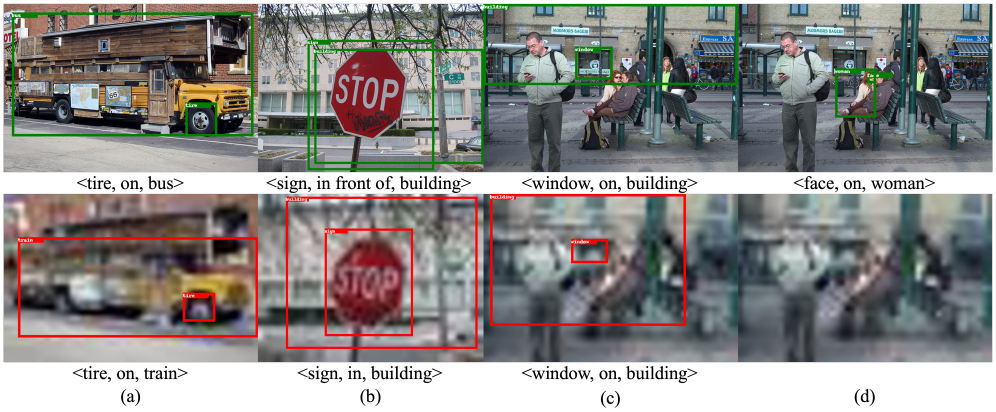

Fig. 2 Comparison of ground-truth annotations on original-resolution images and VCTree detections on low-resolution images. (a)-(d) illustrate four examples: <tire, on, bus>, <sign, in front of, building>, <window, on, building>, and <face, on, woman>.

categories can still be detected after downsampling. However, the spatial relationship between the two is difficult to determine due to the low resolution. In Fig. 2(c), the low-resolution image enables a rough understanding of the overall scene, so the visual relationship "<window, on, building>" can be detected. Nevertheless, the blurred details cause deviations in bounding box predictions. For image details, such as the "woman" and "face" in Fig. 2(d), the model fails to recognize their relation triplet. Therefore, the characteristics of low-resolution scenarios introduce unique challenges, typically requiring models to perform reasoning based on only limited visual cues.

In this paper, we propose a Segmentation-Guided Transformer Network (SGTN). Its core concept is to explicitly incorporate segmentation information, utilizing pixel-level low-level semantics (*e.g.*, entity masks) to compensate for detail loss caused by low resolution, and to jointly predict entities and relationships using a single-stage Transformer architecture. SGTN is an end-to-end framework comprising two-stream feature extraction, multimodal fusion, and relationship prediction. The two-stream feature extraction module employs a visual encoder and a text encoder to extract image visual features and textual semantic features, respectively. The multimodal feature fusion module adopts an encoder-decoder structure to deeply integrate visual and textual features, generating enhanced entity features, relationship features, and mask features. The entity-relation prediction module predicts entity locations, categories, and fine-grained masks from the fused entity features, and predicts relationship triplets between entities by combining entity features and relationship features. We construct downsampled versions of the widely used visual relationship detection dataset *VG* to create evaluation benchmarks *VG_2*, *VG_5*, and *VG_10* for low-resolution scenarios. SGTN is comprehensively evaluated on these benchmarks, demonstrating superior visual relationship detection performance compared to existing methods in low-resolution scenarios. Additionally, an evaluation benchmark *VG_SM* is constructed for visual relationship detection relevant to low-resolution regions. Experimental results show

that the SGTN method achieves excellent performance in detecting visual relationships relevant to low-resolution regions.

The main contributions of this paper can be summarized as follows:

- We study subtle visual relationship detection under low-resolution conditions and build systematic evaluation settings by constructing downsampled *VG* benchmarks as well as a dedicated low-resolution-region benchmark *VG_SM*.
- We propose a segmentation-guided single-stage Transformer that explicitly injects pixel-level mask semantics and performs multimodal fusion for joint prediction of entities and relation predicates under severe resolution degradation.
- Extensive experiments show that SGTN consistently improves robustness to low resolution and recovers missed relationships in degraded regions across all benchmarks.

2 Related Work

Most existing visual relationship detection methods do not address low-resolution scenarios. Nevertheless, developments in small object detection, instance segmentation, and Transformer-based visual relationship detection methods provide theoretical and methodological foundations for subtle visual relationship detection. Some studies in object detection have focused on detecting small objects. Although these methods primarily target the localization and recognition of entities and are still limited in capturing semantic relationships between entities, their effectiveness in low-resolution object recognition establishes a basis for subsequent relationship modeling. Instance segmentation, which extracts semantic information at the pixel level, generates low-level features that serve as effective complementary information for subtle visual relationship detection. Conventional visual relationship detection methods exhibit significant limitations in detecting relationships relevant to low-resolution regions. In contrast, Transformer-based methods, due to their global learning capacity and end-to-end reasoning, are better suited for jointly inferring entities and their relationships in low-resolution scenarios.

2.1 Small Object Detection

In object detection tasks, detecting small and large objects is considered a multi-scale object detection problem, where small objects are often of low resolution. Features extracted from the backbone network display distinct characteristics. Low-level features capture fine-grained details but often contain noise and lack a comprehensive semantic representation. In contrast, high-level features offer richer semantic information but lose detail due to substantially reduced resolution. A key approach to improving detection performance is cross-scale feature fusion. One classical method is the image pyramid [7], which resizes the input image to multiple scales and trains a dedicated detector for each scale. To improve accuracy, Singh et al. [8] proposed SNIP, which performs selective backpropagation based on the different object sizes in each detector. SNIPER [9] further improves the efficiency of SNIP by processing only the background regions around each object rather than every pixel in the image

pyramid, thereby reducing training time. Another line of methods is feature pyramids. FPN [10] connects inherent multi-scale features in convolutional layers via lateral connections and integrates them using a top-down architecture. Subsequently, PANet [11] and BiFPN [12] improve the feature information flow in FPN by using shorter paths. SAN [13] maps multi-scale features into a scale-invariant subspace, enabling the detector to be more robust to scale variations. Context can also benefit low-resolution object detection. The context-attentive CNN [14] enhances object detection performance by leveraging both global and local contextual information. Specifically, it extracts features from multiple subregions of object proposals and integrates them to encode local context. From a data perspective, detection performance can be enhanced by sampling images with small objects and employing segmentation masks to copy and paste these objects, thereby increasing the representation of small objects in the training set [15]. Bai et al. [16], based on generative adversarial networks, employed a super-resolution network as a generator to upsample blurry small images into refined and clear ones. In the discriminator, each super-resolved image patch is described by real/fake scores, object class scores, and regression offsets. During training, the bounding box regression and classification losses in the discriminator are backpropagated to the generator, enabling it to recover more details for more accurate detection.

Small object detection methods aim to improve entity detection in low-resolution conditions. However, these methods primarily focus on the localization and classification of entities and lack the capacity to further analyze the semantic relationships between entities. Nevertheless, the strategy of integrating low-level features and contextual information, commonly used in small object detection, provides valuable insights for subtle visual relationship detection and may facilitate effective detection of entity relationships in low-resolution scenarios.

2.2 Instance segmentation

Segmentation of visual concepts has long been a crucial problem in visual content understanding [17–19], and instance segmentation constitutes a key subtask of image segmentation. Semantic segmentation aims to assign a class label to every pixel in an image, but without distinguishing between different instances of the same class. Typical methods adopt fully connected layers after a series of convolutions [20], and others employ fully convolutional networks [17] to improve performance. Instance segmentation groups pixels belonging to the same semantic class into distinct object instances [21–23], often serving as a segmentation extension to object detection. Mask R-CNN [21] is an instance segmentation framework based on Faster R-CNN. By adding a parallel mask prediction branch, it generates pixel-level segmentation masks for each instance alongside object detection, achieving efficient and accurate segmentation. Bolya et al. [22] proposed a real-time instance segmentation method. It decouples mask generation from object detection, employs parallel branches to predict mask coefficients and prototype masks, and finally combines them linearly to generate instance segmentation results, achieving both efficiency and real-time performance. DETR [24], based on the Transformer architecture, formulates object detection as a direct set prediction problem and can be easily extended to generate panoptic segmentation results in a unified manner. Mask DINO [23] is a unified Transformer-based framework that

integrates object detection and segmentation into a single end-to-end network, and further leverages the DETR architecture and dynamic mask prediction to achieve efficient joint optimization of detection and segmentation.

Although existing instance segmentation methods are not specifically designed for low-resolution images, the segmentation task itself performs pixel-level content understanding. This capability allows these methods to provide low-level semantic information that effectively supports visual relationship detection in low-resolution conditions.

2.3 Transformer-based Visual Relationship Detection

Transformer [25] has demonstrated outstanding performance in the field of natural language processing. Since the introduction of the Vision Transformer (ViT) [26] to computer vision, Transformer models have achieved remarkable results across a wide range of visual tasks. Following the great success of the Transformer-based single-stage object detector DETR [24], research on single-stage visual relationship detection has emerged, building upon single-stage object detectors. These approaches introduce object queries or triplet queries to efficiently learn visual relationships, often employing hybrid CNN-Transformer architectures to directly detect visual relationships from image features. RelTR [27] introduces paired subject and object queries, while SGTR [28] introduces combined queries that are decoupled into subject, object, and predicate. Khandelwal et al. [29] introduced subject, object, and predicate queries, and employed independent multi-layer Transformer decoders for subject, object, and predicate to learn the conditional dependencies between them. SSR-CNN [30] designs a triplet query consisting of subject box, object box, subject, object, and predicate queries. Relationformer [31] does not require separate triplet queries or triplet detectors. It concatenates the final hidden representations of object query pairs with relation tokens and applies a shallow fully connected network for relationship prediction. OvS-GTR [32] is an end-to-end Transformer architecture that can learn visual-concept alignment for identifying nodes and edges.

End-to-end Transformer networks can simultaneously predict entities and relationships, overcoming the limitations of traditional two-stage entity detection methods, and exhibit improved performance in low-resolution scenarios. However, existing methods do not consider low-resolution scenarios, and the features extracted from images remain insufficient for accurate visual relationship detection. We propose the Segmentation-Guided Transformer Network, which incorporates pixel-level features to enhance visual relationship detection performance under low-resolution conditions.

3 Method

In this paper, a Segmentation-Guided Transformer Network (SGTN) is proposed for visual relationship detection under low-resolution conditions. The core concept of SGTN is to address the challenges of visual relationship detection in low-resolution scenarios by utilizing pixel-level low-level information and synchronously predicting entities and relationships through a single-stage Transformer architecture. As shown

in Fig. 3, SGTN comprises a two-stream feature extraction module, a multimodal feature fusion module, and an entity-relation prediction module. The two-stream feature extraction module employs a text encoder and a visual encoder to extract textual and visual features, respectively. The multimodal feature fusion module, consisting of an encoder and a decoder, integrates textual and visual features to generate mask features, entity features, and relation features. The entity-relation prediction module determines the location, category, and mask of entities based on entity features, and infers relation categories by combining entity features with relation features. The external SAM model is used only to generate pseudo mask annotations for mask supervision during training. During inference, SGTN does not rely on SAM or any external mask generator. Instead, it predicts masks and extracts mask features through its own segmentation-relation branch.

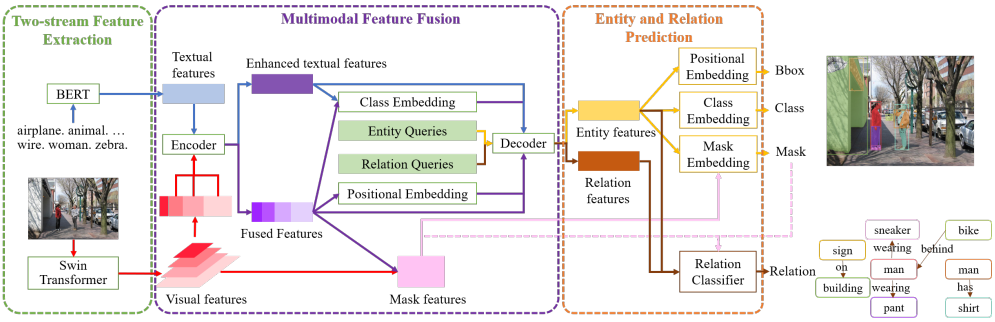

Fig. 3 Pipeline of the Segmentation-Guided Transformer Network.

3.1 Two-Stream Feature Extraction

To address low-resolution scenarios, SGTN utilizes Swin Transformer [33] as the visual encoder to extract multi-scale visual features $\mathbf{F}$, where $F_j \in \mathbf{F}$ denotes the feature of the j-th layer. Swin Transformer is an improved variant of the well-known ViT model and has demonstrated strong performance in multiple computer vision applications, such as semantic segmentation and image classification. Swin Transformer employs a hierarchical architecture that decomposes input images into smaller patches at lower resolutions, and progressively aggregates information from larger spatial regions at higher resolutions. Additionally, it utilizes a sliding window mechanism to efficiently process overlapping image patches, which significantly improves performance.

SGTN adopts BERT [34] as the text encoder to extract textual features T. BERT is a pre-trained language model based on the Transformer architecture that captures deep semantic information in text through bidirectional contextual encoding. The core of BERT is the bidirectional attention mechanism, which enables simultaneous consideration of contextual information and substantially enhances the model's language understanding capability. To enhance the learning of semantic features for visual relationships, SGTN leverages BERT to encode entity categories and aligns visual and

textual features, thereby capturing the semantics of visual features more effectively. Textual prompts for BERT are constructed by concatenating all entity category words, for example, as "airplane. animal. ... woman. zebra." This prompt construction strategy is similar to that used in GLIP [35] and Grounding DINO [36], which generate input prompts by concatenating all noun phrases.

3.2 Multimodal Feature Fusion

The multimodal feature fusion module employs an encoder-decoder architecture to integrate visual and textual features and subsequently generate entity features and relation features. Meanwhile, to provide segmentation guidance for visual relationship detection, SGTN combines the original low-level visual features with the textual-fused low-level features to generate mask features. This design is motivated by the complementary roles of visual details and textual semantic guidance in mask prediction. These output features are then utilized in the entity-relation prediction module to predict entity locations, categories, masks, and relation categories of entity pairs.

The encoder adopts a multi-layer deformable encoder architecture, integrating and enhancing the flattened and concatenated visual features $\bar{F}$ and textual features T via a cross-attention mechanism. In each encoding layer, textual and visual features are initially fused using bidirectional multi-head attention. The initial textual features T'_0 and visual features F'_0 are initialized with T and $\bar{F}$. In the j-th encoding layer, the attention from visual features to textual features A_j^{vl} and from textual features to visual features A_j^{lv} are computed as follows:

$$A_j^{vl} = \text{softmax}\left(\frac{\phi(F'_{j-1})\,\varphi(T'_{j-1})^\top}{\sqrt{d}}\right), \tag{1}$$

$$A_j^{lv} = \text{softmax}\left(\frac{\varphi(T'_{j-1})\,\phi(F'_{j-1})^\top}{\sqrt{d}}\right), \tag{2}$$

where ϕ and φ represent linear transformation operations applied to visual and textual features, respectively, and d is the feature dimension. The fused visual features F'_j are computed by multiplying the attention of visual features to textual features A_j^{vl} by the transformed textual features, followed by a linear transformation, and then adding the result to the visual features from the previous layer. Similarly, the fused textual features T'_j are computed by multiplying the attention of textual features to visual features A_j^{lv} by the transformed visual features, followed by a linear transformation, and then adding the result to the textual features from the previous layer:

$$F'_j = \varphi^\circ(A_j^{vl} \times \varphi'(T'_{j-1})) + F'_{j-1}, \tag{3}$$

$$T'_j = \phi^\circ(A_j^{lv} \times \phi'(F'_{j-1})) + T'_{j-1}, \tag{4}$$

where ϕ', φ', ϕ°, and φ° denote linear transformation operations applied to visual and textual features, respectively. The fused visual features are further enhanced

through multi-scale deformable self-attention encoding to produce F_j°. The fused textual features are enhanced through self-attention encoding to yield T_j°. The fused and enhanced visual features from the final encoder layer are denoted as $\tilde{F}$, and the corresponding textual features are denoted as $\tilde{T}$.

To decode entity and relation features, SGTN constructs entity proposals ρ at multiple scales of $\mathbf{F}$, following the approach in [37]. For the fused and enhanced visual features $\tilde{F}$ and textual features $\tilde{T}$ output by the encoder, SGTN first generates the initial category scores s'^c by calculating their similarity.

$$s'^c = \eta_v(\tilde{F})\,\eta_t(\tilde{T})^\top, \tag{5}$$

where $\eta_v(\cdot)$ and $\eta_t(\cdot)$ project visual and textual features into the same embedding space. The initial location bounding boxes b' are generated using the enhanced visual features $\tilde{F}$ and the entity proposals ρ as follows:

$$b' = \varphi(\tilde{F}) + \rho, \tag{6}$$

where $\varphi(\cdot)$ denotes a linear transformation operation. The positions b° corresponding to the top u category scores in b' are selected as the decoder input, where u denotes the number of entity queries in the decoder.

Segmentation feature extraction begins by separating $\tilde{F}$ according to its original layers, flattening them into two dimensions, and selecting the lowest-level feature $\tilde{\mathcal{F}}$. Simultaneously, the lowest-level feature $\mathcal{F}$ is extracted from the original visual feature $\mathbf{F}$. Since mask prediction is sensitive to spatial details, especially for small or downsampled object regions, SGTN uses $\mathcal{F}$ to preserve localization information. However, $\mathcal{F}$ contains limited semantic information, while $\tilde{\mathcal{F}}$ provides textual semantic guidance but may weaken detailed localization after multimodal encoding. Therefore, SGTN adopts a lightweight fusion strategy similar to FPN, which combines spatial details from $\mathcal{F}$ with semantic guidance from $\tilde{\mathcal{F}}$:

$$F^m = \iota(\delta(\mathcal{F}) + \xi(\tilde{\mathcal{F}})), \tag{7}$$

where ι and δ denote convolution operations, and ξ resamples $\tilde{\mathcal{F}}$ to the same size as $\delta(\mathcal{F})$.

The decoder also receives entity queries q^e and relation queries q^r as input, producing entity features and relation features through a multi-layer deformable decoder architecture. The initial position b_0 is set to b°. The initial entity feature $\widehat{F^e}_0$ and initial relation feature $\widehat{F^r}_0$ are initialized with the entity queries q^e and the relation queries q^r, respectively. At the j-th decoder layer, $\widehat{F^e}_{j-1}$ and $\widehat{F^r}_{j-1}$ are first concatenated and processed with multi-head self-attention to obtain the joint feature $\widehat{F}_{j-1}$. The fused feature is subsequently updated to $\widehat{F}'_{j-1}$ through cross-attention with the encoded textual features $\tilde{T}$. Next, $\widehat{F}'_{j-1}$ is split into the updated entity feature $\widehat{F^e}'_{j-1}$ and relation feature $\widehat{F^r}'_{j-1}$. The entity feature $\widehat{F^e}'_{j-1}$ is updated to $\widehat{F^e}_j$ using deformable cross-attention with the encoded visual features $\tilde{F}$. The relation feature $\widehat{F^r}'_{j-1}$ is updated

to $\widehat{F^r}_j$ using multi-head cross-attention with the encoded visual features $\tilde{F}$. The $\hat{b}_j$ is computed using b° and the updated $\widehat{F^e}_j$ as follows:

$$\hat{b}_j = \varphi(\widehat{F^e}_j) + b^\circ, \tag{8}$$

where $\varphi(\cdot)$ denotes a linear transformation operation. The updated position from the final decoder layer is denoted as the base position $\hat{\flat}$. Entity features from all decoder layers are aggregated as $\widehat{F^e}_j \in \widehat{\mathbf{F^e}}$, while relation features are aggregated as $\widehat{F^r}_j \in \widehat{\mathbf{F^r}}$.

3.3 Entity-Relation Prediction

The entity-relation prediction module leverages the entity features, relation features, and mask features obtained from the multimodal feature fusion module to predict entity categories, positions, segmentation masks, and relation categories.

Entity Category Prediction. Similar to Eq. (5), the entity category score s_j^c at the j-th layer is determined by computing the similarity between the entity features and textual features at that layer:

$$s_j^c = \widehat{F^e}_j \times T, \tag{9}$$

The final entity category score is obtained from the last layer and denoted as $\imath^c$.

Entity Location Prediction. The entity location at the j-th layer is obtained by adding the linearly transformed jth-layer entity feature to the base position $\hat{\flat}$ output from the decoder:

$$b_j = \varphi(\widehat{F^e}_j) + \hat{\flat}, \tag{10}$$

where $\varphi(\cdot)$ denotes a linear transformation, identical to that in Eq. (6). The final entity location is obtained from the updated position at the last layer and denoted as $\flat$.

Entity Segmentation Prediction. The entity masks are computed by multiplying the entity features $\widehat{\mathcal{F}^\rceil}$ output from the last decoder layer by the global mask features F^m:

$$m_i = \sigma\left(\sum_{c=1}^{d_m} K_{i,c}^m F_c^m\right), \tag{11}$$

where $K_i^m = \varpi(\widehat{\mathcal{F}^\rceil}_i)$ denotes the dynamic mask kernel generated by a multilayer perceptron ϖ, F_c^m is the c-th channel of F^m, and $\sigma(\cdot)$ denotes the sigmoid function.

Relation Category Prediction. SGTN first constructs the relation feature for each edge, which consists of six components:

$$\widehat{F^r}_{s,o} = \uplus(\widehat{\mathcal{F}^\rceil}_s, \widehat{\mathcal{F}^\rceil}_o, \widehat{\mathcal{F}^\nabla}, F_s^m, F_o^m, F_{s,o}^m), \tag{12}$$

where $r_{s,o}$ denotes the relation between entity e_s and entity e_o, and $\uplus(\cdot)$ represents the feature concatenation operation. $\widehat{\mathcal{F}^\rceil}_s$ and $\widehat{\mathcal{F}^\rceil}_o$ denote the final-layer features of entity e_s and entity e_o from $\widehat{\mathbf{F^e}}$, respectively, while $\widehat{\mathcal{F}^\nabla}$ denotes the final layer of the

global relation feature $\widehat{\mathbf{Fr}}$. F_s^m and F_o^m denote the mask features of entities e_s and e_o, computed as:

$$F_i^m = \frac{\varrho(m_i \times F^m)}{\varrho(m_i) + \epsilon}, \tag{13}$$

where m_i is the predicted mask of entity e_i, F^m is the global mask features, $\varrho(\cdot)$ denotes summation over the spatial dimensions, and ϵ is a small constant for numerical stability. $F_{s,o}^m$ denotes the mask feature of the relation $r_{s,o}$ between entities e_s and e_o, calculated as follows:

$$F_{s,o}^m = \varrho(\frac{m_s + m_o}{2} \times F^m), \tag{14}$$

where m_s and m_o are the predicted masks of entities e_s and e_o, respectively.

The prediction of predicates between entities is implemented through a relation classifier:

$$s_{s,o}^p = \sigma(\psi(\widehat{Fr}_{s,o})), \tag{15}$$

where $\psi(\cdot)$ denotes a linear transformation operation, and σ is the sigmoid function. The final relation triplet score is computed by multiplying the predicate prediction score and the category prediction scores of the two entities:

$$s_{s,o} = s_{s,o}^p \times s_s^c \times s_o^c. \tag{16}$$

3.4 Loss Function

The loss function comprises four components: entity category loss, entity location loss, entity mask loss, and relation loss.

Entity Category Loss. The entity category loss $\mathcal{L}^c$ is computed using Focal Loss [38]:

$$\begin{aligned}\mathcal{L}^c = \frac{1}{|P^e| + |N^e|} \sum_{e_i \in P^e \cup N^e} \Big[& -\alpha(1 - s_i^c)^\gamma \boldsymbol{c}_i^* \log s_i^c \\ & - (1-\alpha)(s_i^c)^\gamma (1 - \boldsymbol{c}_i^*) \log(1 - s_i^c) \Big],\end{aligned} \tag{17}$$

where P^e and N^e denote the number of positive and negative entity samples, respectively, and $\boldsymbol{c}_i^*$ is the one-hot encoded vector of the true label of entity e_i. $\alpha = 0.25$ is the class-balancing factor and $\gamma = 2.0$ is the focusing parameter.

Entity Location Loss. The location loss includes the L1 loss $\mathcal{L}^b$ and the GIoU loss [39] $\mathcal{L}^g$ for bounding boxes:

$$\mathcal{L}^b = \frac{1}{|P^e|} \sum_{e_i \in P^e} \flat_i - b_i^*, \tag{18}$$

$$\mathcal{L}^g = \frac{1}{|P^e|} \sum_{e_i \in P^e} 1 - \lambda(\flat_i, b_i^*), \tag{19}$$

where b_i^* denotes the ground-truth bounding box of entity e_i, and λ represents the IoU calculation operation.

Entity Mask Loss. To reduce the cost of segmentation annotation, the SGTN method distills the segmentation from the Segment Anything Model (SAM) during training. For a given entity e_i, SAM is used to generate its segmentation result m_i^* within the bounding box b_i^*. The mask loss comprises Focal Loss [38] $\mathcal{L}^{m_f}$ and DICE Loss [40] $\mathcal{L}^{m_d}$:

$$\mathcal{L}^{m_f} = \frac{1}{|P^e|} \sum_{e_i \in P^e} \left[-\alpha(1-m_i)^\gamma m_i^* \log m_i - (1-\alpha) m_i^\gamma (1-m_i^*) \log(1-m_i) \right], \quad (20)$$

$$\mathcal{L}^{m_d} = \frac{1}{|P^e|} \sum_{e_i \in P^e} \left(1 - \frac{2 \times \zeta(m_i \times m_i^*) + 1}{\zeta m_i + \zeta m_i^* + 1} \right), \quad (21)$$

where m_i^* denotes the true mask segmentation for entity e_i, and ζ indicates the summation over the mask matrix.

Relation Loss. The relation category loss $\mathcal{L}^r$ is also computed using Focal Loss:

$$\begin{aligned} \mathcal{L}^r = \frac{1}{|P^r| + |N^r|} \sum_{r_{s,o} \in P^r \cup N^r} \Big[& -\alpha(1-s_{s,o}^r)^\gamma p_{s,o}^* \log s_{s,o}^r \\ & - (1-\alpha)(s_{s,o}^r)^\gamma (1-p_{s,o}^*) \log(1-s_{s,o}^r) \Big], \end{aligned} \quad (22)$$

where P^r and N^r denote the numbers of positive and negative relation samples, respectively, and $p_{s,o}^*$ is the one-hot encoded vector of the ground-truth label of the true relation $r_{s,o}$.

The final loss $\mathcal{L}_{sum}$ is defined as the sum of the four types of losses. In addition to the loss $\mathcal{L}$ computed on the final predictions, the denoising loss $\mathcal{L}_{dn}$ [37] for entity categories and locations, as well as the intermediate outputs loss $\mathcal{L}_{int}$ and the auxiliary loss $\mathcal{L}_{aux}$ for entity categories, locations, and relations, are also included:

$$\mathcal{L}_{sum} = \mathcal{L} + \mathcal{L}_{dn} + \mathcal{L}_{int} + \mathcal{L}_{aux}, \quad (23)$$

$$\mathcal{L} = \mathcal{L}^c + \mathcal{L}^b + \mathcal{L}^g + \mathcal{L}^{m_f} + \mathcal{L}^{m_d} + \mathcal{L}^r, \quad (24)$$

$$\mathcal{L}_{dn} = \mathcal{L}_{dn}^c + \mathcal{L}_{dn}^b + \mathcal{L}_{dn}^g, \quad (25)$$

$$\mathcal{L}_{int} = \mathcal{L}_{int}^c + \mathcal{L}_{int}^b + \mathcal{L}_{int}^g + \mathcal{L}_{int}^r, \quad (26)$$

$$\mathcal{L}_{aux} = \sum_{j \in \{1,2,\ldots,J\}} \mathcal{L}_j{}^c + \mathcal{L}_j{}_{dn}^c + \mathcal{L}_j{}^b + \mathcal{L}_j{}_{dn}^b + \mathcal{L}_j{}^g + \mathcal{L}_j{}_{dn}^g + \mathcal{L}_j{}^r, \quad (27)$$

where the calculation of entity and relation losses in $\mathcal{L}_{dn}$, $\mathcal{L}_{int}$, and $\mathcal{L}_{aux}$ is similar to that of $\mathcal{L}$, and J denotes the number of feature layers output by the encoder.

4 Experiments

The experiments were conducted on a platform featuring an Intel® Xeon(R) CPU E5-2680 v4 processor, four GeForce RTX 3090 GPUs, and 128 GB of memory. The implementation is based on Python, utilizing the PyTorch deep learning framework.

Model training was performed on the training set of the widely used *VG* dataset for visual relationship detection, with a learning rate of 0.00001 and a batch size of 1. Model evaluation was conducted on both the downsampled and filtered versions of the *VG* test set. With Swin-T as the backbone network, the proposed method achieves an average inference time of 0.77 seconds per image on *VG_10*, and improves mRecall@50 by 7.58% on *VG_10* compared to the best existing method. On the *VG_SM* dataset, the approach yields improvements of 26.34% and 33.01% in mRecall@50 for visual relationship detection between small-sized entities and between medium-sized entities, respectively. The calculation of this metric proceeds as follows: the best value achieved by the compared methods on the target metric is first determined. The difference between the score of the proposed method and the best value is then computed, and this difference is divided by the best value of the compared methods. The process provides a normalized measure of the performance variation of the proposed method relative to the optimal level of the compared methods.

4.1 Datasets and Evaluation Metrics

The performance of subtle visual relationship detection is evaluated on the widely used *VG* dataset, with Recall@K and mRecall@K adopted as evaluation metrics. Due to the absence of explicit annotations for relationships relevant to low-resolution regions in existing datasets, the experimental evaluation environment is constructed by downsampling images in *VG* to simulate varying degrees of low-resolution scenarios. The average resolution of the *VG* dataset is 532.45×381.18. The original images are resized to 1/2, 1/5, and 1/10 of their original dimensions, generating three datasets with different resolutions, named *VG_2*, *VG_5*, and *VG_10*, respectively. These datasets are used to validate the robustness of the proposed method under varying resolution levels. In the experiment, the SGTN method was implemented based on the pre-trained GroundingDINO model. To avoid data leakage, images in the training data of this model that overlap with those in the *VG* dataset are excluded during evaluation, resulting in a final test set of 14,700 images. Additionally, following the definition of small and medium entities in the *COCO* dataset, a *VG_SM* dataset is constructed by selecting visual relationships from the *VG* dataset where the subject and/or object bounding boxes are smaller than 32×32 (small) or 96×96 (medium). To some extent, this dataset enables evaluation of the effectiveness of the SGTN method in detecting relationships relevant to low-resolution regions.

4.2 Comparison with Baseline Methods

The proposed SGTN method is compared with existing approaches for visual relationship detection on the VG, VG_2, VG_5, and VG_10 datasets. The results are presented in Table 1 and Table 2, where † indicates that the method does not have official model parameters and instead uses self-trained model parameters. The first seven rows correspond to two-stage methods. Motifs [41, 42] is a classic approach that leverages dataset bias, while TDE-Motifs [41, 42], TDE-VCTree [6, 42], BGNN [43], and GCL [44] are designed for unbiased estimation. SpeaQ [45], SGTR [28], EGTR [46], RelTR [27], and OvSGTR [32] are one-stage methods, and EGTR w. LA is an unbiased estimation

Table 1 Comparison of Recall between the proposed method and existing methods. (Bold numbers indicate the best results. Underlined numbers indicate the second-best results. Recall@K is abbreviated as R@K.)

Method	Backbone	*VG*		*VG_2*		*VG_5*		*VG_10*	
		R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
Motifs [41, 42]	ResNet-101	32.37	37.12	27.59	31.88	12.34	14.83	2.45	3.09
TDE-Motifs [41, 42]	ResNet-101	16.43	20.02	14.04	17.13	5.96	7.51	1.04	1.36
VCTree† [6]	ResNet-101	32.02	36.19	26.73	30.55	10.76	12.80	1.69	2.15
TDE-VCTree† [6, 42]	ResNet-101	30.67	35.34	25.62	29.84	10.83	13.07	1.85	2.34
Transformer† [42]	ResNet-101	30.39	34.75	25.14	29.23	10.25	12.59	1.50	1.97
BGNN [43]	ResNet-101	30.47	35.27	25.99	30.4	12.36	14.79	2.36	3.07
GCL† [44]	ResNet-101	18.73	22.22	15.19	18.55	5.62	7.12	0.85	1.10
SpeaQ [45]	ResNet-101	32.16	35.66	25.61	28.37	8.99	10.17	1.57	1.88
SGTR [28]	ResNet-101	24.52	27.83	19.14	21.77	5.54	6.61	0.42	0.49
EGTR [46]	ResNet-50	30.20	34.35	22.97	26.70	6.85	8.15	1.04	1.24
EGTR w. LA [46]	ResNet-50	18.71	20.48	13.73	15.12	3.22	3.64	0.26	0.31
RelTR [27]	ResNet-50	25.07	27.26	19.51	21.22	4.99	5.48	0.50	0.58
RelTR† [27]	Swin-T	26.70	28.83	21.90	23.66	9.69	10.46	1.99	2.18
OvSGTR [32]	Swin-T	35.83	41.40	31.63	36.88	17.88	21.62	5.71	7.14
OvSGTR [32]	Swin-B	<u>36.66</u>	<u>42.28</u>	<u>34.04</u>	<u>39.66</u>	<u>25.03</u>	<u>29.68</u>	**12.94**	<u>15.85</u>
SGTN	Swin-T	35.57	40.73	31.87	37.04	18.88	22.70	6.31	7.93
SGTN	Swin-B	**37.61**	**43.03**	**35.25**	**40.76**	**25.73**	**30.40**	<u>12.93</u>	**16.00**

method incorporating result adjustment. In addition, a version of RelTR with Swin Transformer as the backbone is provided, and the results indicate that Swin Transformer indeed yields notable improvements for subtle visual relationship detection. Among the three methods using Swin-T as the backbone, SGTN achieves the second-highest Recall on *VG*, and the highest Recall on *VG_2*, *VG_5*, and *VG_10*, where image resolution is reduced. With Swin-B as the backbone, SGTN also achieves the best Recall on *VG*. Beyond the baseline SGTN, SGTN w. LA is introduced, which incorporates result adjustment for unbiased estimation and leads to a significant improvement in the mRecall metric. SGTN w. LA achieves the second-highest mRecall@100 on *VG* and mRecall@50 on *VG_2*, while obtaining the best mRecall across all metrics on *VG_5* and *VG_10*, where image resolution is particularly low. The experimental results demonstrate that the SGTN method is more effective under low-resolution conditions.

High-resolution images often contain small local regions that are effectively low-resolution, where important visual relationships can be easily missed. To evaluate performance in such degraded regions, we benchmark SGTN and OvSGTR, the strongest baseline in our previous experiments, on the *VG_SM* dataset, which is designed for visual relationship detection in low-resolution regions. For a fair comparison, both methods adopt Swin-B as the backbone network. Table 3 presents a comparison of visual relationship detection results between small entities and between medium entities, while Table 4 reports results for relationships between small or medium entities and other entities. Except for mRecall@100 for detecting relationships between small entities and other entities, SGTN outperforms comparison methods across all evaluated metrics. In addition, we provide results of an OvSGTR model trained on *VG_10*. Most metrics are lower than those of the original model, indicating

Table 2 Comparison of mRecall between the proposed method and existing methods. (Bold numbers indicate the best results. Underlined numbers indicate the second-best results. mRecall@K is abbreviated as mR@K.)

Method	Backbone	*VG*		*VG_2*		*VG_5*		*VG_10*	
		mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	mR@50	mR@100
Motifs [41, 42]	ResNet-101	5.82	7.01	4.83	5.83	1.95	2.45	0.37	0.49
TDE-Motifs [41, 42]	ResNet-101	9.18	11.13	7.37	9.22	2.78	3.58	0.39	0.53
VCTree† [6]	ResNet-101	7.19	8.39	5.79	6.90	2.04	2.45	0.30	0.38
TDE-VCTree† [6, 42]	ResNet-101	6.09	7.48	5.00	6.15	1.88	2.32	0.30	0.40
Transformer† [42]	ResNet-101	7.62	9.08	6.30	7.54	2.14	2.70	0.47	0.58
BGNN [43]	ResNet-101	10.26	12.23	8.32	10.15	3.44	4.23	0.81	0.99
GCL† [44]	ResNet-101	13.38	15.13	10.43	11.93	3.32	3.91	0.38	0.57
SpeaQ [45]	ResNet-101	<u>15.01</u>	17.56	11.72	13.43	3.58	4.13	0.68	0.81
SGTR [28]	ResNet-101	13.04	16.45	9.26	12.05	2.54	3.53	0.14	0.19
EGTR [46]	ResNet-50	7.96	10.02	5.89	7.76	1.47	1.95	0.26	0.32
EGTR w. LA [46]	ResNet-50	**17.01**	**21.69**	**12.85**	<u>16.07</u>	3.96	4.83	0.59	0.86
RelTR [27]	ResNet-50	8.43	9.93	5.83	6.72	1.18	1.34	0.01	0.01
RelTR† [27]	Swin-T	9.30	10.71	7.36	8.28	2.91	3.31	0.66	0.70
OvSGTR [32]	Swin-T	7.15	8.72	6.23	7.64	3.09	4.01	0.96	1.23
OvSGTR [32]	Swin-B	7.37	9.01	6.69	8.22	4.61	5.74	2.11	2.71
SGTN	Swin-T	8.51	10.37	7.53	9.36	3.68	4.73	1.11	1.52
SGTN	Swin-B	8.77	10.81	7.54	9.48	5.10	6.37	<u>2.27</u>	2.93
SGTN w. LA	Swin-T	14.48	17.96	<u>12.81</u>	15.96	<u>6.84</u>	<u>8.84</u>	2.09	<u>2.73</u>
SGTN w. LA	Swin-B	14.94	<u>18.50</u>	12.79	**16.27**	**8.45**	**10.68**	**3.70**	**4.97**

Table 3 Comparison of the proposed method and OvSGTR on the *VG_SM* dataset for visual relationship detection, where both subjects and objects are small or medium entities. (Bold numbers indicate the results. Recall@K is abbreviated as R@K, and mRecall@K as mR@K.)

Method	small				medium			
	R@50	R@100	mR@50	mR@100	R@50	R@100	mR@50	mR@100
OvSGTR [32]	13.85	18.58	1.86	2.76	20.45	25.98	4.12	5.45
OvSGTR [32] (VG_10)	11.43	17.96	1.98	2.85	18.19	23.83	3.68	4.94
SGTN	**21.20**	**24.27**	**2.35**	**3.31**	**24.01**	**29.61**	**5.48**	**7.18**

Table 4 Comparison of the proposed method and OvSGTR on the *VG_SM* dataset for visual relationship detection, where either the subject or the object is a small entity or medium entity. (Bold numbers indicate the best results. Recall@K is abbreviated as R@K, and mRecall@K as mR@K.)

Method	small				medium			
	R@50	R@100	mR@50	mR@100	R@50	R@100	mR@50	mR@100
OvSGTR [32]	29.98	36.32	4.58	**7.58**	34.65	40.62	6.16	7.79
OvSGTR [32] (VG_10)	27.04	33.49	4.04	5.14	31.74	37.56	5.72	7.14
SGTN	**31.59**	**37.51**	**5.51**	7.17	**35.84**	**41.54**	**7.58**	**9.69**

that training on low-resolution images does not improve the detection of relationships in low-resolution regions but instead suppresses model effectiveness.

Fig. 4 presents comparison examples of visual relationship detection where both the subject and object are small entities, as well as cases where either the subject or the object is a small entity. If, within the top-100 predictions, there exists a triplet in which the subject, predicate, and object categories together with the subject and

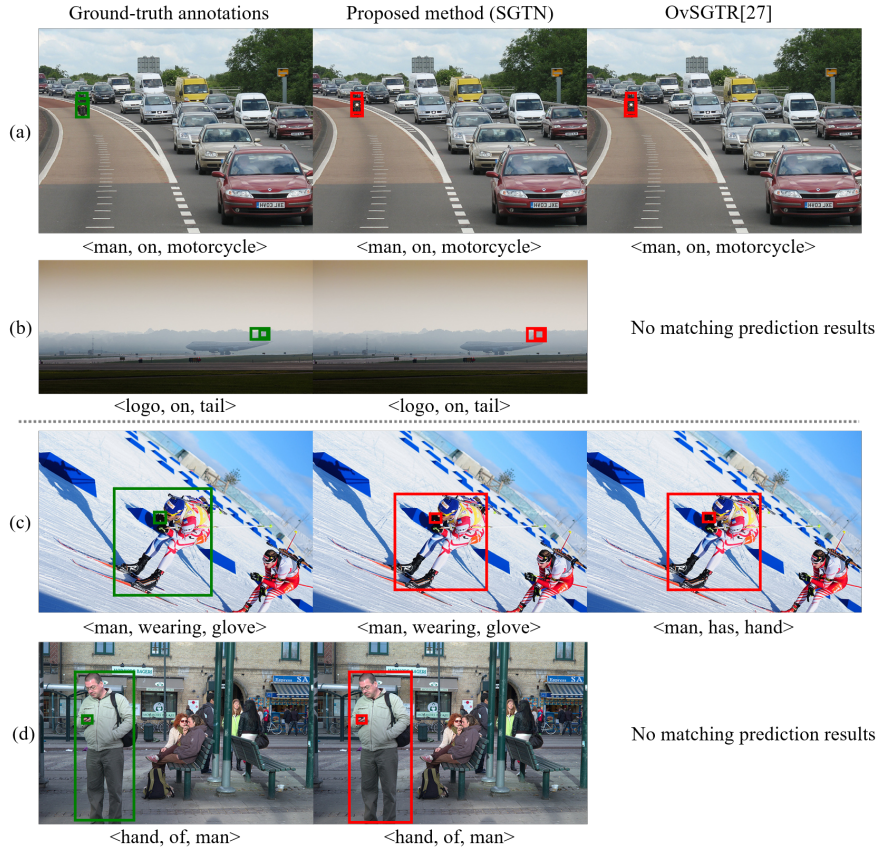

Fig. 4 Examples of ground-truth annotations and predicted results on the *VG_SM* dataset. (a) and (b) show comparison examples of visual relationship detection where both the subject and object are small entities; (c) and (d) show comparison examples where either the subject or the object is a small entity.

object bounding boxes match the ground-truth annotation, that prediction is shown; otherwise, the prediction with matched subject and object bounding boxes is displayed. For relationships between small entities, Fig. 4(a) shows that both SGTN and OvSGTR successfully detect the $\langle \text{man}, \text{on}, \text{motorcycle} \rangle$ relationship in a low-resolution region of a large-scale image. In contrast, Fig. 4(b) demonstrates that SGTN correctly detects the $\langle \text{logo}, \text{on}, \text{tail} \rangle$ relationship in the distant tail region of an airplane, whereas OvSGTR fails to capture it. For relationships between small and other entities, Fig. 4(c) shows that SGTN accurately detects the $\langle \text{man}, \text{wearing}, \text{glove} \rangle$ relationship between the man and the low-resolution hand region, whereas OvSGTR only predicts $\langle \text{man}, \text{has}, \text{hand} \rangle$. This demonstrates that SGTN provides more precise results. Furthermore, as shown in Fig. 4(d), SGTN successfully detects the $\langle \text{hand}, \text{of}, \text{man} \rangle$ relationship between the person and the low-resolution hand region, while OvSGTR fails to detect any corresponding result. The experimental results demonstrate that

the proposed SGTN method for subtle visual relationship detection can effectively complement relationships associated with low-resolution regions of images.

4.3 Ablation Study

To analyze the effectiveness of SGTN, multiple variants are designed, all adopting Swin-T as the backbone network. Table 5 and Table 6 present evaluation results of SGTN with different feature configurations. Table 7 compares the results of SGTN when replacing the embedded mask features with the mask features from the SAM model.

Table 5 Comparison of Recall between the proposed method and its variants. (Bold numbers indicate the best results, underlined numbers indicate the second-best results. Recall@K is abbreviated as R@K.)

Method	*VG*		*VG_2*		*VG_5*		*VG_10*	
	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
OvSGTR [32]	<u>35.83</u>	<u>41.40</u>	31.63	36.88	17.88	21.62	5.71	7.14
SGTN w/o so w/o rln	35.82	**41.63**	<u>32.29</u>	**37.85**	<u>19.13</u>	**23.14**	**6.52**	**8.25**
SGTN w/o rln	**36.00**	41.12	**32.50**	<u>37.46</u>	**19.29**	<u>22.78</u>	<u>6.38</u>	<u>8.02</u>
SGTN w/o rln w. emb	29.84	34.90	27.16	32.13	16.76	20.39	5.88	7.49
SGTN w/o so	35.38	40.69	31.71	36.90	18.72	22.56	6.13	7.78
SGTN w/o fusion	33.90	39.42	28.38	34.09	15.94	19.66	4.71	6.24
SGTN w/o text inter.	35.36	40.70	31.72	37.05	18.67	22.39	6.15	7.74
SGTN w/o pretrained init	7.16	8.98	4.92	6.27	2.10	2.68	1.01	1.22
SGTN	35.57	40.73	31.87	37.04	18.88	22.70	6.31	7.93

Table 6 Comparison of mRecall between the proposed method and its variants. (Bold numbers indicate the best results, underlined numbers indicate the second-best results. mRecall@K is abbreviated as mR@K.)

Method	*VG*		*VG_2*		*VG_5*		*VG_10*	
	mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	mR@50	mR@100
OvSGTR [32]	7.15	8.72	6.23	7.64	3.09	4.01	0.96	1.23
SGTN w/o so w/o rln	7.83	9.35	6.82	8.49	<u>3.58</u>	<u>4.62</u>	1.08	1.42
SGTN w/o rln	7.51	9.06	6.66	8.06	3.39	4.33	<u>1.09</u>	1.42
SGTN w/o rln w. emb	5.33	6.59	4.90	6.06	2.73	3.55	0.92	1.21
SGTN w/o so	7.94	9.76	<u>7.00</u>	<u>8.53</u>	3.48	4.46	1.07	<u>1.43</u>
SGTN w/o fusion	<u>8.09</u>	<u>10.14</u>	6.01	8.07	2.77	3.63	0.82	1.33
SGTN w/o text inter.	8.03	9.99	6.51	8.12	3.31	4.19	1.01	1.34
SGTN w/o pretrained init	1.09	1.47	0.71	0.96	0.30	0.42	0.13	0.18
SGTN	**8.51**	**10.37**	**7.53**	**9.36**	**3.68**	**4.73**	**1.11**	**1.52**

To analyze the contribution of different features to SGTN, we evaluate the effectiveness of five variants with different uses of entity features and relation features. OvSGTR, which does not incorporate entity segmentation or segmentation guidance, serves as the baseline variant for the ablation study. "SGTN w/o so w/o rln" includes

entity segmentation but without segmentation guidance. "SGTN w/o rln" removes the relation mask feature $F^m_{s,o}$. "SGTN w/o rln w. emb" is a variant that removes the relation mask feature and replaces it with subject and object entity mask embeddings. "SGTN w/o so" removes the subject and object mask features F^m_s and F^m_o. To analyze the contribution of different components to the performance improvements of SGTN, we also evaluate three component-level variants. "SGTN w/o fusion" removes the feature fusion layer. "SGTN w/o text inter." removes the text interaction module. "SGTN w/o pretrained init" trains the model without pretrained detector initialization. The evaluation results of these variants are shown in Table 5 and Table 6.

Comparing OvSGTR with "SGTN w/o so w/o rln" reveals that incorporating entity segmentation improves the performance of subtle visual relationship detection. From the comparison between "SGTN w/o so w/o rln" and the full SGTN in Table 6, we observe that segmentation guidance helps improve mRecall, indicating that it enables more unbiased estimation. By comparing the results of OvSGTR, "SGTN w/o so w/o rln", "SGTN w/o rln", and "SGTN w/o so" in Table 6, we find that at higher resolutions (*VG* and *VG_2*), segmentation guidance on relations is more effective than segmentation guidance on subject-object entities for improving mRecall, whereas at lower resolutions (*VG_5* and *VG_10*), both are required to achieve improvement. The comparison between "SGTN w/o rln" and "SGTN w/o rln w. emb" demonstrates that segmentation guidance is essential for effectiveness. Directly using segmentation embeddings cannot improve performance and may even have a negative effect.

The results of "SGTN w/o fusion" show that removing the feature fusion layer leads to lower results than SGTN on all evaluated splits and all Recall/mRecall metrics, indicating that multimodal feature fusion provides stable benefits for relation prediction under different degrees of resolution degradation. The comparison between "SGTN w/o text inter." and SGTN shows that removing textual interaction mainly affects mRecall, while its influence on standard Recall is relatively smaller. This suggests that textual semantic interaction is more helpful for balanced relation recognition, rather than simply increasing the recall of frequent relation categories. Moreover, "SGTN w/o pretrained init" shows the largest performance drop among the three component-level variants, demonstrating that pretrained visual detection knowledge is important for stable entity localization and relation reasoning. In summary, pretrained detector initialization contributes the most to the overall performance, multimodal feature fusion provides stable improvements across both Recall and mRecall, and textual interaction mainly contributes to more balanced relation recognition. Overall, these ablation experiments show that the performance gains of SGTN are derived from the joint contributions of segmentation guidance, multimodal feature fusion, textual semantic interaction, and pretrained visual detection initialization.

We compare SGTN with a diagnostic variant, "SGTN w/o rln w. SAM", which uses mask features generated by the SAM model as the subject and object mask features in "SGTN w/o rln". This variant intentionally introduces external SAM masks for analysis only, whereas the proposed SGTN uses its own predicted masks at inference. The mRecall results are reported in Table 7. Without applying result adjustment for unbiased estimation, SGTN achieves higher mRecall, indicating that it is more effective at analyzing relational semantics on its own. With result adjustment applied, the

Table 7 Comparison of mRecall between the proposed method and its variants using SAM mask features. (Bold numbers indicate the best results. mRecall@K is abbreviated as mR@K.)

Method	*VG*		*VG_2*		*VG_5*		*VG_10*	
	mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	mR@50	mR@100
SGTN w/o rln	7.51	9.06	6.66	8.06	3.39	4.33	1.09	1.42
SGTN w/o rln w. SAM	8.32	10.22	7.27	8.88	3.65	**4.75**	1.06	1.44
SGTN	**8.51**	**10.37**	**7.53**	**9.36**	**3.68**	4.73	**1.11**	**1.52**
SGTN w/o rln w. LA	13.48	16.85	11.71	15.42	6.23	7.89	1.70	2.28
SGTN w/o rln w. SAM w. LA	**14.72**	**18.43**	**12.85**	**16.14**	6.64	8.77	1.63	2.21
SGTN w. LA	14.48	17.96	12.81	15.96	**6.84**	**8.84**	**2.09**	**2.73**

variant using SAM-generated mask features achieves higher mRecall in high-resolution settings, whereas SGTN outperforms in low-resolution scenarios. This demonstrates that SGTN is more effective in adapting to low-resolution conditions.

4.4 Efficiency Analysis

To further evaluate the computational cost introduced by the proposed modules, we add a detailed efficiency comparison between SGTN and the strongest one-stage baseline OvSGTR under a unified setting. All efficiency results are measured on the same single-GPU platform with a batch size of 1 and the same *VG* evaluation protocol. FLOPs are computed for the forward pass of a single image, while the reported forward latency is averaged over repeated runs after warm-up. We also report end-to-end pipeline latency including post-processing, and peak GPU memory is measured during inference.

Table 8 Efficiency comparison between SGTN and the strongest baseline on the *VG* dataset. Forward latency denotes model forward time, while pipeline latency includes post-processing.

Method	Backbone	Train. Params/M	Total Params/M	FLOPs/G	Forward/ms	Pipeline/ms	Memory/GB	R@50
OvSGTR [32]	Swin-B	41.91	238.27	92.04	226.31	244.15	2.98	36.66
SGTN	Swin-B	44.37	240.73	149.08	239.27	700.30	13.23	37.61

As shown in Table 8, compared with OvSGTR, the full SGTN model increases the number of trainable parameters by 2.46M and the total number of parameters by 2.46M. The forward latency increases from 226.31 ms to 239.27 ms, indicating that the additional segmentation-guided modules introduce only a moderate overhead in the main network forward pass. The end-to-end pipeline latency is higher because the full model further introduces relation-aware inference and post-processing.

5 Conclusion

In this paper, we introduced the task of subtle visual relationship detection, aiming to complement visual relationships that are less perceptible in images. A Segmentation-Guided Transformer Network is proposed, which explicitly incorporates segmentation information and leverages pixel-level low-level semantics to compensate for detail loss

caused by low resolution. The network employs a single-stage Transformer architecture with global learning and end-to-end inference capabilities to simultaneously predict entities and their relationships. The proposed method extracts aligned visual and textual features using pretrained vision and textual encoders, then fuses and enhances these features through an encoder-decoder structure and jointly predicts the locations, categories, masks, and relational predicates of subject-object pairs. Experiments are conducted on the *VG* dataset and its downsampled low-resolution variants. Results demonstrate that the proposed method better preserves performance under reduced resolution, achieving a 7.58% improvement in mRecall@50 over the best existing method on *VG_10*. Additionally, we construct a benchmark dataset *VG_SM* for evaluating visual relationship detection in low-resolution regions. Compared to baseline methods, the proposed approach improves mRecall@50 by 26.34% and 33.01% for relationships between small entities and between medium entities, respectively, demonstrating its superior capability in complementing visual relationships relevant to low-resolution regions.

Author contribution

Fan Yu: Conceptualization, Methodology, Software, Writing - original draft. Hanxi Cao: Investigation, Methodology, Validation. Ruichao Hou: Project administration, Writing - original draft. Tongwei Ren: Polishing, Funding acquisition, Resource.

Funding

This work was supported by the National Natural Science Foundation of China (62072232), the Key R&D Project of Jiangsu Province (BE2022138), the Fundamental Research Funds for the Central Universities (021714380026), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

Data availability

All datasets used are publicly available.

Conflict of interest

The authors declare no Conflict of interest.

References

[1] Cheng, J., Wang, L., Wu, J., Hu, X., Jeon, G., Tao, D., Zhou, M.: Visual relationship detection: A survey. IEEE Transactions on Cybernetics **52**(8), 8453–8466 (2022)

[2] Liu, T., An, G., Yang, Z., Ren, X., Ruan, Q.: Eira: an explicit-implicit representation alignment for multimodal relation extraction. Multimedia Systems **31**(6), 472 (2025)

[3] Lei, H., Wu, Z., Shang, L., Zhao, H., Yang, W.: Tcf-detr: multi-scale token-channel fusion transformer for enhanced small object detection. Multimedia Systems **31**(4), 1–15 (2025)

[4] Shi, Z., Hu, J., Ren, J., Ye, H., Yuan, X., Ouyang, Y., He, J., Ji, B., Guo, J.: Hs-fpn: High frequency and spatial perception fpn for tiny object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 6896–6904 (2025)

[5] Rekavandi, A.M., Xu, L., Boussaid, F., Seghouane, A.-K., Hoefs, S., Bennamoun, M.: A guide to image-and video-based small object detection using deep learning: case study of maritime surveillance. IEEE Transactions on Intelligent Transportation Systems (2025)

[6] Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6619–6628 (2019)

[7] Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA Engineer **29**(6), 33–41 (1984)

[8] Singh, B., Davis, L.S.: An analysis of scale invariance in object detection snip. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3578–3587 (2018)

[9] Singh, B., Najibi, M., Davis, L.S.: Sniper: Efficient multi-scale training. In: Advances in Neural Information Processing Systems, vol. 31 (2018)

[10] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)

[11] Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)

[12] Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 10781–10790 (2020)

[13] Kim, Y., Kang, B.-N., Kim, D.: San: Learning relationship between convolutional features for multi-scale object detection. In: European Conference on Computer Vision, pp. 316–331 (2018)

[14] Li, J., Wei, Y., Liang, X., Dong, J., Xu, T., Feng, J., Yan, S.: Attentive contexts for object detection. IEEE Transactions on Multimedia **19**(5), 944–954 (2016)

[15] Kisantal, M., Wojna, Z., Murawski, J., Naruniec, J., Cho, K.: Augmentation for small object detection. In: International Conference on Advances in Computing and Information Technology (2019)

[16] Bai, Y., Zhang, Y., Ding, M., Ghanem, B.: Sod-mtgan: Small object detection via multi-task generative adversarial network. In: European Conference on Computer Vision, pp. 206–221 (2018)

[17] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)

[18] Yang, L., Song, Q., Wang, Z., Jiang, M.: Parsing r-cnn for instance-level human analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 364–373 (2019)

[19] Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 9404–9413 (2019)

[20] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K.P., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets and atrous convolution and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 843–848 (2018)

[21] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)

[22] Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact: Real-time instance segmentation. In: IEEE International Conference on Computer Vision, pp. 9157–9166 (2019)

[23] Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L.M., Shum, H.-Y.: Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3041–3050 (2023)

[24] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229 (2020)

[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008 (2017)

[26] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., *et al.*: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)

[27] Cong, Y., Yang, M.Y., Rosenhahn, B.: Reltr: Relation transformer for scene graph generation. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(9), 11169–11183 (2023)

[28] Li, R., Zhang, S., He, X.: Sgtr: End-to-end scene graph generation with transformer. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 19486–19496 (2022)

[29] Khandelwal, S., Sigal, L.: Iterative scene graph generation. In: Advances in Neural Information Processing Systems, vol. 35, pp. 24295–24308 (2022)

[30] Teng, Y., Wang, L.: Structured sparse r-cnn for direct scene graph generation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 19437–19446 (2022)

[31] Shit, S., Koner, R., Wittmann, B., Paetzold, J., Ezhov, I., Li, H., Pan, J., Sharifzadeh, S., Kaissis, G., Tresp, V., *et al.*: Relationformer: A unified framework for image-to-graph generation. In: European Conference on Computer Vision, pp. 422–439 (2022)

[32] Chen, Z., Wu, J., Lei, Z., Zhang, Z., Chen, C.: Expanding scene graph boundaries: Fully open-vocabulary scene graph generation via visual-concept alignment and retention. In: European Conference on Computer Vision, pp. 108–124 (2024)

[33] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE International Conference on Computer Vision, pp. 10012–10022 (2021)

[34] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 4171–4186 (2019)

[35] Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., *et al.*: Grounded language-image pre-training. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 10965–10975 (2022)

[36] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., *et al.*: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: European Conference on Computer Vision, pp. 38–55 (2024)

[37] Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.-Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In: International Conference on Learning Representations (2022)

[38] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)

[39] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)

[40] Milletari, F., Navab, N., Ahmadi, S.-A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: International Conference on 3D Vision, pp. 565–571 (2016)

[41] Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5831–5840 (2018)

[42] Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H.: Unbiased scene graph generation from biased training. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3716–3725 (2020)

[43] Li, R., Zhang, S., Wan, B., He, X.: Bipartite graph network with adaptive message passing for unbiased scene graph generation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 11109–11119 (2021)

[44] Dong, X., Gan, T., Song, X., Wu, J., Cheng, Y., Nie, L.: Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 19427–19436 (2022)

[45] Kim, J., Park, J., Park, J., Kim, J., Kim, S., Kim, H.J.: Groupwise query specialization and quality-aware multi-assignment for transformer-based visual relationship detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 28160–28169 (2024)

[46] Im, J., Nam, J., Park, N., Lee, H., Park, S.: Egtr: Extracting graph from transformer for scene graph generation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 24229–24238 (2024)