Navigating the Unseen: Zero-shot Scene Graph Generation via Capsule-Based Equivariant Features

Wenhuan Huang, Yi JI, Guiqian Zhu, Li Ying, Chunping Liu*
School of Computer Science and Technology,
Soochow University, Suzhou, China
whhuang@stu.suda.edu.cn {jiyi, gqzhu, ingli, cpliu}@suda.edu.cn

Abstract

In scene graph generation (SGG), the accurate prediction of unseen triplets is essential for its effectiveness in downstream vision-language tasks. We hypothesize that the predicates of unseen triplets can be viewed as transformations of seen predicates in feature space, and the essence of the zero-shot task is to bridge the gap caused by this transformation. Traditional models, however, have difficulty addressing this challenge, which we attribute to their inability to model the predicates equivariant. To overcome this limitation, we introduce a novel framework based on capsule networks (CAPSGG). We propose a ***Three-Stream Pipeline*** *that generates modality-specific representations for predicates, while building low-level predicate capsules of these modalities. Then, these capsules are aggregated into high-level predicate capsules using a* ***Routing Capsule Layer****. In addition, we introduce* ***GroupLoss*** *to aggregate capsules with the same predicate label into groups. This replaces the global loss with the intra-group loss, effectively balancing the learning of predicate invariant and equivariant features while mitigating the impact of the severe long-tail distribution of the predicate categories. Our extensive experiments demonstrate the notable superiority of our approach over state-of-the-art methods, with zero-shot indicators outperforming up to* ***132.26%*** *on the SGCls task than the T-CAR [21]. Our code will be available upon publication.*

1. Introduction

A Scene Graph (SG) is a sophisticated topological mapping of a visual scene, meticulously encoding the interactions and contextual relationships among objects. Scene Graph Generation (SGG) extends beyond mere object detection, which encompasses the inference of object relationships, culminating in constructing a comprehensive scene graph. This technology has become widely applied across various domains, such as image captioning, visual question answering, and image generation.

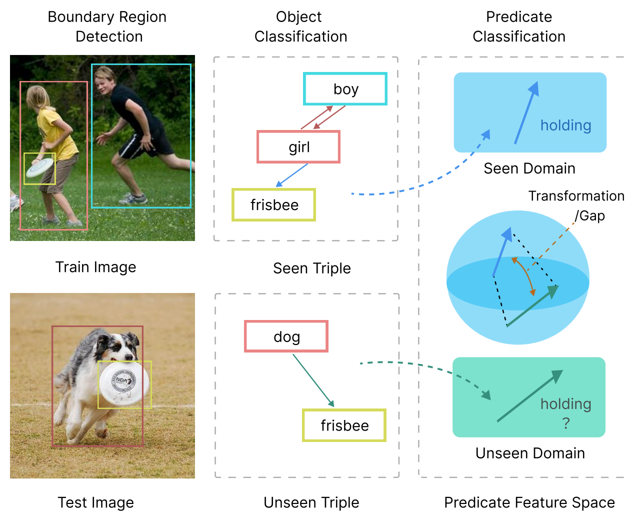

Figure 1. The standard SGG pipeline and Transformation Gap. Zero-Shot SGG aims to acquire knowledge from seen triplets during training and to bridge the domain gap arising from predicate feature transformation in unseen triplets so that ultimately deduce predicates of unseen triplets during testing.

The standard SGG pipeline consists of three stages: boundary region detection, object classification, and predicate classification, with the last stage being the most crucial. Predicate classification uses image features and object labels to predict relationships between objects. To enhance generalization, Li *et al*. [21] have used Triplet Calibration and Reduction to focus on plausible unseen compositions, Kan *et al*. [15] employed Commonsense Knowledge Integration to emphasize the importance of commonsense reasoning in generating scene graphs of unseen triplets. Although notable improvements have been achieved in this task, their performance on zero-shot SGG remains suboptimal. Such limitation is primarily attributed to the models' inability to effectively capture the equivariant property inherent in predicate relationships. As shown in Fig. 1, we hy-

*Corresponding Author

pothesize that seen predicates can be viewed as vectors in a high-dimensional space, and unseen predicates are transformations of these vectors. Although humans can readily discern transformations within the inherent high-dimensional space of their minds, models often focus on invariance, neglecting the learning of these high-dimensional transformations, and thus struggling to adapt them.

We present a novel framework, Capsule Scene Graph Generation (CAPSGG), to strengthen the generalization capabilities of predicate classifiers, and to improve the performance in zero-shot scenarios. The proposed CAPSGG framework Fig. 2 consists of three key components: a Three-Stream Pipeline, a Capsule Network (CapsNet), and a GroupLoss function. Here, we introduced CapsNet to capture features in high-dimensional space. The challenge here is to find a robust predicate representation that facilitates the model to bridge predicate transformations between seen and unseen domains. Capsules solve this by encapsulating the predicate transformations in a higher-dimensional space, where the predicates are equivariant on the spatial manifold of capsules, and the length of the features remains invariant. These three components offer our model a great generalization ability to redirect the model's focus from merely invariant features to a balanced consideration of both invariant and equivariant features, thereby promoting a robust learning framework.

Although distinguishing features between transformations may not fully support category discrimination, it is essential for enabling the model to learn the intrinsic structure of the data [30]. This learning process enhances the model's capacity to generalize and transfer knowledge to novel unseen scenarios. The equivariant property is pivotal in this context, as it enables the model to understand the internal structure of triplet representations. For instance, in the triplet, <dog, holding, frisbee>, despite transformations in the entities' posture or spatial configuration, the model can accurately identify the predicate 'holding' in the variant scenarios. This robustness stems from the model's ability to acquire equivariant features, enabling it to identify the underlying relationship regardless of variations.

We conducted extensive experiments on the Visual Genome (VG) dataset, which demonstrated that our proposed method significantly outperforms the state-of-the-art methods in popular zero-shot metrics (i.e., zR@K, ng-zR@K) for prominent SGG tasks. As shown in Tab. 1 and Tab. 2, with our proposed CAPSGG framework, zero-shot indicators outperform up to **132.26%** on the SGCls task than the T-CAR [21]. Our work presents several key contributions:

1. By clarifying the limitations of existing SGG models in acquiring equivariant features of predicates, we introduced a novel higher-dimensional feature encoder, Capsule Feature Encoder (CFE), to address this challenge. It significantly enhances the model's zero-shot capabilities.

2. We present a Three-Stream Pipeline to accommodate multimodal features, enhancing the model's equivariant feature representation for predicates and thus bolstering its generalization capabilities.

3. We propose GroupLoss, an extension of the traditional MarginLoss, which enhances the model's invariance to predicate representations and mitigates the effects of the long-tail distribution of predicates.

2. Related Work

2.1. Scene Graph Generation

Scene Graph Generation (SGG) is a pivotal task in computer vision, dedicated to extracting and encoding the intricate relationships and interactions among objects within images into a structured format. This structured depiction is not only essential but also indispensable for a range of advanced visual reasoning applications that demand a comprehensive understanding of scene content.

Pioneering works in SGG have endeavored to integrate diverse modalities to enrich predicate understanding, including appearances, relative positions, contextual cues, depth maps, and *etc*. Some approaches focus on the context modeling of entities and relations study; *e.g*. EICR [26] explicitly defined and addressed the context imbalance of subject-object pairs on the SGG dataset, while constructing an environment-invariant relation classifier to tackle the context modeling challenge. CV-SGG [13] believed that visual appearance features may have an unnecessary influence on SGG. SQUAT [14] proposed an edge selection module and quad attention module, which learn to select relevant object pairs and disambiguate them via diverse contextual interactions. Our research is closely related to the existing study, with a focus on how information from various modalities contributes to predicate capsules within the context of individual triplets, specifically emphasizing modeling the equivariant features of predicates.

2.2. Zero-Shot Learning

Zero-Shot Learning has been widely studied in vision tasks such as image classification, with the main approaches based on attributes [20] or semantic embeddings [6, 39]. In the field of SGG, this task was initially proposed by Lu *et al*. [24] and evaluated on the Visual Relationship Dataset (VRD). Several later works have attempted to improve it. Zhang *et al*. [45] learned translation operators in the embedding space; Kan *et al*. [15] used common sense knowledge to assist model reasoning; Li *et al*. [21] optimized a triplet calibration loss; Yu *et al*. [42] generated information on zero-shot triplets before the predicate classifier is trained and then converted the original zero-shot triplets into non-zero samples.

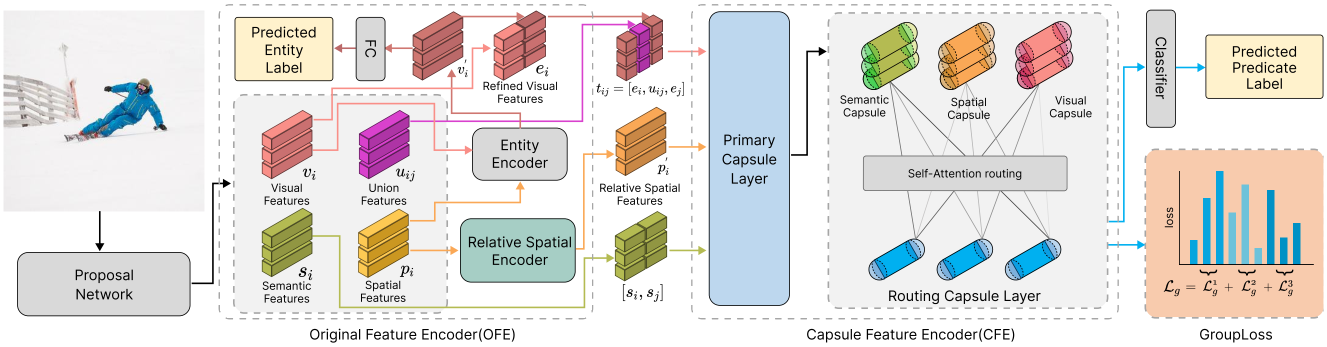

Figure 2. Framework of the proposed CAPSGG model. Given an input image, CAPSGG first detects its object proposals using existing models (*e.g*. Faster-RCNN [29]). OFE then encodes entities and relations from different modalities. These encoded representations are fed into the CFE via a Three-Stream Pipeline, which constructs primary predicate capsules for predicate features from different modalities. Subsequently, the Routing Capsule Layer aggregates these primary capsules to generate the final predicate feature capsule. Finally, GroupLoss is employed to measure the grouping loss, fostering a balance between invariant and equivariant features of the model.

Although these methods advance Zero-Shot Learning significantly, they still perform poorly on zero-shot SGG. Therefore, this paper addresses the zero-shot problem in Scene Graph Generation, which shares similarities with attribute-based zero-shot learning approaches. These approaches assume that all individual object attributes, *e.g*., color and shape, are observed during training, enabling the detection of new categories at test time through attribute combinations. In the context of SGG, while each subject, object, and predicate is presented in the training data, the majority of their potential combinations remain unseen. Our goal is to extract the underlying patterns within the finite set of observed combinations, thereby enabling the model to generalize across the extensive range of unseen combinations.

2.3. Capsule Network

Conventional deep neural networks may struggle to effectively capture the combinatorial and transformational structures of image entities [3, 27, 38]. Hence, to preserve the spatial information, Hinton *et al*. [10] proposed the concept of "capsules" in machine learning terminology to address the representational limitations. A capsule is a vector to represents internal properties that can be used to learn part-whole relationships. CapsNet, as a more effective backbone, was first implemented by Sabour *et al*. [31] in 2017 and has received significant attention from researchers.

Recently, CapsNet has been effectively deployed in various tasks. Yang *et al*. [41] proposed a five-layer CapsNet based on the disease-related compound recognition model to identify disease-related compounds correctly. Heidarian *et al*. [9] introduced a CapsNet-based framework for diagnosing COVID-19 in chest CT scan images. Teto *et al*. [37] conducted a comparative study comparing C-CapsNet and CNN operations in identifying animals in the wild. Other tasks include emotion recognition [1], audio classification [12], and speech detection [16]. Inspired by these works, we integrate a CapsNet encoder into our proposed SGG framework to capture the equivariant features of predicates.

3. Method

This section first formulates the problem of Scene Graph Generation (SGG). Subsequently, we explain the key components of our proposed method in detail.

3.1. Problem Formulation

In the context of SGG, the objective is to generate a graph-structured representation $\mathcal{G} = \{\mathcal{B}, \mathcal{P}, \mathcal{O}\}$ for a given image $\mathcal{I}$, where $\mathcal{B} = \{b_i\}$ denotes the set of bounding boxes of objects, $\mathcal{O} = \{o_i\}$ represents the set of objects, and o_i is the class label of the corresponding box b_i. $\mathcal{P} = \{p_k\}$ is predicates set, where p_k represents the predicate label between two objects (*e.g*., o_i and o_j). Given an image $\mathcal{I}$, the probability of $\mathcal{G}$ can be formulated as:

$$P(\mathcal{G}|\mathcal{I}) = P(\mathcal{B}|\mathcal{I})P(\mathcal{O}|\mathcal{B},\mathcal{I})P(\mathcal{P}|\mathcal{O},\mathcal{B},\mathcal{I}) \quad (1)$$

where $P(\mathcal{B}|\mathcal{I})$ and $P(\mathcal{O}|\mathcal{B},\mathcal{I})$ are usually calculated with pre-trained object detection models, such as Faster-RCNN [29] or Mask-RCNN [8]. Most existing models adhere to this setup [2, 5, 24, 40, 44]. Thus, the primary challenge of the task lies in predicate prediction, *i.e*., calculating $P(\mathcal{P}|\mathcal{O},\mathcal{B},\mathcal{I})$.

3.2. CAPSGG

Fig. 2 illustrates the architecture of our proposed CAPSGG framework. The process begins with an Original Feature Encoding (OFE) module, which extracts visual, semantic, and spatial features of the subject-object relationship. After the Capsule Feature Encoding (CFE) module encapsulates these features as capsules, we form a Three-Stream Pipeline between the two modules. A Routing Capsule Layer then integrates the final predicate capsules, utilizing the equivariance property of CapsNet to effectively manage input variations and unseen triplets. Furthermore, we introduce GroupLoss, which measures intra-group losses for predicates, thereby balancing the learning of the predicate's invariant and equivariant features, and mitigating the impact of the long-tail distribution in predicate frequencies.

3.2.1 Original Feature Encoder

This section addresses the input features from different modalities, with an Entity Encoder and a Relative Position Encoder as its core parts. The Entity Encoder is responsible for refining the features of objects, to facilitate the prediction of the entity labels, as detailed in Eq. (2).

$$\begin{aligned} \boldsymbol{p}_i &= FFN(b_i), \\ \{\boldsymbol{v}_i^{'}\}_{i=1,\ldots,N} &= Enc_{obj}(\{[\boldsymbol{v}_i, \boldsymbol{p}_i]\}_{i=1,\ldots,N}) \end{aligned} \tag{2}$$

where $\boldsymbol{p}_i$ means the encoded position features from bounding box b_i, and $\boldsymbol{v}_i^{'}$ denotes the refined object representations, N is the number of objects, $[\cdot, \cdot]$ represents the concatenation operation, $FFN(\cdot)$ is a three-layer Multi-Layer Perceptron (MLP) with LeakyReLU activation, and Enc_{obj} denotes the entity contextual modules which is a four-layer Transformer.

We employ the Relative Spatial Encoder introduced by Li *et al*. [21] to augment the original spatial feature vector $\boldsymbol{p}_i$, to an enhanced feature $\boldsymbol{p}_i^{'}$, referred as the Relative Spatial Feature. It provides detailed spatial positional information, which is essential for the Capsule Feature Encoding (CFE) module to learn the spatial equivariant properties of the predicates.

Although T-CAR [21] suggests that text features may introduce bias into the model, we claim that they are essential for predicate feature modeling, which is supported by experiment results in Tab. 3. Accordingly, we introduce a third branch to augment the existing dual-branch architecture, dedicated to generating text features that provide essential cues for subsequent predicate modeling.

The Original Feature Encoding (OFE) module allows for the extraction of consistent predicate features across three modalities: visual, semantic, and spatial. To further develop the equivariant features of the predicate, we introduce the Capsule Feature Encoding (CFE) module.

3.2.2 Capsule Feature Encoder

The Capsule Feature Encoder consists of a CapsNet, as shown in Fig. 2. To address the zero-shot Scene Graph Generation (SGG) challenge, we claim that unseen predicates can be represented by transformations of seen predicate features in a high-dimensional feature space. Our objective here is to design a model capable of effectively managing these transformations, a functionality that aligns with the nature of CapsNet. Adopting the inherent equivariant property of CapsNet, we can effectively capture the equivariant features of the predicates.

CapsNet, which has been validated in image classification tasks, allows each capsule to recognize a visual entity under specific observation conditions and deformations. It outputs both the entity's existence probability and its "instance parameters", such as pose, lighting, and deformation. These capsules ensure local invariance in the entity's existence probability as it varies within the capsule's range. The instance parameters, however, are equivariant, adjusting with changes in observation conditions to reflect the entity's intrinsic coordinates on the appearance manifold. Applying this concept to predicates, CapsNet can learn the equivariant features of predicate entities, bridging the gap between seen and unseen domains through consistent model responses to feature transformations.

A typical CapsNet consists of two layers: a Primary Capsule Layer and a Routing Capsule Layer. Specifically, the original CapsNet takes an input image $\mathbf{X}$ of shape $H \times W \times C$, where H, W, and C represent the image's height, width, and channels. To encapsulate the features, a 2D convolution is applied to extract the features of the input image, forming the initial capsule with:

$$\mathbf{U}_{caps} = Caps(\mathrm{ReLU}\left(Conv_{k\times k}(\mathbf{X})\right)) \tag{3}$$

As shown in Fig. 2, we have improved the model to incorporate multimodal features $\mathbf{E}$, carrying the same shape (B, F). B is the batch_size; precisely, it is the number of object pairs sampled in a set of images. And the F is the feature of object pairs. Here, we replace the original 2D convolution layers and Relu activation with a single-layer Multi-Layer Perceptron(MLP) and LeakyReLU activation:

$$\mathbf{U}_{caps} = Caps(\mathrm{LeakyReLU}\left(MLP(\mathbf{E})\right)) \tag{4}$$

In the Routing Capsule Layer, we adopt a low-parameter, non-iterative, parallelizable routing algorithm, known as Self-Attention routing [25], replacing the original computationally intensive dynamic routing algorithm [31]. For the primary capsules of the tri-modal input $\boldsymbol{u}_{vis}, \boldsymbol{u}_{txt}, \boldsymbol{u}_{pos}$, we define weight matrices $\mathbf{W}_{vis}, \mathbf{W}_{txt}, \mathbf{W}_{pos}$, which encode all the affine transformations between capsules in adjacent layers. So, each capsule in layer l will constructs its projections from the layer above, follows Eq. (5):

$$\begin{aligned} \mathbf{U}_{vis}^{l} &= \boldsymbol{u}_{vis}^{l\mathrm{T}} \times \mathbf{W}_{vis}^{l} \\ \mathbf{U}_{txt}^{l} &= \boldsymbol{u}_{txt}^{l\mathrm{T}} \times \mathbf{W}_{txt}^{l} \\ \mathbf{U}_{pos}^{l} &= \boldsymbol{u}_{pos}^{l\mathrm{T}} \times \mathbf{W}_{pos}^{l} \end{aligned} \tag{5}$$

where $\mathbf{U}^l_\cdot$ denotes the prediction output of the l-th capsule, with dimensions (n^l, n^{l+1}, d^{l+1}). Here, n^l and n^{l+1} denote the number of capsules in layer l and $l+1$, respectively, while d^{l+1} indicates the dimensionality of the subsequent layer. Each capsule in n^l, using the weight matrix, predicts the properties of all n^{l+1} capsules. Consequently, the capsules of the next layer $\boldsymbol{s}_n^{l+1}$ can be computed as Eq. (6):

$$\begin{aligned}\hat{\mathbf{U}}^l &= [\mathbf{U}^l_{vis}; \mathbf{U}^l_{txt}; \mathbf{U}^l_{pos}] \\ \boldsymbol{s}_n^{l+1} &= \hat{\mathbf{U}}^l \times \left(\mathbf{C}^l + \mathbf{B}^l\right)\end{aligned} \tag{6}$$

where $\mathbf{C}^l$ is the coupling coefficients generated by the self-attention algorithm, which can be computed via Eq. (7):

$$\begin{aligned}\mathbf{A}^l &= \frac{\hat{\mathbf{U}}^l \times \hat{\mathbf{U}}^{lT}}{\sqrt{d^l}} \\ \mathbf{C}^l &= \frac{\exp \sum_{n^l} \mathbf{A}^l}{\sum_{n^{l+1}} \exp \sum_{n^l} \mathbf{A}^l}\end{aligned} \tag{7}$$

To encode the probability of an entity's presence through the vector's length and to enable the active capsule to predict instantiation parameters the instantiation parameters of higher-level capsules, Hinton *et al*. [10] suggest that the activation function should i) preserve the vector's direction and ii) constrain its length to a range between 0 and 1. Based on this idea, we employ a variant of the original activation function introduced in the [25], known as the squash operation:

$$\mathrm{squash}(s_i) = \left(1 - \frac{1}{e^{||s_i||}}\right) \frac{s_i}{||s_i||} \tag{8}$$

Here, we denote a single capsule as s_i, representing the individual entries i of S. After the operation with Eq. (8), we can obtain the real predicate capsule $\mathbf{U}_i^{l+1}$.

3.2.3 GroupLoss

After the Routing Capsule Layer, the model's output for the predicate category is no longer represented by a scalar, but by a vector. The length of the instantiation vector represents the probability of the existence of the capsule entity. This length should approach 1 iff the predicate entity it means is the sole entity existing in the feature space. We employ MarginLoss, originally designed for multi-label classification, as our main training objective. This choice is based on our hypothesis that fully inhibiting other negative predicates could excessively widen the score difference between predicted positive and negative classes, which could potentially impair the model's generalization ability. Expanding on the MarginLoss, we developed GroupLoss, which initially groups identical predicates and then calculates the loss within each group to replace the calculation of global loss. The specific formula is as follows:

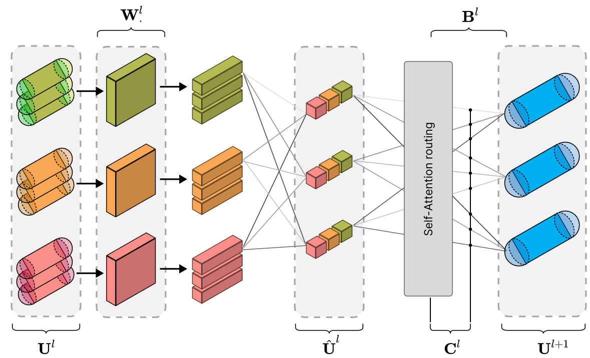

Figure 3. Illustration of the Routing Capsule Layer. Each routing iteration utilizes the capsules $\mathbf{U}.^l$ from the preceding layer, the weight tensor $\mathbf{W}.^l$, coupling coefficients matrices $\mathbf{C}^l$, and priors $\mathbf{B}^l$ to generates the subsequent layer of capsules $\mathbf{U}^{l+1}$.

$$\begin{aligned}v_g &= [v_1, v_2, ..., v_i]_{v_i \in V, l_i = k} \\ L &= T_g \max\left(0, m^+ - \|\mathbf{v}_g\|\right)^2 \\ &\quad + \lambda(1 - T_g) \max\left(0, \|\mathbf{v}_g\| - m^-\right)^2\end{aligned} \tag{9}$$

Here v_g represents a predicate capsule group, where the labels l_i of v_i in the group are identical and equal to k, Additionally, $T_g = 1$ iff the class of predicate group k is present with $m^+ = 0.8$ and $m^- = 0.2$. The parameter λ is used to balance the final length of the target predicate capsule and the background predicate capsule to avoid the length of the target predicate capsule approaching m^- closely. An incorrect λ setting can result in a phenomenon that is called "metric inversion".

4. Experiments

In this section, we conduct extensive experiments on the large-scale Visual Genome (VG) dataset. The results indicate that CAPSGG excels beyond state-of-the-art models with superior performance on zero-shot SGG.

4.1. Settings

Dataset. Our experimental evaluations for zero-shot SGG are conducted using the VG dataset [19]. We adhere to the standard VG150 split [18, 34, 40, 44], which includes the 150 most prevalent object categories and 50 relation categories present within the Visual Genome (VG) dataset.

Tasks. We evaluate our approach on three SGG tasks: (i) Predicate Classification (PredCls), which identifies the relationships between entity pairs given their bounding boxes and categories; (ii) Scene Graph Classification (SGCls), which predicts the categories of triplets based on their bounding boxes; and (iii) Scene Graph Detection (SGDet), which involves detecting bounding boxes for entity pairs and inferring their predicate categories.

Table 1. Zero-Shot performance comparison results of our CAPSGG method with state-of-the-art SGG models on three SGG tasks **without graph constraint**. "-" indicates that the result is either not mentioned in the original paper or the method is not applicable in that configuration

Models	PredCls			SGCls			SGDet		
	zR@20	zR@50	zR@100	zR@20	zR@50	zR@100	zR@20	zR@50	zR@100
IMP [40]	14.4	27.5	38.9	1.5	3.8	6.8	0.1	0.5	1.1
IMP++ [17]	14.6	27.3	39.0	3.1	7.5	11.4	0.1	0.4	0.9
Motifs [44]	14.3	27.4	39.7	3.1	6.9	11.2	1.1	2.7	4.5
VTransE [45]	8.2	17.9	28.6	1.6	3.9	7.0	0.9	1.7	3.2
TDE [35]	8.9	17.4	26.7	1.3	4.2	8.3	0.0	0.1	0.4
BGNN [22]	2.2	7.5	15.6	1.3	3.9	6.8	0.1	0.2	0.5
UVTransE [11]	12.7	25.4	37.3	2.7	5.9	9.8	0.9	2.3	3.9
EBM [32]	13.8	26.2	38.3	4.1	8.5	13.9	1.3	2.7	4.4
KGC* [42]	14.8	28.3	41.3	3.7	8.3	13.4	1.6	3.3	5.5
RGN* [43]	14.9	28.2	41.1	4.8	10.4	16.2	1.7	3.2	5.3
COACHER [15]	13.4	19.3	22.2	-	-	-	-	-	-
T-CAR [21]	<u>27.4</u>	<u>39.8</u>	<u>50.8</u>	<u>7.8</u>	<u>12.4</u>	<u>16.7</u>	<u>3.6</u>	<u>6.2</u>	<u>8.7</u>
CAPSGG(ours)	**29.3**	**51.7**	**68.0**	**16.6**	**28.8**	**35.1**	**9.3**	**12.5**	**13.3**
Promote	6.93%	29.90%	33.86%	112.82%	132.26%	110.18%	158.33%	101.61%	52.87%

* It is a plug-and-play module that uses the best-reported combination of results

Table 2. Zero-Shot performance comparison results of our CAPSGG method with state-of-the-art SGG models on three SGG tasks **with graph constraint**. "-" indicates that the result is not mentioned in the original paper or the method is not applicable in that configuration

Models	PredCls		SGCls		SGDet	
	zR@50/100	R@50/100	zR@50/100	R@50/100	zR@50/100	R@50/100
IMP [40]	17.5 / 19.9	61.3 / 63.3	1.9 / 2.2	61.3 / 35.3	0.5 / 0.9	25.7 / 31.2
IMP++ [17]	19.2 / 22.4	62.1 / 64.3	4.3 / 5.1	40.0 / 40.8	0.4 / 0.9	21.1 / 27.4
Motifs [44]	17.7 / 20.5	65.0 / 66.9	4.1 / 5.0	39.1 / 39.9	1.9 / 2.7	32.6 / 37.0
VTransE [45]	11.7 / 15.0	58.2 / 62.6	1.9 / 2.3	33.4 / 35.6	0.9 / 1.6	27.2 / 31.5
TDE [35]	12.5 / 16.4	45.7 / 51.1	2.6 / 3.5	28.0 / 30.5	2.0 / 2.6	16.7 / 20.3
BGNN [22]	3.5 / 5.2	58.1 / 60.9	1.7 / 2.2	36.2 / 37.4	0.1 / 0.3	25.9 / 31.1
UVTransE [11]	16.5 / 18.9	64.7 / 66.4	3.3 / 3.9	37.9 / 38.8	1.2 / 2.1	31.9 / 36.1
EBM [32]	16.8 / 20.0	64.5 / 66.5	5.3 / 6.2	43.5 / 44.7	2.0 / 3.0	30.3 / 34.6
NARE [7]	13.5 / -	47.6 / 52.0	6.2 / -	32.8 / 35.8	3.3 / -	19.0 / 21.0
KGC* [42]	18.8 / 21.9	- / -	5.0 / 6.1	- / -	2.2 / 3.3	- / -
RGN* [43]	18.4 / 21.2	- / -	6.4 / 7.7	- / -	2.5 / 3.5	- / -
T-CAR [21]	<u>31.9</u> / <u>34.9</u>	60.0 / 63.0	<u>9.3</u> / <u>10.6</u>	40.4 / 42.0	<u>4.7</u> / <u>6.0</u>	28.5 / 32.9
CAPSGG(ours)	**37.7 / 46.1**	63.4 / 67.7	**20.0 / 24.3**	35.4 / 39.0	**10.2 / 12.8**	18.3 / 22.8
Promote	18.18% / 32.09%	- / -	115.05% / 129.25%	- / -	117.02% / 113.33%	- / -

* It is a plug-and-play module that uses the best-reported combination of results

Evaluation Metric and Protocol. We evaluate our SGG method using the Recall@K (R@K) and Zero-Shot Recall@K (zR@K) metrics, as well as their unconstrained versions. Among them, R@K measures the proportion of ground-truth triplets within the top K predicted triplets per image, while zR@K focuses specifically on zero-shot triplets. We calculate both metrics by averaging these proportions across all images. Notably, although these metrics are standard in the field [7, 11, 17, 18, 32, 35, 36], their evaluation protocols vary. The protocols are categorized based on three criteria: the consideration of Object Overlap, the use of a Validation Set, and the application of Frequency Bias. Accordingly, we adopt the unified testing protocol proposed by Li *et al*. [21], where no Object Overlap is required, no Validation Set is applied, and Frequency Bias is removed.

Implementation Details. Following prior works [4, 21, 46], we adopt the pre-trained Faster-RCNN [29] with ResNeXt-101-RPN [23] to detect entities in the images. GloVe [28] is applied to embed the semantic features. We set the parameter λ in Eq. (9) as 0.15, as explained in Sec. 4.3. We optimize our method using SGD, with an initial learning rate of 5^{-3} and a batch size of 4. Our codes are implemented with PyTorch [33] and run on two NVIDIA GeForce RTX 3090 GPUs.

4.2. Comparison with State-of-the-art Methods

To evaluate the performance of our model, we compare it with several state-of-the-art SGG methods on the VG dataset. including IMP [40], VTransE [45], Motifs [44], and IMP++ [17]. These methods emphasize the performance of visible triplets, leading to the poor performance on unseen triplets. We also compare our model with existing methods that focus on studying unseen triplets, including UVTransE [11], EBM [32], NARE [7], GRAPHN [18], COACHER [15], KGC [43], RGN [42], and T-CAR [21], to highlight the superior performance of our model on unseen triplets tasks. In addition, TDE [35] and BGNN [22] for unbiased scene graph generation are also compared.

Comparison with Graph Constraint. Tab. 1 shows the comparison results of our model on the VG dataset with graph constraint. The results demonstrate that our CAPSGG model consistently and significantly outperforms the state-of-the-art methods across all three tasks. In particular, we achieve a performance improvement of **132.26%** on the SGCls task. Compared with models that solely focus on modeling predicate invariant features, our method fully demonstrates that an SGG model can effectively address the gap between different domains in zero-shot tasks by incorporating the equivariant features.

Comparison without Graph Constraint. To comprehensively evaluate the generalization capability of our CAPSGG approach, we present comparison results under the settings without graph constraints, as introduced by Zellers *et al*. [44]. This setting allows multiple edges between subjects and objects in the output scene graph, considering all predicate probabilities for a single object pair. This flexibility generally enables higher recall by allowing the model to make multiple inferences on complex relations. As shown in Tab. 2, our CAPSGG method outperforms state-of-the-art methods across all tasks under these conditions, highlighting its robust generalization ability. Additionally, we observed an intriguing phenomenon of ”metric inversion”, which is discussed in detail in Sec. 4.3.

4.3. Ablation Studies

We conduct ablation studies to evaluate the importance of each key component in CAPSGG, namely the **Three-Stream Pipeline, GroupLoss, and the hyperparameter** λ. Next, we will analyze the results of the ablation experiments for each component individually.

Three-Stream Pipeline. As shown in Fig. 2, the Three-Stream Pipeline spans two modules, OFE and CFE. The OFE module organizes the Three-Stream predicate modality features as the input for the CEF module. Simultaneously, the CEF module merges these features and constructs Primary Predicate Capsules, which are then passed to the Routing Capsule Layer to generate the Final Predicate Capsules. We structured the ablation study by gradually removing each branch while retaining the visual branch, as we consider this a visual task where visual features should primarily guide the model.

Table 3. Ablation studies on Three-Stream Pipeline (OFE side)

Branch			zR@K			ng-zR@K		
vis	txt	pos	20	50	100	20	50	100
✔	✘	✘	21.6	34.9	42.4	25.2	46.2	62.9
✔	✘	✔	18.7	33.7	42.0	22.9	44.2	63.4
✔	✔	✘	21.3	34.2	41.6	26.7	49.4	66.0
✔	✔	✔	**24.2**	**37.7**	**46.1**	**29.3**	**51.7**	**68.0**

Table 4. Ablation studies on Three-Stream Pipeline (CFE side)

Models	zR@K			ng-zR@K		
	20	50	100	20	50	100
CAPSGG†	16.4	33.3	41.9	21.7	50.1	**69.7**
CAPSGG	**24.2**	**37.7**	**46.1**	**29.3**	**51.7**	68.0

† means that the three-branch structure of the CFE module is not used.

As shown in Tab. 3, removing any modality branch significantly impairs the model's zero-shot Scene Graph Generation (SGG) performance, with each branch contributing to the model's capabilities. Interestingly, unlike T-CAR [21], which suggests that semantic labels might hinder generalization, our model benefits significantly from the textual branch. This improvement is due to our model's emphasis on predicate equivariant features rather than invariant ones, allowing it to better leverage diverse modalities and adapt to variations of unseen data.

Additional experiments were conducted on the Capsule Feature Encoding (CFE) module to further evaluate the Three-Stream Pipeline. Here, we replaced the multi-modal capsule construction method with a direct concatenation of features. The results, detailed in Tab. 4, show a significant performance drop without the Three-Stream Pipeline in the CFE module, confirming its effectiveness.

Table 5. Ablation studies on GroupLoss

Tasks	GL	zR@K			ng-zR@K		
		20	50	100	20	50	100
PredCls	✘	19.1	28.9	35.9	23.2	42.0	57.3
	✔	**21.3**	**35.8**	**45.2**	**25.0**	**44.5**	**61.0**
SGCls	✘	7.1	12.0	16.4	5.5	7.8	9.7
	✔	**15.7**	**24.7**	**29.7**	**18.7**	**30.2**	**35.8**

GroupLoss. An ablation experiment was conducted on GroupLoss, revealing a significant performance decline when GroupLoss was removed and only the original MarginLoss from the CapsNet was used, as shown in Tab. 5. This performance decline is primarily caused by the model's diminished ability to effectively capture the invariant features of predicates. In contrast, our GroupLoss aggregates predicate capsules of the same category to calculate the intra-group loss, which implicitly guides the model in capturing the invariant characteristics of the predicates. Our approach helps to mitigate the pronounced long-tail distribution issue, ensuring a more balanced learning process between equivariant and invariant features.

Table 6. Ablation studies on Hyperparameter λ

λ	zR@K			ng-zR@K		
	20	50	100	20	50	100
0.5	14.2	20	23.5	14.4	18.3	19.2
0.15	**12.2**	**20.0**	**24.3**	**16.6**	**28.8**	**35.1**
0.1	9.9	18.2	23.3	14.6	26.3	33.1
0.05	8.5	17.2	23.0	11.4	22.0	31.5

Hyperparameter λ. As mentioned in Sec. 4.2, our ablation study on the hyperparameter λ revealed a phenomenon we term "metric inversion". The experimental results, shown in Tab. 6, indicate that varying the λ values when using the MarginLoss from CapsNet affects model performance. Specifically, assigning excessive weight to the negative class significantly increases the likelihood of "metric inversion", where the model's performance without graph constraints is markedly inferior to that with constraints. This occurs because an overemphasis on the negative class causes the model to focus on negative

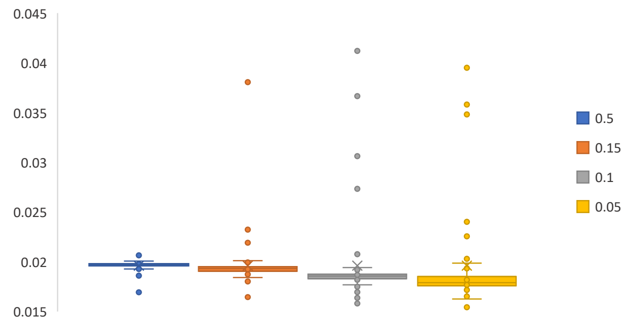

Figure 4. Distribution of Predicate Prediction Scores. This figure presents the intervals of predicate prediction scores across varying λ settings for a consistent random sample.

scores, reducing the distinction between positive and negative classes. For instance, when λ is set to 0.5, the range of predicted scores for predicates is compressed to a narrow interval, as shown in Fig. 4.

Metrics with graph constraints are less affected by class indistinguishability, as they select the label with the highest score. However, without such constraints, metrics must consider all possible categories, making predictions appear random when classes are not well distinguished. We also found that overemphasizing the positive class undermines zero-shot performance. Notably, as λ decreases from 0.15 to 0.05, the distinguishability of positive predicates drops due to lax constraints on negative predicates, while negative predicate scores become more dispersed. This divergence jeopardizes the model's ability to provide reliable guidance during uncertain predictions. Therefore, we conclude that a balance between the distinguishability and divergence of scores from positive and negative predicate classes is an essential factor in determining the final prediction performance in zero-shot SGG tasks.

Apart from these key comparisons, additional quantitative and qualitative experiments are provided in the Supplementary Materials for more comprehensive comparisons.

5. Conclusion

We introduced an innovative framework, Capsule Scene Graph Generation (CAPSGG), for the first attempt at modeling equivariant features within SGG using CapsNet. The proposed framework, integrating a CapsNet, a Three-Stream Pipeline structure, and a Grouploss measure, offers a robust generalization capability for SGG. Our model proficiently manages the complexities of unseen triplets and effectively addresses the zero-shot challenge in SGG. Extensive experimental results evidence the effectiveness and superiority of our method compared to existing models. Our framework expands the applicability of SGG beyond traditional approaches, laying a foundation for more robust visual reasoning applications. The identification of equivariant and "metric inversion" phenomena offer potential insights for future research endeavors.

Acknowledgments. This work was partially supported by a project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[1] Hao Chao, Liang Dong, Yongli Liu, and Baoyun Lu. Emotion recognition from multiband eeg signals using capsnet. *Sensors*, 19(9):2212, 2019. 3

[2] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2019. 3

[3] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 3

[4] Xingning Dong, Tian Gan, Xuemeng Song, Jianlong Wu, Yuan Cheng, and Liqiang Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19427–19436, 2022. 7

[5] Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, and Fei-Fei Li. Visual relationships as functions: Enabling few-shot scene graph prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[6] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013. 2

[7] Arushi Goel, Basura Fernando, Frank Keller, and Hakan Bilen. Not all relations are equal: Mining informative labels for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15596–15606, 2022. 6, 7

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[9] Shahin Heidarian, Parnian Afshar, Nastaran Enshaei, Farnoosh Naderkhani, Moezedin Javad Rafiee, Faranak Babaki Fard, Kaveh Samimi, S Farokh Atashzar, Anastasia Oikonomou, Konstantinos N Plataniotis, et al. Covid-fact: A fully-automated capsule network-based framework for identification of covid-19 cases from chest ct scans. *Frontiers in Artificial Intelligence*, 4:598932, 2021. 3

[10] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, pages 44–51. Springer, 2011. 3, 5

[11] Zih-Siou Hung, Arun Mallya, and Svetlana Lazebnik. Contextual translation embedding for visual relationship detection and scene graph generation. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3820–3832, 2020. 6, 7

[12] Royal Jain. Improving performance and inference on audio classification tasks using capsule networks. *arXiv preprint arXiv:1902.05069*, 2019. 3

[13] Tianlei Jin, Fangtai Guo, Qiwei Meng, Shiqiang Zhu, Xiangming Xi, Wen Wang, Zonghao Mu, and Wei Song. Fast contextual scene graph generation with unbiased context augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6302–6311, 2023. 2

[14] Deunsol Jung, Sanghyun Kim, Won Hwa Kim, and Minsu Cho. Devil's on the edges: Selective quad attention for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18664–18674, 2023. 2

[15] Xuan Kan, Hejie Cui, and Carl Yang. Zero-shot scene graph relation prediction through commonsense knowledge integration. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*, pages 466–482. Springer, 2021. 1, 2, 6, 7

[16] Shakir Khan, Ashraf Kamal, Mohd Fazil, Mohammed Ali Alshara, Vineet Kumar Sejwal, Reemiah Muneer Alotaibi, Abdul Rauf Baig, and Salihah Alqahtani. Hcovbi-caps: hate speech detection using convolutional and bi-directional gated recurrent unit with capsule network. *IEEE Access*, 10:7881–7894, 2022. 3

[17] Boris Knyazev, Harm De Vries, Cătălina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Graph density-aware losses for novel compositions in scene graph generation. *arXiv preprint arXiv:2005.08230*, 2020. 6, 7

[18] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Generative compositional augmentations for scene graph prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15827–15837, 2021. 5, 7

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 5

[20] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013. 2

[21] Jiankai Li, Yunhong Wang, and Weixin Li. Zero-shot scene graph generation via triplet calibration and reduction. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(1):1–21, 2023. 1, 2, 4, 6, 7

[22] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11109–11119, 2021. 6, 7

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 7

[24] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 852–869. Springer, 2016. 2, 3

[25] Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. Efficient-capsnet: Capsule network with self-attention routing. *Scientific reports*, 11(1):14634, 2021. 4, 5

[26] Yukuan Min, Aming Wu, and Cheng Deng. Environment-invariant curriculum relation learning for fine-grained scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13296–13307, 2023. 2

[27] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2865–2873, 2015. 3

[28] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 7

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 3, 7

[30] Mamshad Nayeem Rizve, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10836–10846, 2021. 2

[31] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017. 3, 4

[32] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13936–13945, 2021. 6, 7

[33] Kaihua Tang. A scene graph generation codebase in pytorch, 2020. `https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch`. 7

[34] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6619–6628, 2019. 5

[35] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3716–3725, 2020. 6, 7

[36] Yao Teng and Limin Wang. Structured sparse r-cnn for direct scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19437–19446, 2022. 7

[37] Joel Kamdem Teto and Ying Xie. Automatically identifying of animals in the wilderness: Comparative studies between cnn and c-capsule network. In *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*, pages 128–133, 2019. 3

[38] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5028–5037, 2017. 3

[39] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 69–77, 2016. 2

[40] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017. 3, 5, 6, 7

[41] Bin Yang, Wenzheng Bao, and Jinglong Wang. Active disease-related compound identification based on capsule network. *Briefings in bioinformatics*, 23(1):bbab462, 2022. 3

[42] Xiang Yu, Ruoxin Chen, Jie Li, Jiawei Sun, Shijing Yuan, Huxiao Ji, Xinyu Lu, and Chentao Wu. Zero-shot scene graph generation with knowledge graph completion. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 2, 6, 7

[43] Xiang Yu, Jie Li, Shijing Yuan, Chao Wang, and Chentao Wu. Zero-shot scene graph generation with relational graph neural networks. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1894–1900. IEEE, 2022. 6, 7

[44] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5831–5840, 2018. 3, 5, 6, 7

[45] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5532–5540, 2017. 2, 6, 7

[46] Chaofan Zheng, Xinyu Lyu, Lianli Gao, Bo Dai, and Jingkuan Song. Prototype-based embedding network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22783–22792, 2023. 7