Scene Graph Generation with Role-Playing Large Language Models

Guikun Chen[1*], **Jin Li**[3*], **Wenguan Wang**[1,2†]

[1]Zhejiang University
[2]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Xi'an Jiaotong University
[3]Changsha University of Science & Technology

https://github.com/guikunchen/SDSGG

Abstract

Current approaches for open-vocabulary scene graph generation (OVSGG) use vision-language models such as CLIP and follow a standard zero-shot pipeline – computing similarity between the query image and the text embeddings for each category (*i.e.*, text classifiers). In this work, we argue that the text classifiers adopted by existing OVSGG methods, *i.e.*, category-/part-level prompts, are *scene-agnostic* as they remain unchanged across contexts. Using such fixed text classifiers not only struggles to model visual relations with high variance, but also falls short in adapting to distinct contexts. To plug these intrinsic shortcomings, we devise SDSGG, a *scene-specific* description based OVSGG framework where the weights of text classifiers are adaptively adjusted according to the visual content. In particular, to generate comprehensive and diverse descriptions oriented to the scene, an LLM is asked to play different roles (*e.g.*, biologist and engineer) to analyze and discuss the descriptive features of a given scene from different views. Unlike previous efforts simply treating the generated descriptions as *mutually equivalent* text classifiers, SDSGG is equipped with an advanced *renormalization* mechanism to adjust the influence of each text classifier based on its relevance to the presented scene (this is what the term "*specific*" means). Furthermore, to capture the complicated interplay between subjects and objects, we propose a new lightweight module called mutual visual adapter. It refines CLIP's ability to recognize relations by learning an interaction-aware semantic space. Extensive experiments on prevalent benchmarks show that SDSGG outperforms top-leading methods by a clear margin.

1 Introduction

SGG [1] aims to create a structured representation of an image by identifying objects as nodes and their relations as edges within a graph. The emerging field of OVSGG [2, 3], which broadens the scope of SGG to identify and associate objects beyond a predefined set of categories, has become a research hotspot for its prospective to amplify the practicality of SGG in diverse real-world applications.

OVSGG has achieved notable progress due to the success of vision-language models (VLMs) [4, 5] and prompt learning [6, 7]. Existing OVSGG methods adopt a standard zero-shot pipeline [4], which computes similarity between the visual embedding from query image and the text embedding from pre-defined text classifiers (*c.f.* Fig. 1a). One straightforward direction for OVSGG is to use only the category name (*e.g.*, "riding") [2, 8, 9, 3, 10] as the text classifier and perform vision-language

---

* The first two authors contribute equally to this work.
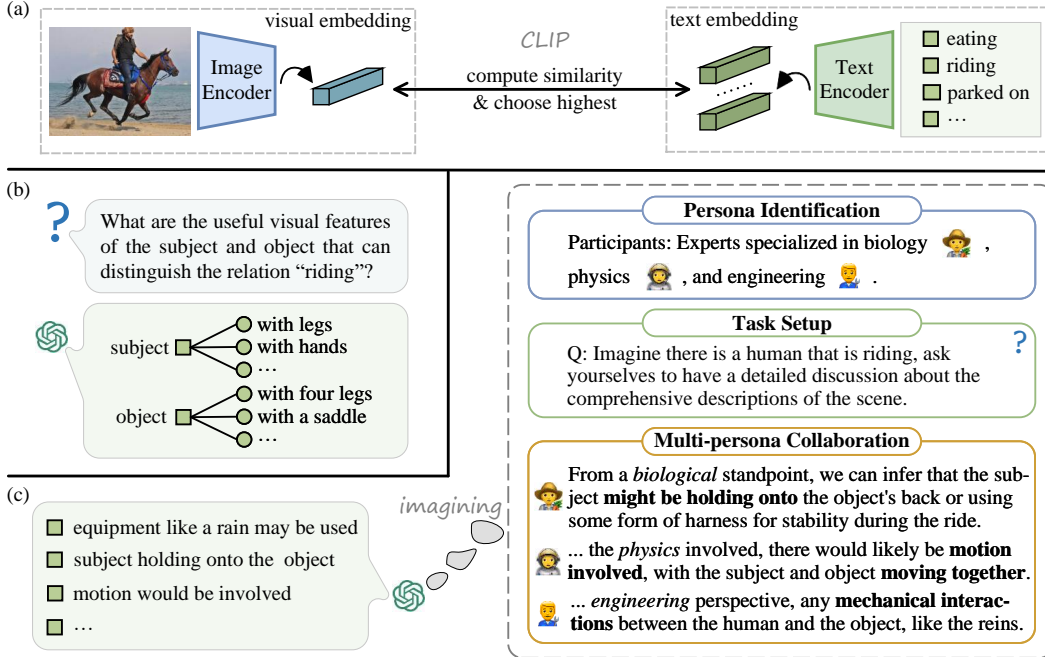† Corresponding Author: Wenguan Wang.

Figure 1: Illustration of the used text classifiers in OVSGG. (a) CLIP performs zero-shot classification by computing similarity between the query image and the text embeddings for each category, then choosing the highest. (b) To further utilize the learned semantic space of CLIP, one can compute similarities of multiple part-level prompts (*e.g.*, the object of ⟨man, riding, horse⟩ may be described with "with four legs" and "with a saddle"). (c) Instead of using these *scene-agnostic* text classifiers, SDSGG adopts comprehensive, *scene-specific* descriptions generated by LLMs, which can adapt to specific contexts by using the proposed renormalization.

alignment as in prompt learning to learn the underlying patterns. On the other hand, [11, 12] argue that such methods fail to utilize the rich context of additional information that language can provide. To address this, [12] decomposes relation detection into several *separate* components, computing similarities by checking how well the visual features of the object match the part-level descriptions[1]. As shown in Fig. 1b, the object of relation "riding" should have four legs, a saddle, *etc*.

Despite these technological advances, we still observe current OVSGG systems lack in-depth inspection of the expressive range of the used text classifiers, which puts a performance cap on them. Concretely, OVSGG models that rely on *scene[2]-agnostic* text classifiers have the following flaws. **First**, methods based on category names [2, 3] struggle to model the large variance in visual relations. Using only category names as classifiers [4] does hold water when applied to object recognition [13]. For instance, CLIP shares common visual features across diverse image-text pairs of tigers which encompasses a variety of tiger appearances and corresponding descriptions. Nonetheless, the scenario becomes far more complex when it comes to relation detection. The visual features that define the relation "on" can vary dramatically across scenes, *e.g.*, "dog on bed" *vs.* "people on road". RECODE [12] proposes to decompose relation detection into recognizing part-level descriptions for both subject and object, hence partially easing the aforementioned difficulty. Yet, it computes similarities for the subject and object *separately* and does not model the *interplay* between subjects and objects. **Second**, part-level prompt based methods uniformly process all descriptions as affirmative classifiers [11, 12], overlooking the possibility that some text classifiers might be contrary to specific contexts. When querying LLMs for distinctive visual features of subjects and objects to distinguish the predicate "riding", with the subject as "human" and the object as "horse", LLMs provide part-level descriptions of the expected appearance of both entities. All generated descriptions are treated equally as definitive text classifiers. However, these descriptions could potentially be misleading, as LLMs produce them without considering the specific context, even resulting in some descriptions that are wholly irrelevant to the presented image. For example, LLMs typically associate the predicate "riding" with the animal

---

[1] The terms "description" and "prompt" are used interchangeably, all denoting text classifiers.

[2] A scene typically involves objects and their interplay and relationships.

2

"with four legs". Nonetheless, such associations are indeed irrelevant, as an animal's legs are not always visible in the presented image, or the animal may have only two legs.

Filling the gaps identified above calls for a fundamental paradigm shift: moving from the fixed, *scene-agnostic* (*i.e.*, category-/part-level) text classifiers towards flexible, *scene-specific* ones. In light of this, we develop SDSGG, a scene-specific description based OVSGG framework that utilizes text classifiers generated from LLMs, complemented by a renormalization technique, to understand scenes from different perspectives. **For the textual part**, given a scene with specified content, an LLM is assigned distinct roles, akin to experts specializing in biology, physics, and engineering, to analyze descriptive scene features comprehensively (*c.f*. Fig. 1c). Such a multi-persona scheme is designed to improve the diversity of the generated scene descriptions as LLMs tend to generate repetitive content. LLM can be queried multiple times to obtain a large number of scene descriptions. Moreover, since not all descriptions are relevant to the presented image (*e.g.*, some parts of the object may not appear), SDSGG is equipped with an advanced mechanism that renormalizes each scene description via opposite descriptions corresponding to the original descriptions. This involves evaluating two vision-language similarities: one for the original scene description and the other for its opposite. The *difference* between the two similarities is viewed as the *self-normalized* similarity of the scene description, allowing for flexible control over its influence. For instance, an irrelevant description would yield a *self-normalized* similarity close to zero, as the two similarity scores of it and its opposite would be very close. By doing so, the generated scene-level descriptions become flexible, *scene-specific* descriptions (SSDs). **For the visual part**, we propose a new adapter for relation detection, called mutual visual adapter, which consists of several lightweight learnable modules. The proposed adapter projects CLIP's semantic embeddings into another interaction-aware space, modeling the complicated interplay between the subject and object through cross-attention.

With the proposed adaptive SSDs, our SDSGG is capable of: **i)** adapting to the given context via evaluating the *self-normalized* similarity of each SSD; **ii)** alleviating the overfitting problem in OVSGG models [2, 3] that use only one classifier; and **iii)** naturally generalizing to novel relations by associating them with SSDs. We validate SDSGG on two widely-used benchmarks, *i.e.* Visual Genome (VG) [14] and GQA [15]. Experimental results show that SDSGG outperforms existing OVSGG methods [3] by a large margin. The strong generalization and promising performance of SDSGG evidence the great potential of *scene-specific* description based relation detection.

2   Related Work

Scene Graph Generation. Since [1] introduces iterative message passing for SGG, research studies in structured visual scene understanding have witnessed phenomenal growth. Tremendous progress has been achieved and can be categorized into: **i)** Two-stage SGG [16, 17, 18, 19, 20], which first detects all objects in the images and then recognizes the pairwise relations between them; **ii)** Debiased SGG [21, 22, 23, 24, 25, 26, 27, 28], which focuses on the problem of long-tailed predicate distribution in the current dataset; **iii)** Weakly-supervised SGG [29, 30, 31, 32, 33], which investigates how to generate scene graph with only image-level supervision; **iv)** One-stage SGG [34, 35, 36, 37, 38, 39], which implements SGG within an end-to-end framework (also exemplified in other relation detection tasks [40, 41, 42]), discarding several hand-crafted procedures; **v)** Open-vocabulary SGG, which learns to recognize unseen categories during training by using category-level [2, 8, 9, 3, 10] or part-level prompts [12]. **vi)** Few-show SGG, which learns to recognize relations given a few examples [43].

Existing OVSGG frameworks adopt a standard open-vocabulary learning paradigm, *i.e.*, perform vision-language alignment in the pre-trained or random initialized semantic space with supervision of only the category names. One except [12] reformulates OVSGG from recognizing category-level prompts into recognizing part-level prompts, by decomposing SGG into several separate components and computing their similarities independently, in a training-free manner. SDSGG represents the best of both worlds. On the one hand, we point out the drawbacks of the commonly used *scene-agnostic* text classifiers and introduce *scene-specific* alternates to understand scenes from different perspectives. On the other hand, SDSGG incorporates a learnable mutual visual adapter to capture the underlying patterns in the dataset and proposes to renormalize text classifiers for adapting to specific contexts.

Open-vocabulary Learning. Most deep neural networks operate on the close-set assumption, which can only identify pre-defined categories that are present in the training set. Early zero-shot learning approaches [44, 45, 46] adopt word embedding projection to constitute the classifiers for

unseen class classification. With the rapid progress of vision language pre-training [4, 5, 47], open vocabulary learning [48] has been proposed and demonstrates impressive capabilities by, for example, distilling knowledge from VLMs [49, 50, 51, 52, 53, 54], exploiting caption data [55, 56], generating pseudo labels [57, 58, 59, 60, 61], training without dense labels [62, 63, 64], joint learning of several tasks [65, 66, 67], and training with more balanced data [68, 69].

While sharing a very high-level idea of vision-language alignment in open-vocabulary methods, our SDSGG **i**) explicitly models the context-dependent scenarios and introduces *scene-specific* text classifiers as the flexible learning targets, and **ii**) incoreporates a new mechanism for computing *self-normalized* similarities, thereby renormalizing text classifiers according to the presented image.

VLMs Meet LLMs [70]. The big win for VLMs has been all about getting the model to match up pictures and their descriptions closely while keeping the mismatched ones apart [4, 5, 47]. This trick, inspired by contrastive learning from self-supervised learning [71, 72, 73, 74, 75], helps VLMs get really good at figuring out what text goes with what image. Moreover, prompt learning acts as a flexible way to communicate with VLMs, giving them a nudge or context to apply their knowledge of images and text in specific ways [6, 76, 77, 7, 78]. In addition to hand-crafted or learnable prompts, [11] offers a fresh perspective, *i.e.*, using LLMs to generate detailed, comprehensive prompts as the inputs of VLMs' text encoder. Many follow-up works [79, 80, 81, 82, 83, 84, 85] across various domains and tasks demonstrate the effectiveness of integrating VLMs and LLMs.

Category-/part-level prompts are *scene-agnostic* and cannot adapt to specific contexts. To this end, SDSGG adopts *scene-specific* descriptions, generated by LLMs in a multi-persona collaboration fashion, as the inputs of CLIP's text encoder. Different from part-level prompt based approaches [11, 12] which processes all part-level prompts as affirmative classifiers, SDSGG provides a flexible alternative via the association between classifiers (*i.e.*, SSDs) and categories, and the renormalizing strategy *w.r.t.* each SSD. Since the learned semantic space of VLMs may not be sensitive to relations [12], we design a lightweight mutual visual adapter to project them into interaction-aware space for capturing the complicated interplay between the subject and object.

3   Methodology

Task Setup and Notations. Given an image I, SGG transforms it into a structured representation, *i.e.*, a directed graph $\mathcal{G} = \{\mathcal{O}, \mathcal{R}\}$, where $\mathcal{O}$ represents localized (*i.e.*, bounding box) objects with object category information and $\mathcal{R}$ represents pairwise relations between objects. For a fair comparison, this work focuses on predicting $\mathcal{R}$ given $\mathcal{O}$, *i.e.*, the predicate classification task which avoids the noise from object detection, as suggested by [1, 3, 12]. Our study delves into the intricacies of transitioning SGG from a traditional closed-set setting to an open vocabulary paradigm. This transition enables the system to recognize previously unseen predicate categories (*i.e.*, `novel` split) by learning from observed predicate categories (*i.e.*, `base` split) during training.

SDSGG follows the standard zero-shot pipeline with VLMs [4], which computes similarity between the visual embedding v and the text embedding t for each category, and the category with highest similarity is viewed as the final classification result (*c.f*. Fig. 2a). For each subject-object pair, v can be derived by feeding cropped patches from the input image I into the visual encoder. The text embedding t used in existing OVSGG frameworks falls into two main settings: **i**) Each category consists of only one text classifier, *i.e.*, the category name itself. **ii**) Each category consists of multiple text classifiers w.r.t. subject and object, *i.e.*, part-level descriptions. SDSGG reformulates the text classifiers as *scene-specific* descriptions which will be detailed in §3.1.

Algorithmic Overview. SDSGG is a SSD based framework for OVSGG, supported by the cooperation of VLMs and LLMs. For the textual part (*c.f*. Fig. 2b), SDSSG enjoys the expressive range of the comprehensive SSDs generated by LLMs' multi-persona collaboration. This is complemented by a renormalizing mechanism to adjust the influence of each text classifier. For the visual part (*c.f*. Fig. 2c), SDSSG is equipped with a mutual visual adapter to aggregate visual features v from I for a given subject-object pair. After introducing how we generate and use SSDs for the text part (§3.1) and the mutual visual adapter for interplay modeling of the subject and object (§3.2), we will elaborate on SDSGG's training objective (§3.3).
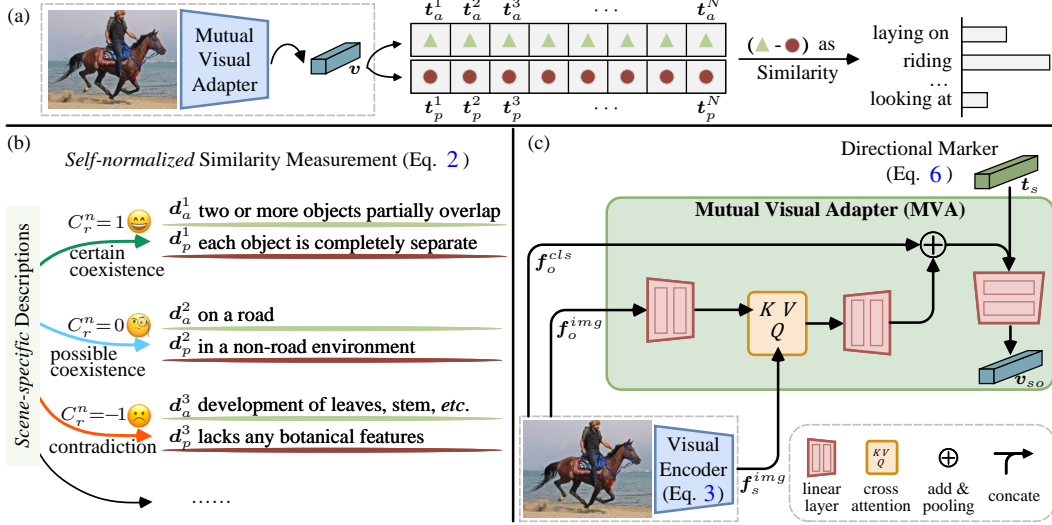
Figure 2: (a) Overview of SDSGG. (b) Each text classifier of SDSGG contains a raw description d_a^n and an opposite description d_p^n. As such, the *self-normalized* similarities can be computed with the association (C_r^n) between predicate categories and SSDs. (c) Given the visual features (*i.e.*, f_s^{img}, f_o^{cls}, and f_o^{img}) of both the subject and object extracted from CLIP's visual encoder, our mutual visual adapter (MVA) projects them into interaction-aware space and models their complicated interplay with cross-attention.

3.1 Scene-specific Text Classifiers

Scene-level Description Generation via Multi-persona Collaboration. Using standard prompts to query LLMs is a direct way to generate scene descriptions. For instance, one could straightforwardly prompt LLM with a question like "Imagine there is an animal that is eating, what should the scene look like?". LLM's response would typically sketch out an envisioned scene based on its statistical training on large corpus. However, these responses may not fully capture the scene's complexity, often overlooking aspects such as the spatial arrangement of elements and the background environment.

To alleviate this, we draw inspiration from recent advances in LLMs' multi-persona capabilities [86, 87, 88, 89]. Specifically, LLM adopts three distinct roles, mirroring the expertise found in experts specializing in biology, physics, and engineering. This approach allows for a comprehensive discussion of what a given scene entails. Because each query to LLM usually only yields 3-5 sentences of description, we query LLM several times, each time giving LLM a different scene content to be discussed, thus obtaining a large number of scene descriptions. Since these initial descriptions may suffer from noise and semantic overlap, we ask LLM to streamline and combine these descriptions, ensuring more cohesive and distinct scene-level descriptions $\mathcal{D}_l = \{d^1, d^2, \cdots, d^N\}$ and corresponding text embeddings $\mathcal{T} = \{t^1, t^2, \cdots, t^N\}$, where N denotes the number of SSDs and text embeddings $\mathcal{T}$ are extracted by the text encoder of CLIP. Due to the limited space, we provide more details and prompts for generating scene descriptions in the appendix (§D).

Association between Scene-level Descriptions and Relation Categories. So far, we have obtained various scene-level descriptions that have the ability to represent diverse scenes. A critical inquiry arises regarding their utility for relation detection, given their lack of explicit association with specific relation categories. To address this, we delineate three distinct scenarios characterizing the interplay between relation categories and scene descriptions: **i**) certain coexistence ($C_r^n = 1$), where a direct correlation exists; **ii**) possible coexistence ($C_r^n = 0$), indicating a potential but not guaranteed association; and **iii**) contradiction ($C_r^n = -1$), denoting an incompatibility between the scene description and relation category. Here C_r^n denotes the correlation between relation $r \in \mathcal{R}$ and n_{th} scene description, and is generated by LLMs (prompts are shown in §D). Such a categorization enables us to calculate the similarity for each relation category:

$$sim(\boldsymbol{v}, r) = \sum_{n=1}^N C_r^n * \langle \boldsymbol{v}, \boldsymbol{t}^n \rangle, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity with temperature [4].

5

Scene-specific Descriptions = Scene-level Descriptions + Reweighing. Upon examining text classifiers in depth, we noticed that certain classifiers are contextually bound, *e.g.*, "two or more objects partially overlap each other" may not exist in all scenes. This observation underscores the necessity for a mechanism to evaluate the significance of each text classifier, rather than applying a uniform weight across the board. This is exactly what the term "*specific*" means. Recall that the similarity measurement between an image and the text "a photo of a cat" alone yields limited insight. However, when juxtaposed with multiple texts, such as "a photo of a cat/dog/tiger", the comparison of similarity scores across these categories reveals which category (cat, dog, or tiger) the image most closely resembles. Inspired by this, we propose the incorporation of an opposite description d_p^n (§D) for each raw scene description d_a^n as a reference point (*e.g.*, "two or more objects partially overlap each other" *vs.* "each object is completely separate with clear space between them"), resulting in SSDs $\mathcal{D}_s = \{(d_a^1, d_p^1), (d_a^2, d_p^2), \cdots, (d_a^N, d_p^N)\}$ and updated text embeddings $\mathcal{T} = \{(\boldsymbol{t}_a^1, \boldsymbol{t}_p^1), (\boldsymbol{t}_a^2, \boldsymbol{t}_p^2), \cdots, (\boldsymbol{t}_a^N, \boldsymbol{t}_p^N)\}$. Subsequently, the *self-normalized* similarity is defined as:

$$sim(\boldsymbol{v}, r) = \sum_{n=1}^{N} C_r^n * (\langle \boldsymbol{v}, \boldsymbol{t}_a^n \rangle - \langle \boldsymbol{v}, \boldsymbol{t}_p^n \rangle). \tag{2}$$

The difference in similarity scores, *i.e.*, $\langle \boldsymbol{v}, \boldsymbol{t}_a^n \rangle - \langle \boldsymbol{v}, \boldsymbol{t}_p^n \rangle$, quantifies the relative contribution of that SSD. By such means, a SSD irrelevant to the presented context will have a minimal effect, as the similarity scores of it ($\langle \boldsymbol{v}, \boldsymbol{t}_a^n \rangle$) and its opposite ($\langle \boldsymbol{v}, \boldsymbol{t}_p^n \rangle$) would be nearly identical.

3.2 Mutual Visual Adapter

After introducing how to obtain the text embeddings and how to compute vision-language similarity, one question remains at this point: how to obtain visual embeddings? When given a subject-object pair with bounding boxes from $\mathcal{O}$, there exist various strategies for aggregating visual features for the subject and object within $\boldsymbol{I}$. For example, traditional closed-set SGG frameworks [16, 17] employ RoI pooling to extract visual features for specified bounding boxes, subsequently fusing these features for further classification. In contrast, the more recent OVSGG framework [12] uses the visual encoder of CLIP to extract visual embeddings of both subject and object. Then, it processes two visual embeddings *independently* through part-level descriptions. Such an independent approach, however, overlooks the informative interplay between the subject and object.

To address this oversight and capture the complicated interactions between subject and object, we introduce a new component: the mutual visual adapter (MVA). MVA is composed of several lightweight, learnable modules designed to fine-tune CLIP's visual encoder specifically for pairwise relation detection. This approach aims to enhance the model's ability to recognize the nuanced interactions that define relationships between subject and object in an image.

Regional Encoder. Given an image $\boldsymbol{I}$ and a subject-object pair with bounding boxes (b_s and b_o) from $\mathcal{O}$, the initial visual embeddings can be obtained from the visual encoder of CLIP:

$$\boldsymbol{f}_{s/o} = [\boldsymbol{f}_{s/o}^{cls} | \boldsymbol{f}_{s/o}^{img}] = [\boldsymbol{f}_{s/o}^{cls} | \boldsymbol{f}_{s/o}^1, \boldsymbol{f}_{s/o}^2, \cdots, \boldsymbol{f}_{s/o}^M] = \texttt{Encoder}_v(\texttt{Crop}(\boldsymbol{I}, b_{s/o})), \tag{3}$$

where M denotes the number of patches, $\texttt{Encoder}_v$ is CLIP's visual encoder that is kept frozen during training, and $\texttt{Crop}$ represents image cropping.

Visual Aggregator. Next, MVA is adopted to aggregate $\boldsymbol{f}_s$ and $\boldsymbol{f}_o$ by cross-attention and two lightweight projection modules. Let the subject part be the query, and the object part be the key and value. The patch embeddings of object $\boldsymbol{f}_o^{img}$ are first projected into low-dimensional, semantic space:

$$\boldsymbol{l}_o^{img} = \texttt{Linear}_{down}(\boldsymbol{f}_o^{img}), \tag{4}$$

where $\texttt{Linear}$ denotes a standard fully connected layer. Afterwards, cross-attention is adopted to capture the complicated interplay between subject and object, resulting in an aggregated visual embedding for the given subject-object pair:

$$\boldsymbol{v}_{so} = \texttt{Linear}_{up}(\texttt{AvgPool}(\texttt{LN}(\boldsymbol{f}_o^{cls} + \texttt{CrossAttn}(\boldsymbol{f}_s^{img}, \boldsymbol{l}_o^{img})))), \tag{5}$$

where $\texttt{AvgPool}$ is the average pooling. $\texttt{LN}$ is the standard layer normarlization. $\texttt{CrossAttn}(\boldsymbol{Q}, \boldsymbol{KV})$ denotes the standard cross-attention operation. $\boldsymbol{v}_{os}$ can be computed in a similar way by exchanging the query and key of cross-attention. Combining them together leads to the final visual embedding $\boldsymbol{v} = (\boldsymbol{v}_{so} + \boldsymbol{v}_{os})/2$ for final similarity measurement. As such, MVA captures the interplay of subject and object in the projected, interaction-aware space.

Directional Marker. One may notice that the structure of MVA is *symmetric* and has no information about which input branch is the subject/object. This has a relatively small effect on semantic relations, but a significant effect on geometric relations. For instance, after exchanging the location of the subject and object (image flipping), the relation "eating" remains unchanged, while the relation "on the left" would become "on the right". Here we simply incorporate two text embeddings (t_s and t_o) of "a photo of subject/object" into MVA and thus update the visual embedding as:

$$v_{so} = \texttt{Linear}\big(\texttt{Concate}(v_{so}, t_s)\big), \tag{6}$$

where $\texttt{Concate}$ denotes concatenation. v_{os} and v can be updated accordingly. Further exploration of directional marker, *e.g.*, incorporating more complex feature fusion modules, is left for future work.

3.3 Training Objective

A typical training objective in open-vocabulary learning aims to bring representations of positive pairs closer and push negative pairs apart in the embedding space. In SDSGG, the term "positive/negative pairs" is not defined at the category level *but at the description level*, requiring losses tailored for different relation-description association types. Given a labeled relation triplet, one simplest contrastive loss can be defined as:

$$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_{t^n \in \mathcal{T}} \big(\underbrace{\langle v, t_a^n \rangle - \langle v, t_p^n \rangle}_{similarity} - \underbrace{\alpha * C_r^n}_{target} \big)^2, \tag{7}$$

where α is a scaling factor. However, as for scene descriptions marked by possible coexistence (*i.e.*, $C_r^n = 0$), there is no direct target that can be used for training. Inspired by the identical mapping in residual learning [90], we make the prediction results of MVA close to those of CLIP. As such, MVA can learn the implicit knowledge embedded in CLIP's semantic space. In addition, this regularization term prevents MVA from overfitting to relations in the $\texttt{base}$ split, which is a common problem in open-vocabulary learning. Hence, the loss is further reformulated as:

$$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_{t^n \in \mathcal{T}} \big(\underbrace{\langle v, t_a^n \rangle - \langle v, t_p^n \rangle}_{similarity} - \underbrace{\alpha * C_r^n}_{target} - \underbrace{(\beta * sim_{CLIP}(I, rel) - \lambda)}_{margin} \big)^2, \tag{8}$$

where β is another scaling factor, $sim_{CLIP}(I, rel)$ denotes the vision-language similarity derived from the original CLIP, and λ is a constant scalar and is empirically set to 3e-2.

4 Experiment

4.1 Experimental Setup

Dataset. We evaluate our method on GQA [15] and VG [14] following [3, 12].

Split. Following previous work [3], VG is divided into two splits: $\texttt{base}$ and $\texttt{novel}$ split. The $\texttt{base}$ split comprises 70% of the relation categories for training, while the $\texttt{novel}$ split contains the remaining 30% categories invisible during training. For a more comprehensive comparison, we also conduct testing on the $\texttt{semantic}$ set, encompassing 24 predicate categories [16, 12] with richer semantics. $\texttt{base}$ and $\texttt{novel}$ split of GQA [15] are obtained in a similar manner (§B).

Evaluation Metrics. We report Recall@K (R@K) and Recall@K(mR@K) following [3, 22].

Base Models and Competitors. As for the $\texttt{base}$ and $\texttt{novel}$ split, we compare SDSGG with two baselines: **i**) CLS [4], which uses only the category-level prompts to compute the similarity between the image and text; and **ii**) Epic [3], a latest OVSGG method, which introduces an entangled cross-modal prompt approach and learns the cross-modal embeddings using contrastive learning. In terms of the $\texttt{semantic}$ split, we compare our SDSGG with three baselines: **i**) CLS [4], which uses only the category-level prompts; **ii**) CLSDE [12], which uses prompts of relation class description; and **iii**) RECODE [12], which uses visual cues of several separate components. Since [10] has neither released the detailed split nor the code, it is not included in the comparisons for fairness.

Implementation Details. Due to limited space, implementation details are left in the appendix (§B).

4.2 Quantitative Comparison Result

We conduct quantitative experiments on VG [14] and GQA [15]. To ensure the performance gain is reliable, each experiment is repeated three times. The average and standard deviation are reported.

Table 1: Quantitative results (§4.2) on VG [14] `base` and `novel`.

Method	Split	R@20↑	R@50↑	R@100↑	mR@20↑	mR@50↑	mR@100↑
CLS[ICML21][4]		2.1	3.2	3.9	7.0	9.0	10.9
Epic[ICCV23][3]	`base`	-	22.6	27.2	-	-	-
Ours		$18.7_{\pm 0.69}$	$26.5_{\pm 0.92}$	$31.6_{\pm 1.00}$	$9.2_{\pm 0.14}$	$12.4_{\pm 0.12}$	$14.8_{\pm 0.10}$
CLS[ICML21][4]		13.2	18.1	22.2	11.5	17.9	23.8
Epic[ICCV23][3]	`novel`	-	7.4	9.7	-	-	-
Ours		$18.4_{\pm 0.53}$	$25.4_{\pm 0.48}$	$29.6_{\pm 0.42}$	$17.1_{\pm 0.42}$	$25.2_{\pm 0.95}$	$31.2_{\pm 1.09}$

Table 2: Quantitative results (§4.2) on VG [14] `semantic`.

Method	R@20↑	R@50↑	R@100↑	mR@20↑	mR@50↑	mR@100↑
CLS[ICML21][4]	7.2	10.9	13.2	9.4	14.0	17.6
CLSDE[NeurIPS23][12]	7.0	10.6	12.9	8.5	13.6	16.9
RECODE†[NeurIPS23][12]	7.3	11.2	15.4	8.2	13.5	18.3
RECODE[NeurIPS23][12]	9.7	14.9	19.3	10.2	16.4	22.7
RECODE⋆[NeurIPS23][12]	10.6	18.3	25.0	10.7	18.7	27.8
Ours	$21.5_{\pm 0.47}$	$29.3_{\pm 0.53}$	$34.9_{\pm 0.66}$	$16.8_{\pm 0.08}$	$22.7_{\pm 0.41}$	$28.4_{\pm 0.67}$

VG [14] `base` and `novel`. Table 1 illustrates our compelling results over the latest OVSGG model, Epic [3]. Since Epic did not release the full code, we report values under the same setting as in the original paper. CLS [4] leverages the original CLIP's checkpoint (*w/o* fine-tuning) and only the category-level prompts. The performance of CLS on the `novel` split is better than that on the `base` split because **i**) the `base` split contains more relation candidates to be classified (35 *vs.* 15); and **ii**) the `base` split contains head, uninformative relations (*e.g.*, "on" and "has") which are hard to be distinguished by CLIP [12]. By perform contrastive learning on the `base` split to learn the entangled cross-modal prompt, Epic [3] achieves significant performance gains on the `base` split (*i.e.*, 22.6% *vs.* 3.2% R@50 and 27.2% *vs.* 3.9% R@100). However, Epic demonstrates worse performance on the `novel` split (*i.e.*, 7.4% *vs.* 18.1% R@50 and 9.7% *vs.* 22.2% R@100), which indicates its overfitting on the training data. By incorporating SSDs, SDSGG outperforms Epic by a large margin on both the `base` split and `novel` split. For instance, SDSGG exceeds Epic by **3.9%/4.4%** R@50/100 on the `base` split and **18.0%/19.9%** R@50/100 on the `novel` split, respectively. The significant performance improvements on the `novel` set demonstrate the strong zero-shot capability of our approach. In addition, our margins over the CLS are **16.6%∼27.7%** R@K and **2.2%∼3.9%** mR@K on the `base` split, and **5.2%∼7.3%** R@K and **5.6%∼7.4%** mR@K on the `novel` split, respectively.

VG [14] `semantic`. In Table 2, we present the numerical results of SDSGG against the latest OVSGG work [12] on the `semantic` split. By leveraging part-level prompts, RECODE demonstrates superior performance compared to CLS [4] and CLSDE [12]. The introduction of filtering strategies, *i.e.*, RECODE⋆, shows substantial improvements. As seen, SDSGG surpasses all counterparts with remarkable gains on all metrics. In particular, SDSGG exceeds RECODE⋆ by **10.9%/11.0%/9.9%** on R@20/50/100, and **6.1%/4.0%/0.6%** on mR@20/50/100. Notably, SDSGG achieves impressive performance without relying on additional augmentation or data [10]. Since an increase on mR@K implies the average of the improvements for all categories, yields on mR@K may be relatively lower than that on R@K. Without bells and whistles, SDSGG establishes a new *state-of-the-art*.

GQA [15] `base` and `novel`. Since the codes of Epic [3] and RECODE [12] are insufficient to support replication, we only compare SDSGG with CLS on GQA [15]. As shown in Table 3, SDSGG outperforms CLS by a large margin across all splits and metrics. Since CLS is not trained on the `base` split, its performance on R@K is rel-

Table 3: Quantitative results on GQA [14] `base` & `novel`.

Method	Split	R@20/50/100	mR@20/50/100
CLS [4]	`base`	4.2 / 6.4 / 7.9	8.9 / 13.2 / 15.3
Ours		**33.4 / 43.9 / 49.5**	**15.6 / 21.0 / 24.5**
		$\pm 1.10 / \pm 1.21 / \pm 1.20$	$\pm 0.47 / \pm 0.38 / \pm 0.79$
CLS [4]	`novel`	21.3 / 28.3 / 32.1	16.6 / 27.0 / 29.4
Ours		**27.2 / 37.4 / 42.9**	**23.8 / 32.8 / 37.3**
		$\pm 0.35 / \pm 0.58 / \pm 0.33$	$\pm 0.21 / \pm 1.35 / \pm 1.33$

atively low due to the massive annotations of geometric relations that are not semantically rich [12].

Taking together, our extensive results provide solid evidence that SDSGG successfully unlocks the power of LLMs in OVSGG, and SSD is a promising alternative to category-/part-level prompts.

Figure 3: Visual results (§4.3) on VG [14].

4.3 Qualitative Comparison Result

Fig. 3 visualizes qualitative comparisons of SDSGG against CLIP [4] on VG [14]. As seen, with the proposed SSDs, SDSGG can generate higher-quality relation predictions even in challenging scenarios. We respectfully refer the reviewer to the appendix (§E) for more qualitative comparisons.

4.4 Diagnostic Experiment

For thorough evaluation, we conduct a series of ablative studies on VG [14].

Textual Part. We first study the effectiveness of our multi-persona collaboration (MPC) for *scene-specific* description generation (§3.1) in Table 4. Here we use the standard prompts to query LLMs for generating descriptions. As seen, without MPC, the performance drops drastically,

Table 4: Ablation studies (§4.4) on MPC.

Method	Split	R@20/50/100	mR@20/50/100
Ours	base	**18.7 / 26.6 / 31.6**	**9.1 / 12.3 / 14.7**
w/o MPC		6.0 / 9.3 / 11.8	2.5 / 5.1 / 7.6
Ours	novel	**18.9 / 25.8 / 30.0**	**16.6 / 25.2 / 31.5**
w/o MPC		5.2 / 8.7 / 11.8	4.9 / 11.0 / 16.2

e.g., 11.8%/11.8% *vs.* 31.6%/30.0% R@100 on the base/novel split, respectively. This indicates the importance of the text classifiers used as they have impact on both training and testing.

While the effectiveness of MPC has been validated, one may wonder: **i**) Why use these three roles? **ii**) How to ensure the completeness and quality of generated classifiers? We want to highlight that: **i**) Different roles are used to increase the variety of descriptions. There is no word on exactly which roles should be used. **ii**) These open problems are beyond the scope of this work. We leave them for future work. **iii**) This work makes the first attempt to enhance classifier generation for OVSGG via MPC. **iv**) The experimental results suggest that the current generated SSDs are *good enough* for commonly used relation categories. To provide more empirical results, we investigate the impact of the proposed *self-normalized* similarities and the number of used SSDs in the appendix (§A).

Visual Part. Then, we examine the impact of mutual visual adapter (MVA, §3.2) and directional marker (DM, §3.2) in Table 5. The 1st row denotes a strong baseline, *i.e.*, a multi-layer perceptron with comparable parameters to aggregate the visual features. Upon projecting the visual features into interaction-aware space and applying cross-

Table 5: Ablation studies (§4.4) on the visual part.

MVA	DM	Split	R@20/50/100	mR@20/50/100
		base	13.1 / 18.7 / 22.4	8.6 / 11.6 / 13.8
✓			17.0 / 23.8 / 28.2	**9.2** / 12.3 / 14.6
✓	✓		**18.7 / 26.6 / 31.6**	9.1 / **12.3 / 14.7**
		novel	17.4 / 24.7 / 28.9	17.2 / **26.5** / 30.9
✓			18.6 / 25.1 / 28.7	**17.4** / 25.0 / 31.0
✓	✓		**18.9 / 25.8 / 30.0**	16.6 / 25.2 / **31.5**

attention, we observe consistent and substantial improvements for both R@K and mR@K on both the base and novel split. These results demonstrate the efficacy of our adapter and the necessity of incorporating cross-attention for capturing the complicated interplay between subjects and objects. Since DM is designed for geometric relations with massive annotations [21, 24], the improvements on R@K are considerable, while those on mR@K are relatively small. See §A for more results.

5 Conclusion

This work presents SDSGG, a *scene-specific* description based framework for OVSGG. Despite the previous works based upon category-/part-level prompts, we argue that these text classifiers are

scene-agnostic, which cannot adapt to specific contexts and may even be misleading. To address this, we carefully design a multi-persona collaboration strategy for generating flexible, context-aware SSDs, a *self-normalized* similarity computation module for renormalizing the influence of each SSD, and a mutual visual adapter that consists of several trainable lightweight modules for learning interaction-aware space. Our approach distinguishes itself by using SSDs derived from LLMs, which are tailored to the content of the presented image. This is further enhanced by MVA, which captures the underlying interaction patterns based on the semantic space of VLMs. We expect the introduction of SDSGG will not only set a new benchmark for OVSGG, but also encourage the community to explore the potential of integrating LLMs with VLMs for deeper, contextual understanding of images.

References

[1] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 5410–5419, 2017.

[2] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Towards open-vocabulary scene graph generation with prompt-based finetuning. In *ECCV*, pages 56–73, 2022.

[3] Qifan Yu, Juncheng Li, Yu Wu, Siliang Tang, Wei Ji, and Yueting Zhuang. Visually-prompted language model for fine-grained scene graph generation in an open world. In *ICCV*, 2023.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[5] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pages 12888–12900, 2022.

[6] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[7] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022.

[8] Kaifeng Gao, Long Chen, Hanwang Zhang, Jun Xiao, and Qianru Sun. Compositional prompt tuning with motion cues for open-vocabulary video relation detection. In *ICLR*, 2023.

[9] Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang-Wen Chen. Learning to generate language-supervised and open-vocabulary scene graph using pre-trained visual-semantic space. In *CVPR*, pages 2915–2924, 2023.

[10] Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Changwen Chen. Expanding scene graph boundaries: Fully open-vocabulary scene graph generation via visual-concept alignment and retention. *arXiv preprint arXiv:2311.10988*, 2023.

[11] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *ICLR*, 2023.

[12] Lin Li, Jun Xiao, Guikun Chen, Jian Shao, Yueting Zhuang, and Long Chen. Zero-shot visual relation detection via composite visual cues from large language models. In *NeurIPS*, 2023.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[14] 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. *IJCV*, 123:32–73, 2017.

[15] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019.

[16] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018.

[17] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, pages 6619–6628, 2019.

[18] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, pages 3746–3753, 2020.

[19] Xin Lin, Changxing Ding, Jing Zhang, Yibing Zhan, and Dacheng Tao. Ru-net: Regularized unrolling network for scene graph generation. In *CVPR*, pages 19457–19466, 2022.

[20] Chaofan Zheng, Xinyu Lyu, Lianli Gao, Bo Dai, and Jingkuan Song. Prototype-based embedding network for scene graph generation. In *CVPR*, pages 22783–22792, 2023.

[21] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, pages 3716–3725, 2020.

[22] Lin Li, Guikun Chen, Jun Xiao, Yi Yang, Chunping Wang, and Long Chen. Compositional feature augmentation for unbiased scene graph generation. In *ICCV*, pages 21685–21695, 2023.

[23] Lin Li, Jun Xiao, Hanrong Shi, Wenxiao Wang, Jian Shao, An-An Liu, Yi Yang, and Long Chen. Label semantic knowledge distillation for unbiased scene graph generation. *IEEE TCSVT*, 2023.

[24] Lin Li, Long Chen, Yifeng Huang, Zhimeng Zhang, Songyang Zhang, and Jun Xiao. The devil is in the labels: Noisy label correction for robust scene graph generation. In *CVPR*, pages 18869–18878, 2022.

[25] Xingchen Li, Long Chen, Jian Shao, Shaoning Xiao, Songyang Zhang, and Jun Xiao. Rethinking the evaluation of unbiased scene graph generation. In *BMVC*, 2023.

[26] Guikun Chen, Lin Li, Yawei Luo, and Jun Xiao. Addressing predicate overlap in scene graph generation with semantic granularity controller. In *ICME*, 2023.

[27] Lin Li, Jun Xiao, Hanrong Shi, Hanwang Zhang, Yi Yang, Wei Liu, and Long Chen. Nicest: Noisy label correction and training for robust scene graph generation. *IEEE TPAMI*, 2024.

[28] Hanrong Shi, Lin Li, Jun Xiao, Yueting Zhuang, and Long Chen. From easy to hard: Learning curricular shape-aware features for robust panoptic scene graph generation. *IJCV*, pages 1–20, 2024.

[29] Jing Shi, Yiwu Zhong, Ning Xu, Yin Li, and Chenliang Xu. A simple baseline for weakly-supervised scene graph generation. In *ICCV*, pages 16393–16402, 2021.

[30] Xingchen Li, Long Chen, Wenbo Ma, Yi Yang, and Jun Xiao. Integrating object-aware and interaction-aware knowledge for weakly supervised scene graph generation. In *ACM MM*, pages 4204–4213, 2022.

[31] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In *CVPR*, pages 3736–3745, 2020.

[32] Keren Ye and Adriana Kovashka. Linguistic structures as weak supervision for visual scene graph generation. In *CVPR*, pages 8289–8299, 2021.

[33] Yuan Yao, Ao Zhang, Xu Han, Mengdi Li, Cornelius Weber, Zhiyuan Liu, Stefan Wermter, and Maosong Sun. Visual distant supervision for scene graph generation. In *ICCV*, pages 15816–15826, 2021.

[34] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Toward a unified transformer-based framework for scene graph generation and human-object interaction detection. *IEEE TIP*, 32:6274–6288, 2023.

[35] Yao Teng and Limin Wang. Structured sparse r-cnn for direct scene graph generation. In *CVPR*, pages 19437–19446, 2022.

[36] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE TPAMI*, 2023.

[37] Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *CVPR*, pages 11546–11556, 2021.

[38] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *CVPR*, pages 19486–19496, 2022.

[39] Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Hydra-sgg: Hybrid relation assignment for one-stage scene graph generation. *arXiv preprint arXiv:2409.10262*, 2024.

[40] Liulei Li, Jianan Wei, Wenguan Wang, and Yi Yang. Neural-logic human-object interaction detection. In *NeurIPS*, 2023.

[41] Liulei Li, Wenguan Wang, and Yi Yang. Human-object interaction detection collaborated with large relation-driven diffusion models. In *NeurIPS*, 2024.

[42] Jianan Wei, Tianfei Zhou, Yi Yang, and Wenguan Wang. Nonverbal interaction detection. In *ECCV*, 2024.

[43] Xingchen Li, Jun Xiao, Guikun Chen, Yinfu Feng, Yi Yang, An-An Liu, and Long Chen. Decomposed prototype learning for few-shot scene graph generation. *ACM TOMM*, 2023.

[44] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, pages 69–77, 2016.

[45] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, pages 384–400, 2018.

[46] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, pages 2152–2161, 2015.

[47] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[48] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, et al. Towards open vocabulary learning: A survey. *IEEE TPAMI*, 2024.

[49] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022.

[50] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *CVPR*, pages 11583–11592, 2022.

[51] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *CVPR*, pages 15254–15264, 2023.

[52] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *CVPR*, pages 1179–1189, 2023.

[53] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, pages 815–824, 2023.

[54] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022.

[55] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, pages 14393–14402, 2021.

[56] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, pages 540–557, 2022.

[57] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *CVPR*, pages 7010–7019, 2023.

[58] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. In *CVPR*, pages 7020–7031, 2022.

[59] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, pages 696–712, 2022.

[60] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, pages 7061–7070, 2023.

[61] Mingfei Gao, Chen Xing, Juan Carlos Niebles, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. Open vocabulary object detection with pseudo bounding-box labels. In *ECCV*, pages 266–282, 2022.

[62] Vibashan VS, Ning Yu, Chen Xing, Can Qin, Mingfei Gao, Juan Carlos Niebles, Vishal M Patel, and Ran Xu. Mask-free ovis: Open-vocabulary instance segmentation without manual mask annotations. In *CVPR*, pages 23539–23549, 2023.

[63] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, pages 18134–18144, 2022.

[64] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *ICML*, pages 23033–23044, 2023.

[65] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *ICCV*, pages 1020–1031, 2023.

[66] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *CVPR*, pages 15116–15127, 2023.

[67] Jie Qin, Jie Wu, Pengxiang Yan, Ming Li, Ren Yuxi, Xuefeng Xiao, Yitong Wang, Rui Wang, Shilei Wen, Xin Pan, et al. Freeseg: Unified, universal and open-vocabulary image segmentation. In *CVPR*, pages 19446–19455, 2023.

[68] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, pages 350–368, 2022.

[69] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *NeurIPS*, 2024.

[70] Wenguan Wang, Yi Yang, and Yunhe Pan. Visual knowledge in the big model era: Retrospect and prospect. *FITEE*, 2024.

[71] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020.

[72] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[73] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, pages 22243–22255, 2020.

[74] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9620–9629, 2021.

[75] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE TPAMI*, 43(11):4037–4058, 2020.

[76] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *CVPR*, pages 4953–4963, 2022.

[77] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727, 2022.

[78] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, pages 19113–19122, 2023.

[79] Mayug Maniparambil, Chris Vorster, Derek Molloy, Noel Murphy, Kevin McGuinness, and Noel E O'Connor. Enhancing clip with gpt-4: Harnessing visual descriptions as prompts. In *ICCV*, pages 262–271, 2023.

[80] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, pages 15691–15701, 2023.

[81] Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. In-context impersonation reveals large language models' strengths and biases. In *NeurIPS*, volume 36, 2024.

[82] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *CVPR*, pages 15211–15222, 2023.

[83] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *CVPR*, pages 19187–19197, 2023.

[84] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *ICCV*, pages 3090–3100, 2023.

[85] Lin Li, Guikun Chen, Jun Xiao, and Long Chen. Compositional zero-shot learning via progressive language-based observations. *arXiv preprint arXiv:2311.14749*, 2023.

[86] Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*, 2023.

[87] Haoyang Liu, Yijiang Li, Jinglin Jian, Yuxuan Cheng, Jianrong Lu, Shuyi Guo, Jinglei Zhu, Mianchen Zhang, Miantong Zhang, and Haohan Wang. Toward a team of ai-made scientists for scientific discovery from gene expression data. *arXiv preprint arXiv:2402.12391*, 2024.

[88] Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of EMNLP*, pages 2550–2575, 2023.

[89] Yu Shang, Yu Li, Fengli Xu, and Yong Li. Defint: A default-interventionist framework for efficient reasoning with hybrid large language models. *arXiv preprint arXiv:2402.02563*, 2024.

[90] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[91] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[92] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *ICML*, pages 9099–9117, 2022.

[93] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, pages 357–366, 2021.

[94] Wenxiao Wang, Wei Chen, Qibo Qiu, Long Chen, Boxi Wu, Binbin Lin, Xiaofei He, and Wei Liu. Crossformer++: A versatile vision transformer hinging on cross-scale attention. *IEEE TPAMI*, 2023.

[95] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. In *ICLR*, 2023.

[96] Yaohua Zha, Jinpeng Wang, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In *ICCV*, pages 14161–14170, 2023.

[97] Lifeng Fan, Wenguan Wang, Siyuan Huang, Xinyu Tang, and Song-Chun Zhu. Understanding human gaze communication by spatio-temporal graph reasoning. In *ICCV*, pages 5724–5733, 2019.

[98] Jiaming Lei, Lin Li, Chunping Wang, Jun Xiao, and Long Chen. Seeing beyond classes: Zero-shot grounded situation recognition via language explainer. In *ACM MM*, 2024.

Summary of the Appendix

For a better understanding of the main paper, we provide additional details in this supplementary material, which is organized as follows:

- §A introduces more ablative experiments.
- §B details the implementation details.
- §C provides the pseudo code of SDSGG.
- §D shows the generated *scene-specific* descriptions and the corresponding prompts.
- §E offers more qualitative results.
- §F discusses our limitations, societal impact, and directions of future work.
- §G presents more experimental results.

A   More Ablative Experiment

Self-normalized Similarity. Furthermore, we study the effectiveness of *self-normalized* similarity for evaluating the influence of each *scene-specific* description (§3.1) in Table S1. Here we remove the opposite description and use only the original scene description as the baseline. As seen, without the reference of opposite description, the performance drops drastically, *e.g.*, 28.3%/25.5%/28.4% *vs*. **31.6%/30.0%/35.1**% R@100 & 12.1%/29.8%/24.7% *vs*. **14.7%/31.5%/28.6**% mR@100 on the `base`/`novel`/`semantic` split, respectively. This indicates the importance of renormalizing similarity scores, and also validates our hypothesis – the *absolute* value of one single similarity yields only limited insight, while the *difference* of two similarities provides more information.

Table S1: Ablation studies (§A) on *self-normalized similarity* (SNS, §3.1).

Method	Split	R@20↑	R@50↑	R@100↑	mR@20↑	mR@50↑	mR@100↑
Ours	base	**18.7**	**26.6**	**31.6**	**9.1**	**12.3**	**14.7**
w/o SNS		16.1	23.3	28.3	7.0	10.0	12.1
Ours	novel	**18.9**	**25.8**	**30.0**	**16.6**	**25.2**	**31.5**
w/o SNS		14.7	20.6	25.5	14.2	21.5	29.8

Number of SSDs. We first study the impact of the number of used classifiers (*i.e.*, SSDs). The results, presented in Table S2, reveal a considerable performance enhancement when the number of pairs increased from 11 to 21, affirming the efficiency of SSDs. However, we also witnessed a dip in performance when too many SSDs were employed. This could be due to the unnecessary pairing of a relation category with an excessive number of descriptions (52 descriptions for 26 pairs).

Table S2: Ablation studies (§A) on the number of used classifiers (Eq. 2). The adopted hyperparameters are marked in red.

# Pairs	Split	R@20	R@50	R@100	mR@20	mR@50	mR@100
11		13.9	19.8	24.0	8.6	12.0	14.3
16	base	19.4	27.5	32.9	8.6	11.8	14.2
21		18.7	26.6	31.6	9.1	12.3	14.7
26		11.7	16.7	20.0	7.7	11.1	13.5
11		18.4	23.9	26.8	16.6	23.3	27.2
16	novel	17.9	24.4	28.7	16.4	23.2	28.7
21		18.9	25.8	30.0	16.6	25.2	31.5
26		12.4	18.5	23.7	14.3	21.8	27.2

Scaling Factors. We then study the effectiveness of the scaling factors used in our training objectives (§3.3) in Table S3. As seen, the performance remains consistent for those compared hyperparameters. This indicates the robustness of SDSGG to changes in the scale of the training targets. Considering the performance on all metrics together, α and β are set to be 2 and 1e-1, respectively.

Number of Attention Heads. We further investigate the impact of the number of attention heads H used in the mutual visual adapter (§3.2, Eq. 5). As shown in Table S4, the performance improves

15

Table S3: Ablation studies (§A) on the scaling factors used in the training objectives, *i.e.*, α and β (§3.3, Eq. 8). The adopted hyperparameters are marked in red.

α	β	Split	R@20↑	R@50↑	R@100↑	mR@20↑	mR@50↑	mR@100↑
1.5	1e-1		16.8	24.0	28.7	9.8	12.7	15.0
2.5	1e-1		19.4	27.3	32.4	9.4	12.6	14.9
2	3e-1	base	18.8	26.7	31.8	9.4	12.6	14.9
2	5e-1		17.2	24.4	29.1	9.2	12.2	14.4
2	0		18.6	26.4	31.7	9.2	12.1	14.3
2	1e-1		18.7	26.6	31.6	9.1	12.3	14.7
1.5	1e-1		19.8	25.5	29.3	17.5	24.5	30.8
2.5	1e-1		18.8	25.2	29.8	19.5	25.9	32.1
2	3e-1	novel	18.4	25.2	29.7	16.5	24.4	30.4
2	5e-1		18.5	25.0	29.3	15.8	23.1	29.2
2	0		18.8	25.5	29.6	17.3	23.6	30.1
2	1e-1		18.9	25.8	30.0	16.6	25.2	31.5

from 32.7% to **35.1**% R@100 on the semantic split when increasing the number of heads from 4 to 8, but the number of parameters steadily increase as the number of heads grows. When increasing the number of heads from 8 to 16, we observe some improvements (*e.g.*, 31.6% to 32.8% R@100 on the base split), but also some performance drops (*e.g.*, 30.0% to 29.0% R@100 on the novel split). Consequently, we set $H = 8$ as the default to strike an optimal balance between accuracy and computation cost.

Table S4: Ablation studies (§A) on the number of attention heads H used in the mutual visual adapter (§3.2, Eq. 5). The adopted hyperparameters are marked in red.

H	Param.	Split	R@20↑	R@50↑	R@100↑	mR@20↑	mR@50↑	mR@100↑
4	4.2M		18.9	27.0	32.5	9.1	12.2	14.4
8	7.1M	base	18.7	26.6	31.6	9.1	12.3	14.7
16	12.8M		19.4	27.5	32.8	9.5	12.7	15.1
4	4.2M		18.2	25.3	29.9	15.9	24.8	31.0
8	7.1M	novel	18.9	25.8	30.0	16.6	25.2	31.5
16	12.8M		18.0	25.3	29.0	15.8	24.0	29.1

Margin. Last, we examine the impact of the margin λ in our training objectives (§3.3, Eq. 8). As shown in Table S5, the performance remains consistent for those compared values, which indicates the robustness of our learning procedure. λ is set to be 3e-2 by default.

Table S5: Ablation studies (§A) on the margin λ (Eq. 8). The adopted hyperparameters are marked in red.

λ	Split	R@20	R@50	R@100	mR@20	mR@50	mR@100
5e-2		19.2	26.9	32.0	9.5	12.6	15.0
3e-2	base	18.7	26.6	31.6	9.1	12.3	14.7
1e-2		18.6	26.0	30.8	8.8	11.8	13.8
5e-2		18.7	24.8	29.1	17.0	24.0	30.3
3e-2	novel	18.9	25.8	30.0	16.6	25.2	31.5
1e-2		18.0	24.6	28.7	16.6	24.4	29.2

B   Implementation Details

Training. Our model is trained with a batch size of 4. One RTX 3090 is used for training. During the training process, images are resized to dimensions within the range of [600, 1,000]. For each relation, up to 50K samples are included. Random flipping is adopted for data augmentation. SGD is adopted for optimization. The initial learning rate, momentum, and weight decay are set to be 2e-2, 9e-1, 1e-4, respectively. We utilize the pre-trained weights of CLIP to initialize our model. To avoid data leakage, we remove annotations in the training set which contains categories in the novel split.

Testing. Testing is conducted on the same machine as in training. No data augmentation is used during testing. In terms of the `semantic` split, we directly applied the same weights trained on the `base` split for testing. A similar filtering strategy [12] is adopted.

Codebase and Architecture. We use the same codebase as in [21]. We adopt GPT-3.5 from OpenAI as our LLM. In terms of CLIP, we employ a widely used architecture, *i.e.*, ViT-B/32, for initializing our mutual visual adapter.

Split. We visualize all relation categories in Table S6. To guarantee fairness, we select GQA's [15] relation categories that also exist in VG [14]. As such, we can reuse SSDs obtained in our experiments on VG, thus verifying the generalizability of the proposed SSDs and self-normalizing similarity.

Table S6: Relation categories in each split.

Split	Relation Categories
Base	watching, of, hanging from, to, near, carrying, parked on, covered in, wearing, sitting on, made of, on, standing on, from, in front of, belonging to, between, above, attached to, walking on, behind, in, holding, against, has, looking at, under, at, playing, riding, covering, for, with, wears, over
Novel	flying in, painted on, mounted on, using, and, on back of, growing on, lying on, along, part of, eating, laying on, walking in, across, says
Semantic	watching, growing on, hanging from, eating, carrying, parked on, covered in, says, using, flying in, painted on, sitting on, lying on, standing on, walking in, mounted on, attached to, walking on, holding, looking at, playing, riding, covering, laying on

(a) VG [14]

Split	Relation Categories
Base	in front of, watching, riding, covered in, under, wearing, behind, parked on, covering, above, of, on, holding, walking on, at, carrying, with, sitting on, in, standing on, near
Novel	lying on, looking at, growing on, walking in, attached to, using, mounted on, hanging from, flying in, eating

(b) GQA [15]

C   Pseudo Code

The pseudo code of SDSGG is given in Algorithm S1 and Algorithm S2. We respectfully refer the reviewer to the supplementary Python files for the PyTorch implementation of SDSGG's key modules. Moreover, to guarantee reproducibility, our full code and pre-trained models will be publicly available.

Algorithm S1 Pseudo-code for MVA of SDSGG in a PyTorch-like style.

```python
"""
img_feats1: subject or object visual features, where the first vector is CLS feature, and the left
    vectors are patch features.
img_feats2: subject or object visual features. The format is the same as img_feats1.
dire_mark_feature: marker used to identify the direction of the relation.
"""
def MVA(img_feats1, img_feats2, dire_mark_feat):
    # Project features into low-dimensional, interation-aware space.
    patch_feats = LinearDown(img_feats1[1:]) # Eq. 4

    # Model the interplay between the subject and object
    out = CrossAttn(patch_feats, img_feats2[1:]) # Eq. 5

    # Project features into the original dimension.
    out = LinearUp(out) # Eq. 5

    # Add direction marker.
    mva_out = Linear(Cat([out, dire_mark_feature]))
    mva_out = (mva_out + img_feats1[0]).mean() # Eq. 6
    return mva_out
```

Algorithm S2 Pseudo-code for the forward process of SDSGG in a PyTorch-like style.

```
"""
bboxes: bounding boxes of all objects.
img: input image.
text_raw_des: raw description.
text_op_des: opposite description.
asso: the association presented in "Association between Scene-level Descriptions and Relation Categories".
targets: training targets.
"""

def forward(bboxes, img, text_raw_des, text_op_des, asso, targets):

    # Batch crop and encode images.
    crop_img = Crop(img, bboxes)
    img_feats = EncodeImage(crop_img) # CLIP's visual encoder, Eq. 3

    # Tokenize and encode descriptive text.
    text_raw_feats = EncodeText(Tokenize(text_raw_des)) # CLIP's text encoder
    text_op_feats = EncodeText(Tokenize(text_op_des))

    # Define directional marker
    mark_sub = EncodeText(Tokenize("a photo of subject"))
    mark_obj = EncodeText(Tokenize("a photo of object"))

    scores, losses = []
    # Calculate self-normalized similarity for each subject-object pair
    for sub, obj in sub_obj_pairs:

        mva_out_s2o = MVA(img_feats[sub], img_feats[obj], mark_sub)
        mva_out_o2s = MVA(img_feats[obj], img_feats[sub], mark_obj)
        mva_out = (mva_out_s2o + mva_out_o2s) / 2.0

        sim_raw = CosSim(mva_out, text_raw_feats)
        sim_op = CosSim(mva_out, text_op_feats)

        score = ((sim_raw - sim_op) * asso).sum() # Eq. 2
        scores.append(score)

        # If is training
        if training:
            loss = compute_loss(sim_raw, sim_op, targets) # Eq. 8
            losses.append(loss)

    return scores, losses
```

D Scene-specific Description

The detailed, *scene-specific* descriptions generated by LLM's multi-persona collaboration are provided in Fig. S1. As seen, our *scene-specific* descriptions (21 pairs, 42 descriptions in total) cover different scenes from a comprehensive view, *e.g.*, *spatial* descriptions ("two or more objects partially overlap each other", "interaction between objects", *etc.*), *environment* (background) descriptions ("on a road", "on a flat plane, it should appear balanced with no visible tilting", *etc.*), *semantic* descriptions ("belong to animal or human behavior", "may have contact behavior", *etc.*), *appearance* descriptions ("might have flat teeth or sharp teeth", "have a curvy body", *etc.*).
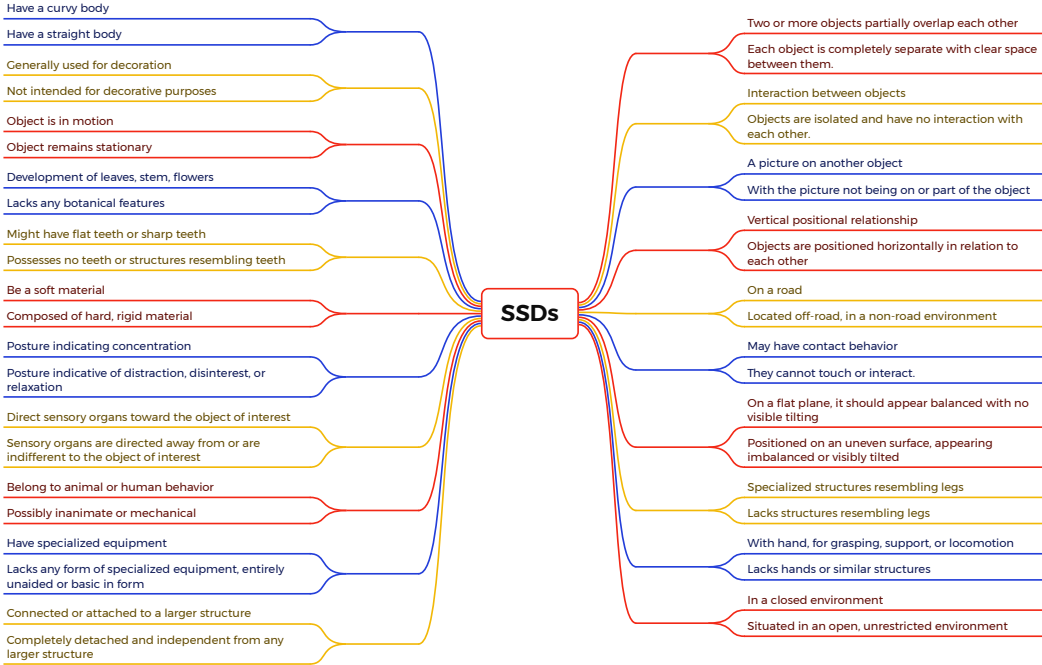

Figure S1: Illustration of the generated scene-specific descriptions.

Next, we detail each step to generate scene-specific descriptions.

❶ **Initial Description Generation.** This step can be repeated many times to generate a large number (72 after manual selection) of scene descriptions. The prompt is shown in Fig. S2.

❷ **Summarizing Descriptions.** Since these initial descriptions may suffer from noise and semantic overlap, we ask LLM to streamline and combine these descriptions, ensuring more cohesive and distinct scene-level descriptions. The prompt is shown in Fig. S3.

❸ **Description-Relation Association.** After obtaining various scene descriptions, a critical inquiry arises regarding their utility for relation detection, given their lack of explicit association with specific relation categories. To address this, we delineate three distinct scenarios characterizing the interplay between relation categories and scene descriptions: **i**) certain coexistence ($C_r^n = 1$), where a direct correlation exists; **ii**) possible coexistence ($C_r^n = 0$), indicating a potential but not guaranteed association; and **iii**) contradiction ($C_r^n = -1$), denoting an incompatibility between the scene description and relation category. Here C_r^n denotes the correlation between relation $r \in \mathcal{R}$ and n_{th} scene description, and is generated by LLMs. The prompt is shown in Fig. S4.

❹ **Opposite Description Generation.** Since classifiers are contextually bound, we generate opposite descriptions to compute *self-normalized* similarities. The prompt is shown in Fig. S5.

```
# Input: {scene content to be discussed}
# Output: 3~5 descriptions
"""
Begin by embodying three distinct personas: a biology expert, a physics expert, and
    an engineering expert. Each expert will articulate a step-by-step approach
    along with their thought process, considering various hypothetical scenarios
    relevant to their field of study, and then share their insights with the
    group. If any expert does not know the answer, he will exit the discussion.
    Once all experts have provided their analyses, summarize the final generic
    scene descriptions.
The generic scene description involves the specific appearance and the visible
    action/motion/interaction may appear in the scene (use your imagination here).
    Here are some examples of generic scene descriptions:
{some in-context examples}
Here is an example of discussion:
Discussion started!
Question: Suppose there is ... (this involves a detailed discussion of three roles)

Discussion started!
Question: Suppose there is {scene content to be discussed}, please give concise,
    generic scene descriptions of this scene.
"""
```

Figure S2: Prompts for initial description generation.

```
# Input: initial {scene descriptions}, all {relation categories}, {number of final
    scene descriptions}
# Output: final scene-level descriptions
"""
Here is a text pool that includes a series of descriptive texts:
{scene descriptions}
Here are all the relation categories:
{relation categories}
You are asked to pick {number of final scene descriptions} descriptive statements
    from the text pool that can describe at least two predicates. Think step by
    step.
"""
```

Figure S3: Prompts for summarizing descriptions.

```
# Input: {scene descriptions}, all {relation categories}
# Output: description-relation associations
"""
For {scene descriptions}, decide whether it is likely to appear in one of the
    following flat photos where {relation categories} appears.
The judgment result is to choose between [certain coexistence, possible
    coexistence, and contradiction]. When the photo shows that the scene must have
    a certain description, it is judged as "contradiction". When the photo has
    little relationship with the description, it is judged as "possible
    coexistence". When the photo shows that the scene must not appear with a
    certain description, it is judged as "contradiction".
"""
```

Figure S4: Prompts for generating description-relation associations.

```
# Input: {scene descriptions} (original scene descriptions)
# Output: final scene-level descriptions
"""
Here are some descriptions:
{scene descriptions}
Please give their opposite descriptions.
"""
```

Figure S5: Prompts for opposite description generation.

E   More Qualitative Comparison Result

We provide more visual results that compare SDSGG to CLIP [4] in Fig. S6. It can be observed that SDSGG performs robust in hard cases and can consistently deliver more satisfying results, based upon the *scene-specific* descriptions and *self-normalized* similarities.
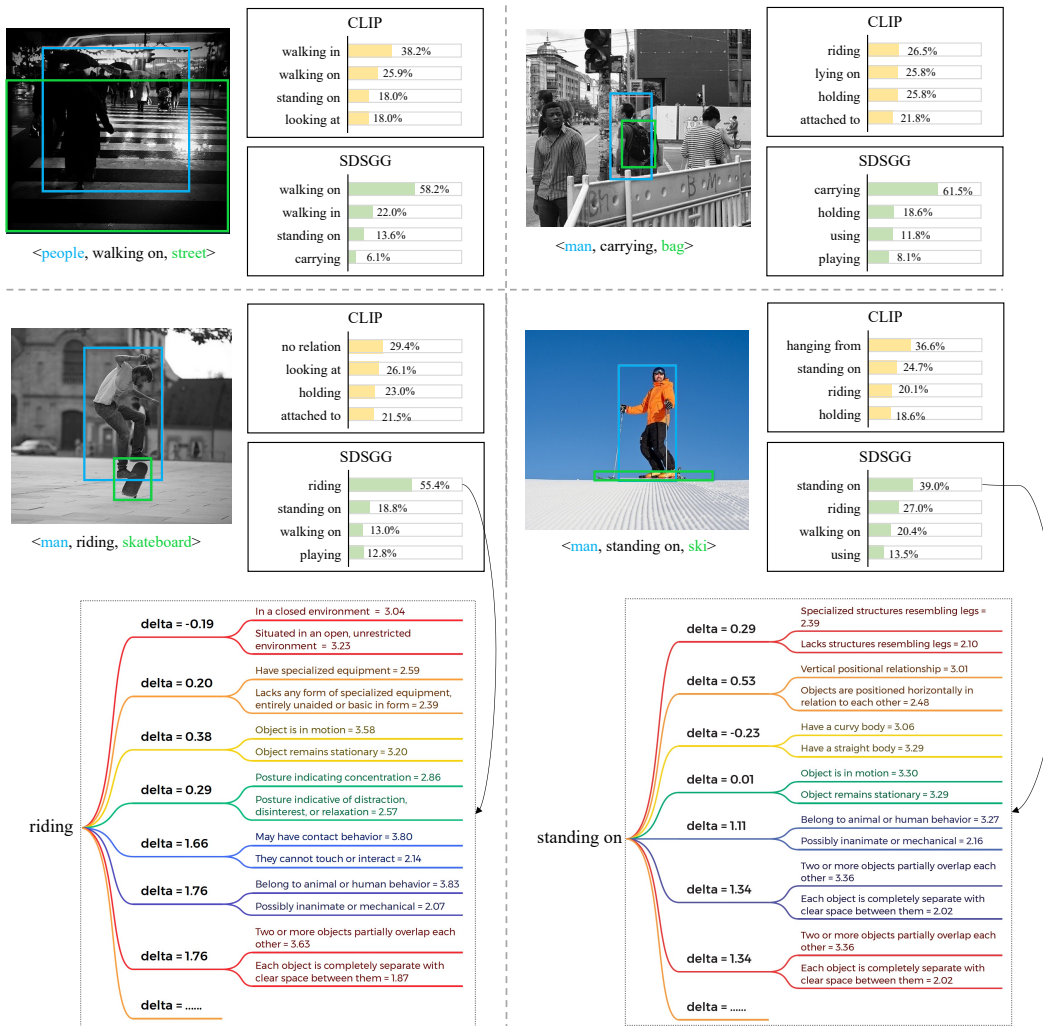

Figure S6: Visual results (§E) on VG [14] `test`. As seen, SDSGG makes the right predictions based upon the *scene-specific* descriptions and *self-normalized* similarities.

F Discussion

Limitation Analysis. Following [12], currently our algorithm is specifically designed for the predicate classification task [1, 16] to make fair comparisons with relation detection models. Such a pairwise classification task gives the ground-truth annotations of objects in the scene for easier representation learning of relations. However, these annotations are unavailable in real-world applications.

Societal Impact. This work points out the drawbacks of existing *scene-agnostic* text classifiers and accordingly introduces several new modules for both the visual and textual components, leading to a *scene-specific* description based OVSGG framework that combines the strengths of VLMs and LLMs. Like every coin has two sides, using our framework will have both positive and negative impacts. On the positive side, our work contributes to research on intelligent scene understanding, and thus is expected to eventually benefit industries such as autonomous driving. For potential negative social impact, the reliance on VLMs and LLMs could lead to the perpetuation of biases and inequalities present in the data used in these models' large-scale pre-training stage.

Future Work. As mentioned before, the focus of this work is not on object detection. It is interesting to extend our algorithm to handle the object detection task simultaneously by, for example, incorporating set-prediction architectures [91]. Moreover, the design of our multi-persona collaboration stands for an early attempt and deserves to be further explored. In addition, the architectural designs of directional marker and mutual visual adapter certainly worth further explorations, *e.g.*, efficiency [92], architecture [93, 94], and adaptive prompting [95, 96]. Furthermore, extending our algorithm to other relation detection tasks [97, 40, 42, 98] may lead to an uniformed relation detection algorithm.

G More Experiment

Training on the Full Set of Relation. We trained our model with frequency bias [25] on the full set of relations. The results are shown in Table S7.

Table S7: Results on the full set of relation.

Method	mR@50↑	mR@100↑
Ours	28.7	34.2

Different Base/Novel Splits. We trained our model on different base/novel splits to investigate the robustness further. Specifically, we **i**) change the proportion of the base and novel split and **ii**) change the categories within the base and novel split (*i.e.*, different No. for the same ratio). The results are shown in Table S8.

Table S8: Results on different base/novel splits.

No.	base:novel	Base		Novel	
		mR@50↑	mR@100↑	mR@50↑	mR@100↑
1 (paper)	35:15	12.3	14.7	25.2	31.5
2	35:15	12.4	14.8	24.3	28.2
3	32:18	11.9	14.4	23.9	28.4
4	32:18	13.6	15.9	20.6	26.7
5	38:12	11.8	14.2	23.7	29.9
6	38:12	11.5	13.7	22.6	27.1

Inference Time. Since the renormalization and similarity measurement involve only a few matrix operations that can be omitted from the complexity analysis, we will focus on the inference time of three main modules. The results are shown in Table S9.

Table S9: Inference time for different modules.

Module	Inference Time (ms/pair)
CLIP's Visual Encoder	6.5
Mutual Visual Adapter	0.2
CLIP's Text Encoder	5.4

Different Personas. The involvement of multiple personas in the discussion process enhances the diversity of generated descriptions. The key point of our multi-persona collaboration is about the "collaboration" rather than a specific persona. Actually, using only one persona can even decrease the diversity of generated descriptions and hurt the performance, as it can only generate descriptions from its own viewpoint without discussion with others. In addition, we changed the system prompt of LLM from the default like "you are a helpful AI assistant" into a persona-specific one like "you are a biologist". We then evaluate the performance of our model with these generated descriptions. The results are shown in Table S10.

Table S10: Comparison of different personas.

Method	Base		Novel	
	mR@50↑	mR@100↑	mR@50↑	mR@100↑
Multi-persona Collaboration	12.3	14.7	25.2	31.5
Biologist Persona	6.3	8.4	12.9	18.4
Engineer Persona	7.3	9.7	8.4	11.5
Physicist Persona	4.8	6.9	9.1	14.7

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We carefully described our contributions in the abstract and introduction.

   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the appendix, we discussed our limitations, societal impact, and directions for future work.

   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper is not about theory.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provided details about the methodology and implementation in the main paper and appendix. The code will be publicly available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code will be publicly available at https://github.com/guikunchen/SDSGG.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We present the experimental setup and details in the main paper and appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We run each experiment three times and report the average and standard deviation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We introduce the used computer resources in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We carefully reviewed the NeurIPS Code of Ethics

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: In the appendix, we discussed our limitations, societal impact, and directions for future work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cited related papers.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.