HiKER-SGG: Hierarchical Knowledge Enhanced Robust Scene Graph Generation

Ce Zhang   Simon Stepputtis   Joseph Campbell   Katia Sycara   Yaqi Xie
School of Computer Science, Carnegie Mellon University
{cezhang, sstepput, jacampbe, katia, yaqix}@cs.cmu.edu

Abstract

*Being able to understand visual scenes is a precursor for many downstream tasks, including autonomous driving, robotics, and other vision-based approaches. A common approach enabling the ability to reason over visual data is Scene Graph Generation (SGG); however, many existing approaches assume undisturbed vision, i.e., the absence of real-world corruptions such as fog, snow, smoke, as well as non-uniform perturbations like sun glare or water drops. In this work, we propose a novel SGG benchmark containing procedurally generated weather corruptions and other transformations over the Visual Genome dataset. Further, we introduce a corresponding approach, **Hi**erarchical **K**nowledge **E**nhanced **R**obust **S**cene **G**raph **G**eneration (HiKER-SGG), providing a strong baseline for scene graph generation under such challenging setting. At its core, HiKER-SGG utilizes a hierarchical knowledge graph in order to refine its predictions from coarse initial estimates to detailed predictions. In our extensive experiments, we show that HiKER-SGG does not only demonstrate superior performance on corrupted images in a zero-shot manner, but also outperforms current state-of-the-art methods on uncorrupted SGG tasks. Code is available at https://github.com/zhangce01/HiKER-SGG.*

1. Introduction

Visual scene understanding and the ability to extract information from images has made significant progress through the development of deep learning [7, 16, 74]. Particularly, Scene Graph Generation (SGG) [5, 87, 89] from visual inputs is a powerful method of extracting semantic information from images, enabling many subsequent reasoning tasks [14, 46, 68, 72, 93]. However, most existing studies in this field assume access to "clean" images. This contrasts with real-world situations where images often have corruptions like sun glare, dust, water drops, and rain [20, 23, 54, 67]. Being exposed to and handling such corruptions is a challenging task for many systems as it is unlikely that models can be sufficiently trained to handle such domain shifts. Inspired by the human ability to recognize objects in corrupted images using prior domain knowledge, our work leverages similar knowledge for scene graph generators. This not only enables accurate identification in corrupted images but also improves over state-of-the-art model performance on clean images.

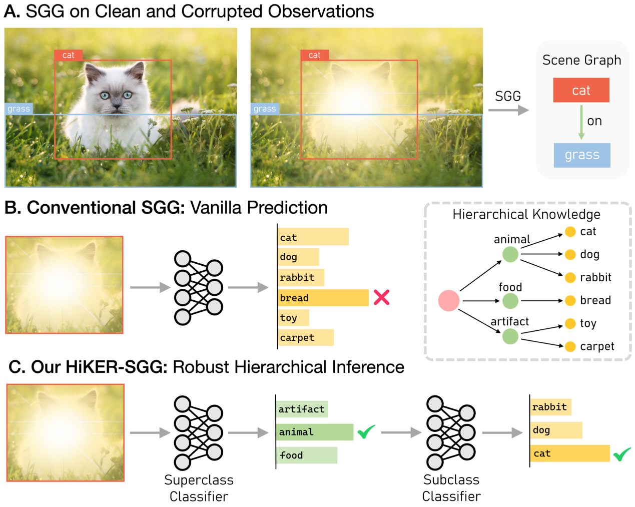


Figure 1. **We introduce a novel task: robust SGG in the presence of real-world corruptions**. Consider an image of a cat obscured by sun glare as an example, where conventional methods often struggle. Our HiKER-SGG leverages hierarchical knowledge to first infer the broader category of an object, for example, `animal`, before continuing to a more granular identification of an object constrained to various animals. By utilizing such an approach, we simplify the process to correctly identify it as a `cat`.

In this work, we propose a novel method – **Hi**erarchical **K**nowledge **E**nhanced **R**obust **S**cene **G**raph **G**eneration (HiKER-SGG) – which utilizes a hierarchical approach that reasons over multiple levels of domain knowledge with increasing granularity in order to generate accurate scene graphs for both corrupted and clean images. Further, we introduce an accompanying benchmark – Corrupted Visual Genome (VG-C) – providing 20 procedurally generated image corruptions, resembling common transformation and various weather conditions. The proposed benchmark fills a crucial gap in the field of scene graph generation and offers a comprehensive evaluation platform to assess the robustness

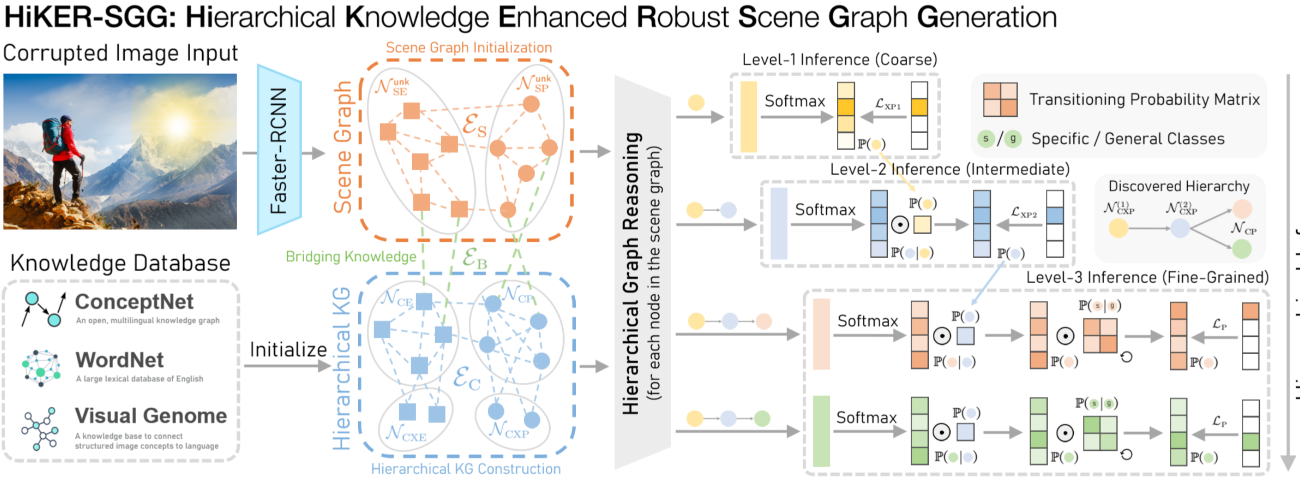

HiKER-SGG: Hierarchical Knowledge Enhanced Robust Scene Graph Generation
Corrupted Image Input
Faster-RCNN
Scene Graph Initialization
Scene Graph
$\mathcal{N}_{SE}^{unk}$
$\mathcal{N}_{SP}^{unk}$
$\mathcal{E}_S$
Bridging Knowledge
$\mathcal{E}_B$
Knowledge Database
ConceptNet
An open, multilingual knowledge graph
WordNet
A large lexical database of English
Visual Genome
A knowledge base to connect structured image concepts to language
Initialize
Hierarchical KG
$\mathcal{N}_{CE}$
$\mathcal{N}_{CP}$
$\mathcal{E}_C$
$\mathcal{N}_{CXE}$
$\mathcal{N}_{CXP}$
Hierarchical KG Construction
Hierarchical Graph Reasoning
(for each node in the scene graph)
Level-1 Inference (Coarse)
Softmax
$\mathcal{L}_{XP1}$
$\mathbb{P}(\cdot)$
Transitioning Probability Matrix
s / g Specific / General Classes
Level-2 Inference (Intermediate)
$\mathcal{L}_{XP2}$
$\mathbb{P}(\cdot|\cdot)$
Discovered Hierarchy
$\mathcal{N}_{CXP}^{(1)}$
$\mathcal{N}_{CXP}^{(2)}$
$\mathcal{N}_{CP}$
Level-3 Inference (Fine-Grained)
$\mathbb{P}(s|g)$
$\mathcal{L}_P$
Hierarchical Inference

Figure 2. **HiKER-SGG overview**. Hierarchical knowledge graphs are constructed from an external knowledge base. Given an image, we first initialize the scene graph using an off-the-shelf detector, Faster-RCNN [56]. We then create bridging connections between the hierarchical knowledge graph and the initial scene graph and perform message passing for hierarchical graph reasoning. Finally, we design a hierarchical inference process to guide the model in making step-by-step predictions explicitly.

of SGG models in adverse conditions.

Our method, HiKER-SGG, is visualized in Figure 1: When given a previously unseen corrupted image, HiKER-SGG first identifies object candidates by utilizing a pre-trained object detector. For each proposed image region (*e.g.*, a region surrounding a `cat`), HiKER-SGG determines the type of the object by first identifying its high-level type (*e.g.*, `animal`) before proceeding to more granular predictions by selecting `cat` among the possible `animals`. A key benefit of our proposed hierarchical approach is that the individual classification tasks at each level of our hierarchy are simpler than learning to create detailed predictions directly. Through each level of our hierarchy, the search space is constrained to the children of the previously identified superclass, making HiKER-SGG a powerful method for scene graph generation, particularly in the presence of image corruptions without requiring explicit training on corrupted images. Making a fundamental determination whether or not the depicted object is an `animal` or an `artifact` may still be accurate despite the corruption, which allows for more accurate object classification in subsequent levels of our hierarchy.

To evaluate the effectiveness of our proposed HiKER-SGG, we conduct comprehensive experiments on both the original clean Visual Genome (VG) dataset and our introduced VG-C benchmark. Remarkably, our proposed HiKER-SGG outperforms state-of-the-art models on clean images, and exhibits exceptional zero-shot performance in handling various types of corrupted observations.

Our work opens new research avenues and emphasizes the need for robust vision models to handle real-world image challenges and proposes the following contributions:

- We propose HiKER-SGG, a novel method for generating scene graphs through a hierarchical inference approach over structured domain knowledge, allowing it to gradually specify increasingly granular classifications through iterative sub-selection.
- We introduce a new synthetic VG-C benchmark for SGG, containing 20 challenging image corruptions, including simple transformations and severe weather conditions.
- Extensive experiments demonstrate that HiKER-SGG outperforms current state-of-the-art methods on SGG tasks, while simultaneously providing a strong zero-shot baseline for generating scene graphs from corrupted images.

2. Related Work

Scene Graph Generation. Scene graph generation has emerged as a key area of focus in computer vision research, with the goal of offering a structured depiction of an image through the identification of objects and their intricate relations [5, 66]. Furthermore, numerous studies illustrate that scene graphs can serve as a valuable source of auxiliary information, thereby enhancing image understanding for applications such as image retrieval [33, 70, 83], image captioning [31, 44, 79], image synthesis [18, 34, 73], and visual question answering [39, 53, 88]. The seminal work in this domain was conducted by Xu *et al.* [75], which employs iterative message passing to generate visually grounded scene graphs. Subsequent to this pioneering work, several researchers have adopted the message passing mechanism to better comprehend visual context [12, 21, 48, 71, 78].

While traditional SGG approaches have shown promising results, they often suffer from the long-tailed distribution of relation predicates [15, 27, 45, 61]. Predicates in visual relations are often unevenly distributed, with head predicates (*e.g.*, `on`, `have`) dominating the relation expressions [24, 32, 42, 63, 76, 77]. Such general relation expressions, however, offer limited utility for in-depth visual relation analysis [1, 19, 22]. To address this challenge, He *et al.* [26] introduces a knowledge transfer mechanism to leverage insights from head relations to enhance the representation of tail relations. Guo *et al.* [22] refines biased predicate predictions based on the confusion matrix generated by training data. Our work differs from conventional SGG in that we don't assume that observations are perfect. We allow for corruptions in images, which are typical in real-world situations.

Knowledge Based SGG. Recently, several approaches have been proposed to integrate external knowledge, referred to as *commonsense*, to refine predicate and object prediction [2, 3] and enhance the generalizability of the SGG model [8, 21, 41, 81, 86]. Specifically, GB-Net [85] suggests that a scene graph can be perceived as an instantiation of a commonsense knowledge graph conditioned by the content of the image, and employs GGNN [47] to iteratively propagate messages between these two graphs for SGG task. Furthermore, EB-Net [9] advances this by enriching the knowledge graph for SGG with off-scene entities, thereby offering a more comprehensive and context-aware scene graph representation. In this work, we extend this by introducing superclass nodes and incorporating hierarchical edges into the knowledge graph, thereby facilitating hierarchical prediction for SGG models. This is particularly advantageous when observations are corrupted, where features for specific classes are not easily detectable. In such cases, the hierarchical knowledge guides the model to first detect the superclass features. By adopting this approach, we can streamline the search space and facilitate more accurate predictions for finer classes.

Corrupted Observation Perception. In many computer vision tasks, it is a common assumption among researchers that the input image is invariably flawless and clear. However, this is often not the case in practical scenarios. To address this important issue, several benchmarks have been introduced to assess the robustness of the neural network models to real-world corruptions [28, 50]. Within the context of corruption robustness, recent advancements can be broadly categorized into transfer learning [51, 64, 65], adversarial training [30, 37, 57], data augmentation [29, 82, 90, 91], and large-scale pre-training [4, 17, 55]. Recently, LogicDef [80] proposes a logic rules based defense method for adversarial patch attacks on images with multiple objects, utilizing logic rules learned from object relations to identify the attacked object. However, their approach assumes that the attack patch is on one single object, known to be under attack. Additionally, they assume that the relations between objects remain unaffected by the attack. In contrast, our work allows for corruption to occur at any location, potentially impacting an unknown number of objects and relations, which is more challenging as well as more realistic. To the best of our knowledge, ours is the first work to introduce corruptions into SGG and to propose the integration of hierarchical knowledge to ensure robust SGG in the presence of such corruptions.

3. HiKER-SGG

We introduce a novel framework HiKER-SGG, as illustrated in Figure 2, to enable robust scene understanding for observations with potential corruptions.

3.1. Problem Definition

Given an image $\mathcal{I}$ in a dataset $\boldsymbol{\mathcal{I}}$, the SGG model aims to generate a directed scene graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where each node $\mathcal{N}_i \in \mathcal{N}$ in the scene graph represents a localized object with bounding box b_i and object class $\mathcal{C}_i^{\mathrm{E}}$, and each edge $\mathcal{E}_i \in \mathcal{E}$ denotes a predicate class $\mathcal{C}_i^{\mathrm{P}}$ between two objects. A well-constructed scene graph $\mathcal{G}$ contains a collection of visual relation triplets ($\langle$`subject-predicate-object`$\rangle$), which can be utilized to comprehensively describe the image $\mathcal{I}$.

Our proposed HiKER-SGG follows a two-stage paradigm. We first generate a set of entity proposals with corresponding features using an off-the-shelf object detector (*e.g.* Faster-RCNN [56]) with a feature extraction network (*e.g.* VGG [58] or ResNet [25]). The features extracted from the union box between two entities are used to represent their associated predicates. Leveraging these features, we jointly make predictions for both the entity and predicate classes.

3.2. Hierarchical Structure Discovery

At the center of our work lies the hierarchical representation of domain knowledge. In this section, we introduce our automated approach to define hierarchies given GloVe [52] word embeddings and pattern similarity using MotifNet [87]. A straightforward method is to manually set up these hierarchical relations. For instance, we can follow Zellers *et al.* [87] to categorize 50 predicate classes into 3 superclasses, namely `geometric`, `possessive`, and `semantic`, respectively. Similarly, the 150 object classes can also be categorized into 12 superclasses, such as `artifact`, `animal`, *etc.*

However, we recognize that there are various reasonable criteria for defining these hierarchies (*e.g.*, by functions, sizes, materials). Setting up these hierarchies manually introduces subjectivity, which could hinder the capability of our approach on the unbiased SGG task. To address this issue, we adopt a hierarchical clustering [35] algorithm, capable of revealing multi-level clusters based on a similarity metric, to discover the hierarchical structure for the entity and predicate classes. The similarity function used in hierarchical clustering is the weighted sum of the following two similarities:

(1) Semantic Similarity. We use the GloVe [52] word embeddings $\mathbf{e}^{\mathrm{E}}$ and $\mathbf{e}^{\mathrm{P}}$ to calculate the cosine similarity between each pair of entities (E) and predicates (P):

$$\mathcal{S}_{\text{sem}}\left(\mathcal{C}_i^{\mathrm{E/P}}, \mathcal{C}_j^{\mathrm{E/P}}\right) = \frac{\mathbf{e}_i^{\mathrm{E/P}} \cdot \mathbf{e}_j^{\mathrm{E/P}}}{\left\|\mathbf{e}_i^{\mathrm{E/P}}\right\| \left\|\mathbf{e}_j^{\mathrm{E/P}}\right\|}. \quad (1)$$

(2) Pattern Similarity. We employ the MotifNet [87] baseline to generate confusion matrices $\mathcal{R}^{\mathrm{E/P}}$ for both entities and predicates on the training dataset of Visual Genome [38]. Each matrix entry, $\mathcal{R}_{ij}$, indicates the likelihood (between 0 and 1) that the actual class is i and the predicted class is j. Recognizing that similar classes often have similar patterns that might confuse our model, we compute the similarity based on the probability of the baseline method's misclassification between pairs of entities and predicates, written as

$$\mathcal{S}_{\mathrm{pat}}\left(\mathcal{C}_i^{\mathrm{E/P}}, \mathcal{C}_j^{\mathrm{E/P}}\right) = \mathcal{R}_{ij}^{\mathrm{E/P}} + \mathcal{R}_{ji}^{\mathrm{E/P}} \tag{2}$$

The hierarchies discovered through this method, which consider both semantic and pattern similarities, offer a more effective guidance for our hierarchical prediction approach, as discussed in Section 4.3. More details about the clustering algorithm and hierarchy visualization can be found in Section A.1 of the Supplementary Materials.

3.3. Hierarchical Knowledge Construction

In the previous section, we discovered the hierarchies using those two metrics. This section details the representation of this hierarchical knowledge in our commonsense graph.

Commonsense Knowledge Graph. Initially, we construct a commonsense knowledge graph that does not incorporate hierarchical knowledge. Similar to GB-Net [85], we leverage a commonsense knowledge graph which contains the possible relations between objects derived from extensive datasets like ConceptNet [59], WordNet [49], *etc*. Its edges serve as repositories of information regarding the general knowledge associated with objects, exemplified by connections such as `man-wears-shirt` and `cat-is-animal`. For simplicity, we define our commonsense graph as comprising a set of commonsense entity (CE) nodes $\mathcal{N}_{\mathrm{CE}}$ and commonsense predicate (CP) nodes $\mathcal{N}_{\mathrm{CP}}$ that are present in our SGG task. The edges in the commonsense graph $\mathcal{E}_{\mathrm{C}}$ store the relations between each pair of nodes in both sets, which can be formally denoted as

$$\mathcal{E}_{\mathrm{C}} = \{\mathcal{E}_{\texttt{relation}}^{\mathrm{CE}\to\mathrm{CP}}\} \cup \{\mathcal{E}_{\texttt{relation}}^{\mathrm{CP}\to\mathrm{CE}}\} \cup \{\mathcal{E}_{\texttt{relation}}^{\mathrm{CE}\to\mathrm{CE}}\} \cup \{\mathcal{E}_{\texttt{relation}}^{\mathrm{CP}\to\mathrm{CP}}\}. \tag{3}$$

We initialize the CE and CP nodes features with a linear projection of their word embeddings [52] $\mathbf{e}_i^{\mathrm{E}}$ and $\mathbf{e}_i^{\mathrm{P}}$:

$$\mathbf{x}_i^{\mathrm{CE}} = \texttt{LinearProj}(\mathbf{e}_i^{\mathrm{E}}),\ \mathbf{x}_i^{\mathrm{CP}} = \texttt{LinearProj}(\mathbf{e}_i^{\mathrm{P}}). \tag{4}$$

Hierarchical Commonsense Knowledge Graph. To integrate hierarchical information discovered in Section 3.2 into the prediction process, we introduce a set of specialized entity and predicate nodes across different levels within the commonsense knowledge graph, referred to as commonsense superclass entity (CXE) and commonsense superclass predicate (CXP) nodes[1], as shown in Figure 2. These nodes are denoted as $\mathcal{N}_{\mathrm{CXE}}$ and $\mathcal{N}_{\mathrm{CXP}}$, and correspond to a set of overarching categories for entities and predicates, respectively.

The initial representations of these superclass nodes are established by averaging the representations of N_k subclass CE/CP nodes associated with each superclass, as follows:

$$\mathbf{x}_k^{\mathrm{CXE/CXP}} = \frac{\sum_i \mathbf{x}_i^{\mathrm{CE/CP}}}{N_k} = \frac{\sum_i \texttt{LinearProj}(\mathbf{e}_i^{\mathrm{E/P}})}{N_k}. \tag{5}$$

We also establish binary connections $\mathcal{E}_{\texttt{hierarchical}}^{\mathrm{CXP}\to\mathrm{CP/CXP}}$ and $\mathcal{E}_{\texttt{hierarchical}}^{\mathrm{CP/CXP}\to\mathrm{CXP}}$ within the node sets $\mathcal{N}_{\mathrm{CXP}}$ and $\mathcal{N}_{\mathrm{CP}}$ to encode hierarchical information[2]. Similar hierarchical edges are also established for the entity nodes. These edges also facilitate message passing, enabling the updating of superclass node representations, which are subsequently employed in computing superclass similarities. The final edges in the commonsense graph $\mathcal{E}_{\mathrm{C}}$ can be represented by

$$\begin{aligned}\mathcal{E}_{\mathrm{C}} = &\{\mathcal{E}_{\texttt{relation}}^{\mathrm{CE}\to\mathrm{CP}}\} \cup \{\mathcal{E}_{\texttt{relation}}^{\mathrm{CP}\to\mathrm{CE}}\} \cup \{\mathcal{E}_{\texttt{relation}}^{\mathrm{CE}\to\mathrm{CE}}\} \cup \\ &\{\mathcal{E}_{\texttt{relation}}^{\mathrm{CP}\to\mathrm{CP}}\} \cup \{\mathcal{E}_{\texttt{hierarchical}}^{\mathrm{CXE}\to\mathrm{CE/CXE}}\} \cup \{\mathcal{E}_{\texttt{hierarchical}}^{\mathrm{CE/CXE}\to\mathrm{CXE}}\} \cup \\ &\{\mathcal{E}_{\texttt{hierarchical}}^{\mathrm{CXP}\to\mathrm{CP/CXP}}\} \cup \{\mathcal{E}_{\texttt{hierarchical}}^{\mathrm{CP/CXP}\to\mathrm{CXP}}\}.\end{aligned} \tag{6}$$

3.4. Scene Graph Initialization

So far, we developed a hierarchical commonsense knowledge graph sourced from knowledge databases. Our next step is to construct a scene graph from the given input image.

A scene graph is different from a commonsense graph in that: (1) each scene entity (SE) node $\mathcal{N}_{\mathrm{SE}}$ is associated with a bounding box, *i.e.* $\mathcal{N}_{\mathrm{SE}} \subseteq [0,1]^4 \times \mathcal{N}_{\mathrm{CE}}$; (2) each scene predicate (SP) node $\mathcal{N}_{\mathrm{SP}}$ is associated with a pair of SE nodes, *i.e.* $\mathcal{N}_{\mathrm{SP}} \subseteq \mathcal{N}_{\mathrm{SE}} \times \mathcal{N}_{\mathrm{SE}} \times \mathcal{N}_{\mathrm{CP}}$. The directed edges $\mathcal{E}_{\mathrm{S}}$ in the scene graph can be similarly defined as

$$\mathcal{E}_{\mathrm{S}} = \{\mathcal{E}_{\texttt{subjectOf}}^{\mathrm{SE}\to\mathrm{SP}}\} \cup \{\mathcal{E}_{\texttt{objectOf}}^{\mathrm{SE}\to\mathrm{SP}}\} \cup \{\mathcal{E}_{\texttt{hasSubject}}^{\mathrm{SP}\to\mathrm{SE}}\} \cup \{\mathcal{E}_{\texttt{hasObject}}^{\mathrm{SP}\to\mathrm{SE}}\}. \tag{7}$$

In our SGG settings, the true classes for the SE/SP nodes might not be provided, which requires us to predict them. Therefore, we modify the scene graph entity nodes needed to be classified as $\mathcal{N}_{\mathrm{SE}}^{\mathsf{unk}} \subseteq [0,1]^4$, and scene graph predicate nodes needed to be classified as $\mathcal{N}_{\mathrm{SP}}^{\mathsf{unk}} \subseteq \mathcal{N}_{\mathrm{SE}} \times \mathcal{N}_{\mathrm{SE}}$, where $\mathcal{N}_{\mathrm{SE/SP}}^{\mathsf{unk}}$ means the classes of the SE/SP nodes are unknown.

To initialize the scene graph for each sample, we first utilize the object detector to find potential objects. We then create a SE node for each object and a SP node for each pair of objects. The SE node is initialized by RoI-aligned [56] feature vector $\mathbf{v}_i^{\mathrm{E}}$, and the SP node is initialized by RoI feature $\mathbf{v}_i^{\mathrm{P}}$ of the union bounding box:

$$\mathbf{x}_i^{\mathrm{SE}} = \texttt{FCNet}(\mathbf{v}_i^{\mathrm{E}})\ , \quad \mathbf{x}_i^{\mathrm{SP}} = \texttt{FCNet}(\mathbf{v}_i^{\mathrm{P}}), \tag{8}$$

where `FCNet` denotes a fully connected network. It should be noted that the weights for these two fully connected networks are distinct and not shared.

[1] We use "X" as the notation for "superclass" to avoid ambiguity.

[2] In order to represent the multi-level hierarchy we discovered, two CXE/CXP nodes at different levels may also exhibit a hierarchical relation.

3.5. Bridging Hierarchical Knowledge and SGG

To bridge the knowledge graph and the scene graph, we create *bridge edges* $\mathcal{E}_{\text{B}}$ to facilitate the mutual information flow during training. Specifically, these bi-directional bridge edges link an entity or predicate from the scene graph to its corresponding labels in the commonsense graph[3]. The bridge edges $\mathcal{E}_{\text{B}}$ can be defined as

$$\mathcal{E}_{\text{B}} = \{\mathcal{E}^{\text{SE}\to\text{CE}}_{\texttt{classTo}}\} \cup \{\mathcal{E}^{\text{SP}\to\text{CP}}_{\texttt{classTo}}\} \cup \{\mathcal{E}^{\text{CE}\to\text{SE}}_{\texttt{hasInst}}\} \cup \{\mathcal{E}^{\text{CP}\to\text{SP}}_{\texttt{hasInst}}\}. \quad (9)$$

Initially, we link each SE node to multiple CE nodes and assign weights based on the labels predicted by Faster R-CNN. The edges between SP and CP nodes start as an empty set and will be updated during message propagation. Enforcing the information flow between the knowledge graph and the scene graph, we adopt a variant of GGNN [47] to update node representations and propagate messages among nodes using a Gated Recurrent Unit (GRU) [11] updating rule:

$$\mathbf{x}_i^{\phi} \leftarrow \texttt{GRU Update}(\mathbf{x}_i^{\phi}), \quad (10)$$

where $\leftarrow$ denotes updating the node representation, with the superscript $\phi \in \{\text{SE}, \text{SP}, \text{CE}, \text{CP}, \text{CXE}, \text{CXP}\}$.

After each iteration of message propagation, we compute the similarities of each SE/SP node to all CE/CP nodes by

$$\text{sim}\left(\mathbf{x}_i^{\phi}, \mathbf{x}_j^{\phi}\right) = \left(\texttt{FCNet}\left(\mathbf{x}_i^{\phi}\right)\right)^{\top}\left(\texttt{FCNet}\left(\mathbf{x}_j^{\phi}\right)\right). \quad (11)$$

The pairwise similarities, which quantify the connections between scene nodes and commonsense nodes, are used to update the weights of the bridge edges after each iteration. Explicitly, the weights of the bridge edges $\mathcal{E}_{\text{B}}$ are updated by

$$\mathbf{w}_{ij}^{\text{SE}\leftrightarrow\text{CE}} \leftarrow \frac{\exp\left(\text{sim}\left(\mathbf{x}_i^{\text{SE}}, \mathbf{x}_j^{\text{CE}}\right)\right)}{\sum_{j'} \exp\left(\text{sim}\left(\mathbf{x}_i^{\text{SE}}, \mathbf{x}_{j'}^{\text{CE}}\right)\right)}, \quad (12)$$

$$\mathbf{w}_{ij}^{\text{SP}\leftrightarrow\text{CP}} \leftarrow \frac{\exp\left(\text{sim}\left(\mathbf{x}_i^{\text{SP}}, \mathbf{x}_j^{\text{CP}}\right)\right)}{\sum_{j'} \exp\left(\text{sim}\left(\mathbf{x}_i^{\text{SP}}, \mathbf{x}_{j'}^{\text{CP}}\right)\right)}, \quad (13)$$

where $\mathbf{w}_{ij}^{\text{SE}\leftrightarrow\text{CE}}$ and $\mathbf{w}_{ij}^{\text{SP}\leftrightarrow\text{CP}}$ represent the shared weights of bi-directional bridge edges connecting a specific pair of SE/SP and CE/CP nodes, respectively. After t steps of message propagation, we can leverage the node representations from both graphs to infer the unknown class of SE/SP nodes.

3.6. Hierarchical Inference

Using the updated node representations in both graphs, we propose to determine the class of each unknown SE/SP node by a hierarchical inference process. Here, we present the inference process for predicate classification only. The same paradigm is also applied to entity nodes.

Specifically, We enforce our model to infer the predicate class sequentially from higher to lower levels. For simplicity, we introduce our approach using a 3-level hierarchy; however, this hierarchical inference scheme is scalable to accommodate a more complex hierarchy. In the 3-level case, the CXP nodes can be split into two groups: higher-level nodes denoted by $\mathcal{N}_{\text{CXP}}^{(1)}$ and lower-level nodes denoted by $\mathcal{N}_{\text{CXP}}^{(2)}$. The hierarchical path from the top superclass node to the final subclass node can be expressed as $\mathcal{N}_{\text{CXP}}^{(1)} \to \mathcal{N}_{\text{CXP}}^{(2)} \to \mathcal{N}_{\text{CP}}$, which corresponds to the classification sequence from higher to lower predicate class: $\mathcal{C}^{\text{XP1}} \to \mathcal{C}^{\text{XP2}} \to \mathcal{C}^{\text{P}}$.

Specifically, we first compute the similarities between the node representations of each SP node and the higher-level CXP nodes within the hierarchical knowledge graph to determine the level-1 superclass probabilities, written as

$$\mathbb{P}\left(\mathcal{C}^{\text{XP1}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}\right) = \texttt{Softmax}\left(\text{sim}\left(\mathbf{x}_i^{\text{SP}}, \mathbf{x}_{k_1}^{\text{CXP1}}\right)\right). \quad (14)$$

Here, k_1 denotes the level-1 superclass indices, $\mathbf{x}_{k_1}^{\text{CXP1}}$ denotes the node representation for $\mathcal{N}_{\text{CXP}}^{(1)}$, and $\text{sim}(\cdot, \cdot)$ is defined according to Equation (11).

Once we have classified the level-1 superclass for each unknown predicate node in the scene graph, we then examine the conditional probabilities $\mathbb{P}\left(\mathcal{C}^{\text{XP2}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}, \mathcal{C}^{\text{XP1}}\right)$, *i.e.*, the probabilities of level-2 superclass predicates given the level-1 superclass. The probabilities can be computed as follows:

$$\mathbb{P}\left(\mathcal{C}^{\text{XP2}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}, \mathcal{C}^{\text{XP1}}\right) = \texttt{Softmax}\left(\text{sim}\left(\mathbf{x}_i^{\text{SP}}, \mathbf{x}_{k_2}^{\text{CXP2}}\right)\right), \quad (15)$$

where k_2 denotes the level-2 superclass predicate indices in a given level-1 superclass. Ultimately, the conditional probabilities of final subclass predicates can be written as

$$\mathbb{P}\left(\mathcal{C}^{\text{P}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}, \mathcal{C}^{\text{XP2}}\right) = \texttt{Softmax}\left(\text{sim}\left(\mathbf{x}_i^{\text{SP}}, \mathbf{x}_j^{\text{CP}}\right)\right). \quad (16)$$

In general, given an unknown predicate node, the predicted probability of each predicate category can be computed by multiplying the three probabilities derived above:

$$\begin{aligned} \mathbb{P}\left(\mathcal{C}^{\text{P}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}\right) &= \mathbb{P}\left(\mathcal{C}^{\text{XP2}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}\right) \cdot \mathbb{P}\left(\mathcal{C}^{\text{P}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}, \mathcal{C}^{\text{XP2}}\right) \quad (17) \\ &= \mathbb{P}\left(\mathcal{C}^{\text{XP1}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}\right) \cdot \mathbb{P}\left(\mathcal{C}^{\text{XP2}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}, \mathcal{C}^{\text{XP1}}\right) \cdot \mathbb{P}\left(\mathcal{C}^{\text{P}} | \mathcal{N}_{\text{SP}}^{\textsf{unk}}, \mathcal{C}^{\text{XP2}}\right). \end{aligned}$$

3.7. Adaptive Refinement

Due to the inherent bias in the Visual Genome [38] dataset, most existing SGG models tend to favor commonly occurring predicate classes. In this work, we integrate an adaptive refinement mechanism into our model to mitigate biases in predicate classes. This enhancement aims to predict more specific and informative predicates (*e.g.*, `standing on`, `sitting on`), as opposed to general ones (*e.g.*, `on`). Essentially, our goal is to find transitioning probabilities $\mathbb{P}(\mathcal{C}_s^{\text{P}} | \mathcal{C}_g^{\text{P}})$ that can convert a general prediction into a more specific prediction for predicate classes.

Unlike previous method like G2S [22] which incorporates fixed transitioning probabilities to debias the predictions, our adaptive refinement dynamically updates the transition probabilities during the training process. Specifically, we adopt the predicate confusion matrix generated by the Mo-

[3] Given the symmetric nature of the relation, the bridge edges are implemented as bi-directional directed edges with shared weights.

tifNet [87] baseline as initialization for $\mathcal{R}$. We then create a transitioning probability matrix by row-normalizing the diagonal-augmented confusion matrix:

$$\mathcal{T} = \texttt{RowNormalize}\left(\mathcal{R} + I\right), \tag{18}$$

where I represents an identity matrix of the same size as the confusion matrix $\mathcal{R}$. The transitioning probability $\mathbb{P}(\mathcal{C}^{\text{P}}_s|\mathcal{C}^{\text{P}}_g)$ can be subsequently represented by a particular entry $\mathcal{T}_{ij}$, which aligns with the respective classes $\mathcal{C}^{\text{P}}_s$ and $\mathcal{C}^{\text{P}}_g$.

Combining this refinement with our hierarchical prediction approach, we can rewrite Equation (17) as:

$$\begin{aligned}\mathbb{P}\big(\mathcal{C}^{\text{P}}|\mathcal{N}^{\mathsf{unk}}_{\text{SP}}\big) =& \mathbb{P}\big(\mathcal{C}^{\text{XP1}}|\mathcal{N}^{\mathsf{unk}}_{\text{SP}}\big) \cdot \mathbb{P}\big(\mathcal{C}^{\text{XP2}}|\mathcal{N}^{\mathsf{unk}}_{\text{SP}}, \mathcal{C}^{\text{XP1}}\big) \\ & \cdot \mathbb{P}\big(\mathcal{C}^{\text{P}}|\mathcal{N}^{\mathsf{unk}}_{\text{SP}}, \mathcal{C}^{\text{XP2}}\big) \cdot \mathbb{P}\big(\mathcal{C}^{\text{P}}_s|\mathcal{C}^{\text{P}}_g\big). \end{aligned} \tag{19}$$

During the training stage, we aim to uncover deeper correlations between predicate classes, facilitating a more fine-grained prediction. Therefore, we propose to re-evaluate our SGG model on the training dataset after each training epoch to obtain a new $\mathcal{T}^m$ following Equation (18). We then blend this matrix with the one from the previous epoch using a weighted linear combination:

$$\mathcal{T}^m \leftarrow \alpha\mathcal{T}^m + (1-\alpha)\mathcal{T}^{m-1}, \tag{20}$$

where m represents the current epoch index, and α is a hyperparameter to control the update rate. This updated matrix will be used for predicate classification in the next training epoch. Additional discussions on adaptive refinement are provided in Section A.3 of the Supplementary Materials.

During the training stage, we update our parameters using the following loss terms to supervise both the superclass and subclass predictions defined in Equations (14) and (19):

$$\begin{aligned}\mathcal{L}_{\text{XP1}} &= \texttt{NLL Loss}\big(\mathbb{P}\big(\mathcal{C}^{\text{XP1}}|\mathcal{N}^{\mathsf{unk}}_{\text{SP}}\big), \texttt{OneHot}\big(\mathcal{C}^{\text{XP1}}_{\text{GT}}\big)\big), \\ \mathcal{L}_{\text{XP2}} &= \texttt{NLL Loss}\big(\mathbb{P}\big(\mathcal{C}^{\text{XP2}}|\mathcal{N}^{\mathsf{unk}}_{\text{SP}}\big), \texttt{OneHot}\big(\mathcal{C}^{\text{XP2}}_{\text{GT}}\big)\big), \\ \mathcal{L}_{\text{P}} &= \texttt{NLL Loss}\big(\mathbb{P}\big(\mathcal{C}^{\text{P}}|\mathcal{N}^{\mathsf{unk}}_{\text{SP}}\big), \texttt{OneHot}\big(\mathcal{C}^{\text{P}}_{\text{GT}}\big)\big), \end{aligned} \tag{21}$$

where $\mathcal{C}^{\text{XP1/XP2}}_{\text{GT}}$ and $\mathcal{C}^{\text{P}}_{\text{GT}}$ represent the ground-truth labels for the superclass and subclass predicates, respectively.

4. Experiments

In this section, we conduct extensive experiments on the large-scale Visual Genome (VG) [38] dataset and our corrupted Visual Genome (VG-C) benchmark. The results indicate that HiKER-SGG excels beyond state-of-the-art models with superior performance on both clean and corrupted images. It is noteworthy that our method is corruption-agnostic, as it is trained solely on clean images and directly tested on corrupted ones without additional training.

4.1. Experimental Settings

Datasets. Following the literature [9, 85], we conduct experiments using the widely recognized Visual Genome (VG) [38] dataset, which consists of 108,077 images, each annotated with objects and relations. Following previous work [75], we filter the dataset to use the most frequent 150 object classes and 50 predicate classes for experiments.

To standardize and evaluate SGG robustness, we create a **corrupted Visual Genome (VG-C) benchmark**, which comprises 20 corruption types designed to simulate realistic corruptions that may occur in real-world scenarios. Specifically, the first 15 types of corruption introduced by Hendrycks *et al.* [28] are widely recognized as standard benchmarks for evaluating robustness. To further align with real-world scenarios, we introduce 5 additional types of *natural* corruption[4] to our evaluation: sun glare, water-drop, wildfire smoke, rain, and dust. A detailed description and visualization of the VG-C dataset are provided in Section B.2 of the Supplementary Materials.

Tasks and Metrics. We assess the effectiveness of our proposed approach in the context of two standard SGG tasks: Predicate Classification (PredCls) and Scene Graph Classification (SGCls). We evaluate the performance of the SGG models by top-k mean triplet recall (mR@k) metric on both the PredCls and SGCls tasks. We also report the constrained (C) and unconstrained (UC) performance results, depending on the presence or absence of the graph constraint. This constraint restricts our SGG model to predict only a single relation between each pair of objects.

Implementation Details. We use the Faster-RCNN [56] as the object detector, which is based on VGG-16 [58] backbone provided by Zellers *et al.* [87]. Regarding `FCNet` in Equations (8) and (11), we follow GB-Net [85] to use 3-layer fully connected networks with `ReLU` activation. We set the message propagation steps $t = 3$ and use a 1024-dimensional vector to represent each node. The hyperparameter α in Equation (20) is set to 0.9. We also adopt the BPL [22] method to train our SGG model with unbiased data. In our experiments, we train our model for 30 epochs, initializing the learning rate at 1×10^{-4}. A single NVIDIA Quadro RTX 6000 GPU is used for all the experiments.

Baselines. We compare our performance with the following state-of-the-art SGG methods: IMP+ [75], Neural Motifs [87], VCTree [62], PCPL [77], CogTree [84], EBM [60], G2S [22], DLFE [10], RTPB [6], PPDL [43], NICE [40], NARE [19], HML [13], SQUAT [36], PE-Net [92], PE-Net + SIL [69]. Additionally, we compare our approach with SGG methods that are knowledge graph-based, which are closely related to our work: GB-Net [85] and EB-Net + EOA [9]. For a fair comparison, we present the performance results of these methods directly from their respective original papers.

4.2. Results and Discussions

Quantitative Results. In Table 1, we report our performance results for the PredCls task and SGCls tasks on clean

[4] Here, *natural* corruptions refer to image degradations that arise from real-world environmental factors affecting the scene being captured.

Table 1. **Performance comparison with the state-of-the-art SGG methods on the Visual Genome [38] dataset**. The best results for each metric are in **bold**, while the second-best results are underlined. "-" denotes unavailable results due to incompatible experimental settings.

Method	Venue	PredCls			SGCls		
		mR@20: UC/C	mR@50: UC/C	mR@100: UC/C	mR@20: UC/C	mR@50: UC/C	mR@100: UC/C
IMP+ [75]	*CVPR'17*	- / -	20.3 / 9.8	28.9 / 10.5	- / -	12.1 / 9.8	16.9 / 10.5
Neural Motifs [87]	*CVPR'18*	- / 10.8	24.8 / 14.0	37.3 / 15.3	- / 6.3	13.5 / 7.7	19.6 / 8.2
VCTree [62]	*CVPR'19*	- / 14.0	- / 17.9	- / 19.4	- / 8.2	- / 10.1	- / 10.8
PCPL [77]	*ACMMM'20*	- / -	50.6 / 35.2	62.6 / 37.8	- / -	26.8 / 18.6	32.8 / 19.6
Transformer + CogTree [84]	*IJCAI'21*	- / 22.9	- / 28.4	- / 31.0	- / 13.0	- / 15.7	- / 16.7
VCTree + EBM [60]	*CVPR'21*	- / 14.2	- / 18.0	- / 28.8	- / 8.2	- / 10.2	- / 11.0
G2S: Transformer [22]	*ICCV'21*	- / 26.7	- / 31.9	- / 34.2	- / 15.7	- / 18.5	- / 19.4
MotifNet + DLFE [10]	*ACMMM'21*	- / 22.1	- / 26.9	- / 28.8	- / 12.8	- / 15.2	- / 15.9
MotifNet + RTPB [6]	*AAAI'22*	- / 28.8	- / 35.3	- / 37.7	- / 16.3	- / 19.4	- / 20.6
MotifNet + PPDL [43]	*CVPR'22*	- / 27.9	- / 32.2	- / 33.3	- / 15.8	- / 17.5	- / 18.2
MotifNet + NICE [40]	*CVPR'22*	- / 23.7	- / 29.8	- / 32.2	- / 13.6	- / 16.7	- / 17.9
MotifNet + NARE [19]	*CVPR'22*	- / 21.3	- / 27.1	- / 29.7	- / 11.3	- / 14.3	- / 15.7
Transformer + HML [13]	*ECCV'22*	- / 27.4	- / 33.3	- / 35.9	- / 15.7	- / 19.1	- / 20.4
SQUAT [36]	*CVPR'23*	- / 25.6	- / 30.9	- / 33.4	- / 14.4	- / 17.5	- / 18.8
PE-Net [92]	*CVPR'23*	- / 25.8	- / 31.4	- / 33.5	- / 15.2	- / 18.2	- / 19.3
PE-Net + SIL [69]	*ACMMM'23*	- / 26.9	- / 33.1	- / 35.3	- / 16.7	- / 19.9	- / 20.7
GB-Net [85]	*ECCV'20*	23.8 / 15.3	41.1 / 19.3	55.4 / 20.9	13.1 / 7.9	21.4 / 9.6	29.1 / 10.2
EB-Net + EOA [9]	*WACV'23*	39.8 / 30.8	54.9 / 36.7	66.3 / 39.2	19.6 / 14.9	26.7 / 17.3	32.5 / 18.3
HiKER-SGG (Ours)	-	**42.1 / 33.4**	**57.9 / 39.3**	**69.2 / 41.2**	**22.6 / 18.2**	**30.0 / 20.3**	**36.7 / 21.4**

Table 2. **Performance comparison with the state-of-the-art SGG methods for the PredCls task on the corrupted Visual Genome [38] dataset**. We report the accuracy in percentage for the mR@20: UC/C, mR@50: UC/C, mR@100: UC/C metrics, structured in six rows. The best results for each metric are in **bold**. The last column reports the average mean recall across all 20 types of corruption, and the percentage decrease in blue when compared to the mean recall on clean images. †We evaluate these methods using the codes provided by the authors.

	Method	gaus	shot	imp	dfcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	sun	wtd	smk	rain	dust	Average mR
mR@20: C/UC	GB-Net† [85]	15.2	16.0	15.2	16.9	14.9	16.5	16.6	17.9	18.9	21.4	21.6	14.7	16.8	16.6	18.2	16.7	17.8	16.0	20.1	18.5	17.3 (-27.3%)
	EB-Net† [9]	28.0	29.8	27.4	31.2	26.5	30.3	30.5	32.1	33.2	35.8	36.3	27.3	30.3	27.0	30.6	30.6	30.7	33.7	35.6	30.1	30.9 (-22.4%)
	HiKER-SGG	**31.1**	**33.3**	**31.5**	**35.4**	**28.5**	**35.0**	**34.1**	**36.5**	**37.7**	**39.8**	**40.8**	**30.5**	**33.7**	**31.3**	**34.2**	**33.5**	**34.9**	**37.1**	**39.8**	**32.6**	**34.6 (-17.8%)**
	GB-Net† [85]	10.3	10.6	10.4	11.6	10.4	10.9	10.7	11.9	12.3	13.7	13.8	10.0	11.1	10.8	11.7	11.1	11.2	10.5	13.0	12.1	11.4 (-25.5%)
	EB-Net† [9]	21.7	22.8	20.4	24.9	19.6	23.2	23.8	23.2	24.6	27.5	28.0	20.1	23.1	21.1	23.6	24.0	23.4	25.6	27.3	22.9	23.5 (-23.7%)
	HiKER-SGG	**24.8**	**25.8**	**24.8**	**27.5**	**22.4**	**27.4**	**26.4**	**27.8**	**28.7**	**31.1**	**31.5**	**23.3**	**26.0**	**24.3**	**26.5**	**26.3**	**26.8**	**28.5**	**30.9**	**24.9**	**26.8 (-19.8%)**
mR@50: C/UC	GB-Net† [85]	27.5	28.7	27.6	30.8	26.4	29.8	29.9	31.9	33.8	37.2	37.6	26.3	29.9	30.0	33.0	29.5	32.3	28.7	35.8	32.8	31.0 (-24.6%)
	EB-Net† [9]	42.1	43.7	41.5	44.9	40.2	45.6	44.2	46.9	47.7	50.4	51.2	41.2	44.1	41.4	45.1	45.4	45.5	48.4	49.7	44.6	45.2 (-17.7%)
	HiKER-SGG	**46.7**	**48.4**	**46.9**	**50.2**	**43.2**	**49.6**	**48.3**	**51.3**	**52.5**	**55.1**	**55.9**	**45.0**	**48.1**	**46.0**	**49.9**	**48.6**	**50.0**	**52.4**	**54.8**	**47.0**	**49.5 (-14.5%)**
	GB-Net† [85]	13.3	13.6	13.3	15.1	13.6	14.1	14.0	15.4	15.6	17.4	17.5	13.0	14.5	14.4	15.2	14.5	14.6	13.6	16.6	15.4	14.7 (-24.2%)
	EB-Net† [9]	24.8	27.6	25.6	28.3	25.9	28.9	29.4	29.3	30.5	32.0	32.8	26.1	28.6	26.3	27.9	29.2	28.6	30.8	31.8	27.2	28.6 (-22.1%)
	HiKER-SGG	**30.1**	**31.7**	**30.4**	**33.2**	**28.3**	**33.3**	**32.1**	**34.1**	**34.4**	**37.3**	**37.4**	**28.8**	**31.7**	**30.1**	**32.9**	**32.5**	**32.2**	**34.5**	**36.7**	**30.4**	**32.6 (-17.0%)**
mR@100: C/UC	GB-Net† [85]	40.1	41.9	40.1	43.8	37.8	42.9	42.7	45.1	47.1	50.8	51.7	37.8	42.8	42.9	46.6	42.5	46.1	41.2	49.6	45.9	44.0 (-20.6%)
	EB-Net† [9]	54.7	56.0	52.9	56.8	52.4	55.6	55.3	58.4	59.9	61.6	61.1	53.3	55.0	54.3	57.7	56.4	57.6	59.0	60.7	54.8	56.7 (-14.5%)
	HiKER-SGG	**59.3**	**60.3**	**58.6**	**62.3**	**55.6**	**61.9**	**59.8**	**63.4**	**64.0**	**66.9**	**67.4**	**56.4**	**60.1**	**58.4**	**62.3**	**59.8**	**62.1**	**63.7**	**66.3**	**58.9**	**61.4 (-11.3%)**
	GB-Net† [85]	14.8	15.1	14.6	16.6	15.1	15.6	15.6	16.9	17.1	19.1	19.0	14.4	16.0	16.0	16.8	16.1	16.1	15.0	18.1	17.0	16.3 (-22.0%)
	EB-Net† [9]	28.7	30.1	27.8	31.9	27.1	31.1	30.5	32.8	32.4	36.1	35.7	28.2	30.9	28.4	30.9	31.4	31.0	31.8	33.9	29.6	31.0 (-20.9%)
	HiKER-SGG	**32.7**	**33.8**	**32.6**	**36.0**	**30.4**	**35.7**	**34.7**	**36.3**	**36.7**	**39.9**	**39.7**	**31.1**	**34.2**	**32.7**	**35.4**	**34.9**	**35.4**	**37.1**	**39.2**	**32.6**	**35.1 (-14.8%)**

images in the Visual Genome [38] dataset. With the hierarchical predicate prediction paradigm, our method consistently outperforms the knowledge graph-based GB-Net [85] and EB-Net + EOA [9] methods. When compared with other state-of-the-art SGG methods, our HiKER-SGG still achieves competitive performance in terms of mean recall.

We also show our results on the VG-C dataset in Table 2 to demonstrate our method also generalizes well to unseen real-world corruptions. Specifically, we compare our performance with that of the knowledge graph-based methods across all six metrics. Table 2 illustrates that our method achieves an average improvement of around 4% across all six metrics for all 20 types of corruption. Moreover, relative to the clean image benchmark, our method exhibits a lower percentage of performance degradation, showcasing our model's resilience in handling such corrupted scenarios. For instance, in the presence of impulse noise corruption, our mR@20, when considering graph constraints, experiences an 8.6% reduction, dropping from 33.4% to 24.8%. In comparison, the EB-Net [9] method shows a greater 10.4% degradation, decreasing from 30.8% to 20.4%.

Qualitative Results. To provide further insights into the effectiveness of our method, we visualize some scene graphs generated by our method and the baseline GB-Net [85] method, under both clean and corrupted scenarios in Figure 3. In the upper left section of the image, we can observe the scene graphs generated by both methods on the clean image. Notably, while GB-Net tends to predict more general predicate classes (*e.g.*, `on`), our method accurately predicts the ⟨`train-has-engine`⟩ and ⟨`logo-in-train`⟩ triplets.

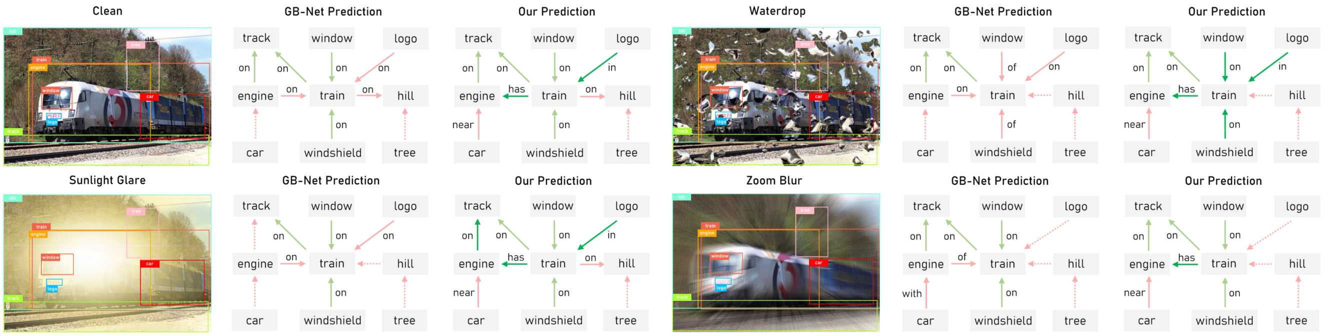


Figure 3. **Qualitative comparisons on the PredCls task**. The visualized predicted predicates are picked from the top 50 predicted triplets. Here, red dashed lines denote undetected predicates, solid red lines denote incorrect predictions, and solid green lines indicate correct predictions. For an easier comparison, predicates correctly predicted by our method but incorrectly by GB-Net are highlighted in dark green.

Table 3. **Ablation studies on the PredCls task using VG dataset**. PH and EH refer to predicate and entity hierarchical prediction heads respectively, and $\mathcal{M}/\mathcal{D}$ indicate whether these hierarchies are manually configured ($\mathcal{M}$) following Zellers *et al.* [87] or discovered ($\mathcal{D}$) by hierarchical clustering. AR refers to adaptive refinement.

PH	EH	AR	mR@20: UC/C	mR@50: UC/C	mR@100: UC/C
✗	✗	✗	39.8 / 30.8	54.9 / 36.7	66.3 / 39.2
✗	✗	✓	40.4 / 31.4	55.7 / 37.2	67.1 / 39.8
$\mathcal{M}$	✗	✗	41.6 / 32.9	57.3 / 37.5	68.1 / 39.6
$\mathcal{M}$	$\mathcal{M}$	✗	41.4 / 33.1	57.6 / 37.9	68.2 / 39.7
$\mathcal{M}$	$\mathcal{M}$	✓	41.8 / 33.2	57.7 / 38.1	68.7 / 40.0
$\mathcal{D}$	$\mathcal{D}$	✗	41.7 / 33.2	57.7 / 38.8	69.0 / 40.4
$\mathcal{D}$	$\mathcal{D}$	✓	**42.1 / 33.4**	**57.9 / 39.3**	**69.2 / 41.2**

We also illustrate the SGG results under sun glare, waterdrop, and zoom blur corruptions obtained by both methods in Figure 3. In these challenging scenarios, non-hierarchical GB-Net [85], struggles to detect the relation since the region feature is corrupted. In comparison, our method firstly determines the superclass relation rather than directly proceeding to subclass classification. This strategy enhances the robustness of our proposed method, enabling it to consistently generate a similar scene graph as in clean images.

4.3. Ablation Studies

Effectiveness of Each Component. To systematically analyze the impacts of different components in HiKER-SGG, we conduct an ablation study on the Visual Genome [38] dataset in Table 3. We have the following key observations: (1) The inclusion of the hierarchical inference process for predicate alone enhances the mR@k by 1.0%, and adding the hierarchical inference process for entity further boosts mR@k by an additional 0.5%; (2) Replacing manually configured hierarchical structures with those discovered ones yields a non-trivial 0.4%∼0.7% increase in mR@k; (3) Implementing the adaptive refinement contributes to a further improvement in performance by 0.2%∼0.8% mR@k.

Hyperparameter Analysis for α. We conduct experiments with five distinct values for the hyperparameter α and report the mR under the PredCls setting in Table 4. We can observe that our setting of $\alpha = 0.9$ yields the highest performance. The reason may be that this optimal value effectively balances the surface-level and deeper biases among the predicate and entity classes, which contributes to the improved unbiased prediction capabilities of our HiKER-SGG model.

Table 4. Hyperparameter analysis for α in Equation (20).

Value of α	mR@50	mR@100
$\alpha = 0.5$	56.7 / 38.1	66.9 / 40.0
$\alpha = 0.8$	57.4 / 38.5	68.5 / 40.7
$\alpha = 0.9$	**57.9 / 39.3**	**69.2 / 41.2**
$\alpha = 0.95$	57.6 / 38.9	69.1 / 40.9
$\alpha = 0.99$	57.6 / 38.7	68.8 / 40.5

Table 5. Training time and parameter count of HiKER-SGG compared with other methods.

Method	Training	# params
KERN [8]	179.1 min	405.2M
GB-Net [85]	84.6 min	444.6M
EB-Net [9]	89.7 min	448.8M
HiKER-SGG	101.3 min	455.9M

Efficiency Comparison. We also compare the training time and the number of parameters of our HiKER-SGG with other methods in Table 5. Our HiKER-SGG divides a general classifier into multiple smaller hierarchical classifiers, thereby maintaining relatively high efficiency compared to non-hierarchical methods such as GB-Net [85] and EB-Net [9]. Specifically, while incorporating only 7M additional parameters and extending the training time by only 12 minutes per epoch, our HiKER-SGG exhibits significantly enhanced robustness with both clean and corrupted images.

5. Conclusion

In this work, we first introduce a novel task, robust SGG in the presence of real-world corruptions. To address the challenge of interpreting visual scenes with corruptions, we then propose the **Hi**erarchical **K**nowledge **E**nhanced **R**obust **S**cene **G**raph **G**eneration (HiKER-SGG) framework. HiKER-SGG is corruption-agnostic, trained exclusively on clean images yet tested on corrupted ones without further training. It leverages hierarchical knowledge from external sources and a hierarchical prediction head, serving as an algorithmic prior for decision-making, to effectively reason and correct inaccuracies. Moreover, we developed a corrupted Visual Genome (VG-C) benchmark with 20 different corruptions to standardize and evaluate SGG robustness. Through extensive experiments, we have demonstrated that HiKER-SGG outperforms the state-of-the-art models on both clean and corrupted images.

Acknowledgement

This work has been funded in part by the Army Research Laboratory (ARL) under grant W911NF-23-2-0007 and W911NF-19-2-0146, and the Air Force Office of Scientific Research (AFOSR) under grants FA9550-18-1-0097 and FA9550-18-1-0251.

References

[1] Aniket Agarwal, Ayush Mangal, et al. Visual relationship detection using scene graphs: A survey. *arXiv preprint arXiv:2005.08045*, 2020. 3

[2] Sarthak Bhagat, Simon Stepputtis, Joseph Campbell, and Katia Sycara. Knowledge-guided short-context action anticipation in human-centric videos. *arXiv preprint arXiv:2309.05943*, 2023. 3

[3] Sarthak Bhagat, Simon Stepputtis, Joseph Campbell, and Katia Sycara. Sample-efficient learning of novel visual concepts. In *CoLLAs*, pages 637–657. PMLR, 2023. 3

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 3

[5] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *IEEE TPAMI*, 45(1):1–26, 2021. 1, 2

[6] Chao Chen, Yibing Zhan, Baosheng Yu, Liu Liu, Yong Luo, and Bo Du. Resistance training using prior bias: toward unbiased scene graph generation. In *AAAI*, pages 212–220, 2022. 6, 7

[7] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *CVPR*, pages 2624–2632, 2019. 1

[8] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, pages 6163–6171, 2019. 3, 8, 15

[9] Zhanwen Chen, Saed Rezayi, and Sheng Li. More knowledge, less bias: Unbiasing scene graph generation with explicit ontological adjustment. In *WACV*, pages 4023–4032, 2023. 3, 6, 7, 8, 14, 15

[10] Meng-Jiun Chiou, Henghui Ding, Hanshu Yan, Changhu Wang, Roger Zimmermann, and Jiashi Feng. Recovering the unbiased scene graphs from the biased ones. In *ACM MM*, pages 1581–1590, 2021. 6, 7

[11] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014. 5

[12] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *CVPR*, pages 3076–3086, 2017. 2

[13] Youming Deng, Yansheng Li, Yongjun Zhang, Xiang Xiang, Jian Wang, Jingdong Chen, and Jiayi Ma. Hierarchical memory learning for fine-grained scene graph generation. In *ECCV*, pages 266–283. Springer, 2022. 6, 7

[14] Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. In *ICCV*, pages 15404–15413, 2021. 1

[15] Xingning Dong, Tian Gan, Xuemeng Song, Jianlong Wu, Yuan Cheng, and Liqiang Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *CVPR*, pages 19427–19436, 2022. 2

[16] SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *NeurIPS*, 2016. 1

[17] Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted image modeling for self-supervised visual pre-training. In *ICLR*, 2023. 3

[18] Azade Farshad, Yousef Yeganeh, Yu Chi, Chengzhi Shen, Böjrn Ommer, and Nassir Navab. Scenegenie: Scene graph guided diffusion models for image synthesis. In *ICCV*, pages 88–98, 2023. 2

[19] Arushi Goel, Basura Fernando, Frank Keller, and Hakan Bilen. Not all relations are equal: Mining informative labels for scene graph generation. In *CVPR*, pages 15596–15606, 2022. 3, 6, 7

[20] Nicholas Gray, Megan Moraes, Jiang Bian, Alex Wang, Allen Tian, Kurt Wilson, Yan Huang, Haoyi Xiong, and Zhishan Guo. Glare: A dataset for traffic sign detection in sun glare. *IEEE TITS*, 2023. 1

[21] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, pages 1969–1978, 2019. 2, 3

[22] Yuyu Guo, Lianli Gao, Xuanhan Wang, Yuxuan Hu, Xing Xu, Xu Lu, Heng Tao Shen, and Jingkuan Song. From general to specific: Informative scene graph generation via balance adjustment. In *ICCV*, pages 16383–16392, 2021. 3, 5, 6, 7

[23] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *ICCV*, pages 10203–10212, 2019. 1

[24] Xianjing Han, Xingning Dong, Xuemeng Song, Tian Gan, Yibing Zhan, Yan Yan, and Liqiang Nie. Divide-and-conquer predictor for unbiased scene graph generation. *IEEE TCSVT*, 32(12):8611–8622, 2022. 2

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[26] Tao He, Lianli Gao, Jingkuan Song, Jianfei Cai, and Yuan-Fang Li. Learning from the scene and borrowing from the rich: tackling the long tail in scene graph generation. In *IJCAI*, pages 587–593, 2021. 3

[27] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. State-aware compositional learning toward unbiased training for scene graph generation. *IEEE TIP*, 32:43–56, 2022. 2

[28] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2018. 3, 6, 14, 15, 16

[29] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2019. 3

[30] Charles Herrmann, Kyle Sargent, Lu Jiang, Ramin Zabih, Huiwen Chang, Ce Liu, Dilip Krishnan, and Deqing Sun. Pyramid adversarial training improves vit performance. In *CVPR*, pages 13419–13429, 2022. 3

[31] Junhua Jia, Xiangqian Ding, Shunpeng Pang, Xiaoyan Gao, Xiaowei Xin, Ruotong Hu, and Jie Nie. Image captioning based on scene graphs: A survey. *Expert Systems with Applications*, page 120698, 2023. 2

[32] Bowen Jiang and Camillo J Taylor. Scene graph generation from hierarchical relationship reasoning. *arXiv preprint arXiv:2303.06842*, 2023. 2

[33] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015. 2

[34] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, pages 1219–1228, 2018. 2

[35] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967. 3

[36] Deunsol Jung, Sanghyun Kim, Won Hwa Kim, and Minsu Cho. Devil's on the edges: Selective quad attention for scene graph generation. In *CVPR*, pages 18664–18674, 2023. 6, 7

[37] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. In *UAI*, pages 1012–1021. PMLR, 2022. 3

[38] 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. 4, 5, 6, 7, 8, 12, 14, 15

[39] Stan Weixian Lei, Difei Gao, Jay Zhangjie Wu, Yuxuan Wang, Wei Liu, Mengmi Zhang, and Mike Zheng Shou. Symbolic replay: Scene graph as prompt for continual learning on vqa task. In *AAAI*, pages 1250–1259, 2023. 2

[40] 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. 6, 7

[41] 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. 3

[42] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, pages 11109–11119, 2021. 2

[43] Wei Li, Haiwei Zhang, Qijie Bai, Guoqing Zhao, Ning Jiang, and Xiaojie Yuan. Ppdl: Predicate probability distribution based loss for unbiased scene graph generation. In *CVPR*, pages 19447–19456, 2022. 6, 7

[44] Xiangyang Li and Shuqiang Jiang. Know more say less: Image captioning based on scene graphs. *IEEE TMM*, 21(8): 2117–2130, 2019. 2

[45] Xingchen Li, Long Chen, Jian Shao, Shaoning Xiao, Songyang Zhang, and Jun Xiao. Rethinking the evaluation of unbiased scene graph generation. In *BMVC*, 2022. 2

[46] Xinghang Li, Di Guo, Huaping Liu, and Fuchun Sun. Embodied semantic scene graph generation. In *CoRL*, pages 1585–1594. PMLR, 2022. 1

[47] Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *ICLR*, 2016. 3, 5

[48] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *ECCV*, pages 335–351. Springer, 2018. 2

[49] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 4

[50] Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. In *NeurIPS*, pages 3571–3583, 2021. 3

[51] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *CVPR*, pages 14765–14775, 2022. 3

[52] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 3, 4

[53] Tianwen Qian, Jingjing Chen, Shaoxiang Chen, Bo Wu, and Yu-Gang Jiang. Scene graph refinement network for visual question answering. *IEEE TMM*, 25:3950–3961, 2023. 2

[54] Yuhui Quan, Shijie Deng, Yixin Chen, and Hui Ji. Deep learning for seeing through window with raindrops. In *ICCV*, pages 2463–2471, 2019. 1

[55] 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. PMLR, 2021. 3

[56] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, page 91–99, 2015. 2, 3, 4, 6, 14

[57] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *ECCV*, pages 53–69. Springer, 2020. 3

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 6, 14

[59] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, page 4444–4451, 2017. 4

[60] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal.

Energy-based learning for scene graph generation. In *CVPR*, pages 13936–13945, 2021. 6, 7

[61] Shuzhou Sun, Shuaifeng Zhi, Qing Liao, Janne Heikkilä, and Li Liu. Unbiased scene graph generation via two-stage causal modeling. *IEEE TPAMI*, 2023. 2

[62] 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. 6, 7

[63] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, pages 3716–3725, 2020. 2

[64] Yushun Tang, Qinghai Guo, and Zhihai He. Cross-inferential networks for source-free unsupervised domain adaptation. In *ICIP*, pages 96–100. IEEE, 2023. 3

[65] Yushun Tang, Ce Zhang, Heng Xu, Shuoshuo Chen, Jie Cheng, Luziwei Leng, Qinghai Guo, and Zhihai He. Neuro-modulated hebbian learning for fully test-time adaptation. In *CVPR*, pages 3728–3738, 2023. 3

[66] Hongshuo Tian, Ning Xu, An-An Liu, Chenggang Yan, Zhendong Mao, Quan Zhang, and Yongdong Zhang. Mask and predict: Multi-step reasoning for scene graph generation. In *ACM MM*, pages 4128–4136, 2021. 2

[67] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul De Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *IJCV*, 129:341–360, 2021. 1

[68] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *CVPR*, pages 3961–3970, 2020. 1

[69] Jingyi Wang, Can Zhang, Jinfa Huang, Botao Ren, and Zhidong Deng. Improving scene graph generation with superpixel-based interaction learning. In *ACM MM*, pages 1809–1820, 2023. 6, 7

[70] Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *WACV*, pages 1508–1517, 2020. 2

[71] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In *CVPR*, pages 8188–8197, 2019. 2

[72] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Sketching image gist: Human-mimetic hierarchical scene graph generation. In *ECCV*, pages 222–239. Springer, 2020. 1

[73] Yang Wu, Pengxu Wei, and Liang Lin. Scene graph to image synthesis via knowledge consensus. In *AAAI*, pages 2856–2865, 2023. 2

[74] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434. Springer, 2018. 1

[75] 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, 6, 7

[76] Li Xu, Haoxuan Qu, Jason Kuen, Jiuxiang Gu, and Jun Liu. Meta spatio-temporal debiasing for video scene graph generation. In *ECCV*, pages 374–390. Springer, 2022. 2

[77] Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In *ACM MM*, pages 265–273, 2020. 2, 6, 7

[78] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, pages 670–685. Springer, 2018. 2

[79] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, pages 10685–10694, 2019. 2

[80] Yuan Yang, James C Kerce, and Faramarz Fekri. Logicdef: An interpretable defense framework against adversarial examples via inductive scene graph reasoning. In *AAAI*, pages 8840–8848, 2022. 3

[81] Keren Ye and Adriana Kovashka. Linguistic structures as weak supervision for visual scene graph generation. In *CVPR*, pages 8289–8299, 2021. 3

[82] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *NeurIPS*, pages 13276–13286, 2019. 3

[83] Sangwoong Yoon, Woo Young Kang, Sungwook Jeon, SeongEun Lee, Changjin Han, Jonghun Park, and Eun-Sol Kim. Image-to-image retrieval by learning similarity between scene graphs. In *AAAI*, pages 10718–10726, 2021. 2

[84] Jing Yu, Yuan Chai, Yujing Wang, Yue Hu, and Qi Wu. Cogtree: Cognition tree loss for unbiased scene graph generation. In *IJCAI*, pages 1274–1280, 2021. 6, 7

[85] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *ECCV*, pages 606–623. Springer, 2020. 3, 4, 6, 7, 8, 14, 15

[86] Alireza Zareian, Zhecan Wang, Haoxuan You, and Shih-Fu Chang. Learning visual commonsense for robust scene graph generation. In *ECCV*, pages 642–657. Springer, 2020. 3

[87] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018. 1, 3, 4, 6, 7, 8, 14

[88] Cheng Zhang, Wei-Lun Chao, and Dong Xuan. An empirical study on leveraging scene graphs for visual question answering. In *BMVC*, 2019. 2

[89] Ce Zhang, Simon Stepputtis, Joseph Campbell, Katia Sycara, and Yaqi Xie. Robust hierarchical scene graph generation. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*, 2023. 1

[90] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 3

[91] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. In *NeurIPS*, pages 38629–38642, 2022. 3

[92] 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. 6, 7

[93] Yifeng Zhu, Jonathan Tremblay, Stan Birchfield, and Yuke Zhu. Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs. In *ICRA*, pages 6541–6548. IEEE, 2021. 1

HiKER-SGG: Hierarchical Knowledge Enhanced Robust Scene Graph Generation

Supplementary Material

In this supplementary document, we provide additional details and experimental results to enhance understanding and insights into our proposed HiKER-SGG. This supplementary document is organized as follows:

- The hierarchical clustering process and visualization results in Section 3.2 are shown in Section A.1.
- A more detailed description for both the scene graph and the hierarchical knowledge graph in Section 3.3-3.5 is shown in Section A.2.
- More discussions about the adaptive refinement in Section 3.7 is in Section A.3.
- We provide a more comprehensive description of the experimental settings used in our study in Section B.1.
- A detailed description and visualization of the VG-C dataset are provided in Section B.2.
- We also present more experimental results in Section B.3 and B.4.
- Finally, we discuss the potential limitations of the proposed HiKER-SGG in Section C.

A. More Details about HiKER-SGG

A.1. Hierarchical Clustering

As we introduced in Section 3.2, we use a hierarchical clustering algorithm to discover both the entity and predicate hierarchies. We provide the pseudocode in Algorithm 1. Specifically, hierarchical clustering initializes with individual class names as separate clusters and repeatedly merges the two clusters with the highest similarity until only one cluster remains. During each iteration, it updates the similarity measures of the newly formed cluster with the remaining clusters, ensuring that the most similar clusters are merged at each step. This process results in a hierarchical structure of clusters based on the defined similarity metric.

After completing the hierarchical clustering, we select the three lowest-level clusters to conduct our hierarchical inference process defined in Section 3.6. The discovered predicate and entity hierarchies are visualized in Figure A1. Notably, the discovered hierarchical structure reasonably clusters similar classes in the same superclass, for example, (1) `wearing` and `wears` in predicate classes, and (2) `boy`, `girl`, `child`, and `kid` in entity classes. Although most hierarchical relationships are accurately identified, some may appear noisy from a human perspective. Nevertheless, given that our clustering is based on pre-defined similarity metrics, we do not perform additional cleaning and believe our model is equipped to handle these issues.

In Section 4.3, we have shown that replacing manually configured hierarchical structures with those discovered ones yields a non-trivial 0.4%∼0.7% increase in mR@k metrics. These results demonstrate that the hierarchies uncovered by this method, provide more effective guidance for our hierarchical inference approach.

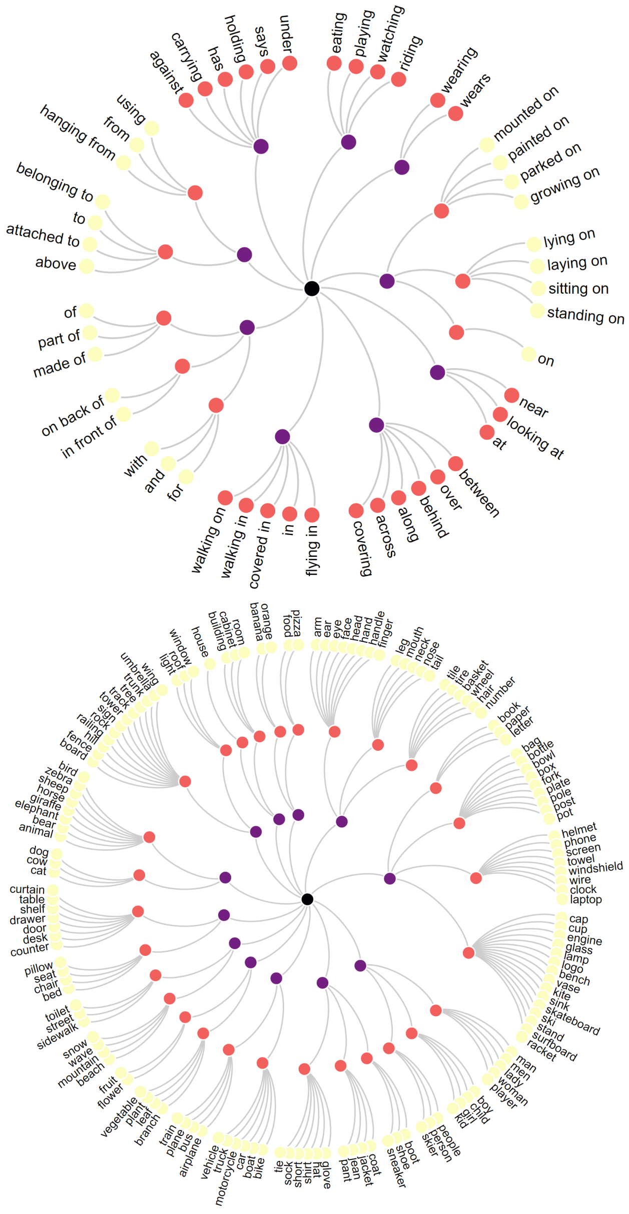


Figure A1. **The visualization of the discovered hierarchies on the Visual Genome [38] dataset**. Top: the discovered hierarchy for 50 predicate classes; Bottom: the discovered hierarchy for 150 entity classes. In this work, we simply utilize 3-level hierarchies for the hierarchical inference process.

Algorithm 1: Hierarchical Clustering Algorithm

```
Input: Category set $\mathcal{C} = \{\mathcal{C}_i\}_{i=1}^{n}$; Similarity metric
    $\mathsf{Sim}(\cdot,\cdot)$ defined in Section 3.2.
Output: The hierarchy of clusters $L$.
Initialize the clusters $L$ with $n$ clusters, each
  containing a class name;
repeat
    Find pairs of clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ in $L$ with highest
      similarity $\mathsf{Sim}(\mathcal{C}_1, \mathcal{C}_2)$;
    Merge $\mathcal{C}_1$ and $\mathcal{C}_2$ into a new cluster $\mathcal{C}_{12}$;
    Remove $\mathcal{C}_1$ and $\mathcal{C}_2$ from $L$;
    for each cluster $\tilde{\mathcal{C}} \in L$ do
        Update the similarity of the created cluster
          with other clusters with $\mathsf{Sim}(\mathcal{C}_{12}, \tilde{\mathcal{C}})$;
    end
    Add this new cluster $\mathcal{C}_{12}$ to $L$;
until $|L| = 1$;
```

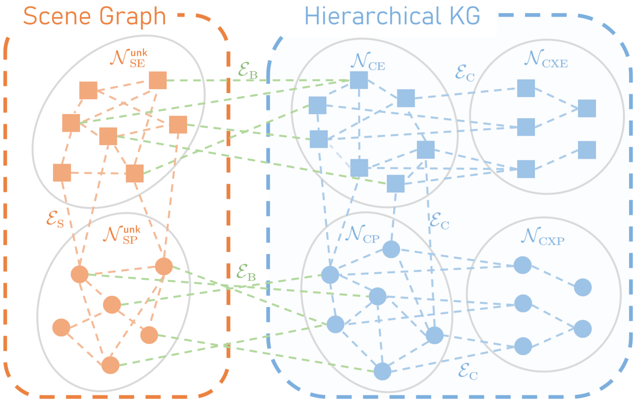

Scene Graph
Hierarchical KG
$\mathcal{N}_{\mathrm{SE}}^{\mathrm{unk}}$
$\mathcal{E}_{\mathrm{B}}$
$\mathcal{N}_{\mathrm{CE}}$
$\mathcal{E}_{\mathrm{C}}$
$\mathcal{N}_{\mathrm{CXE}}$
$\mathcal{E}_{\mathrm{S}}$
$\mathcal{N}_{\mathrm{SP}}^{\mathrm{unk}}$
$\mathcal{N}_{\mathrm{CP}}$
$\mathcal{E}_{\mathrm{C}}$
$\mathcal{N}_{\mathrm{CXP}}$
$\mathcal{E}_{\mathrm{B}}$
$\mathcal{E}_{\mathrm{C}}$

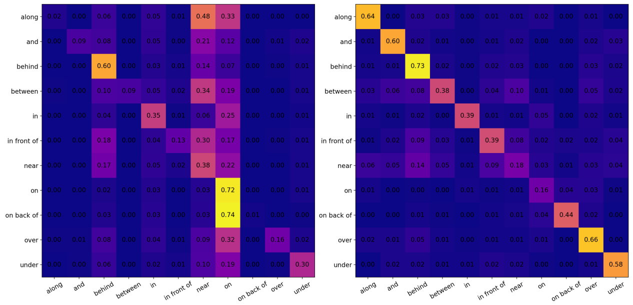
Figure A2. **The architecture and notations of our scene graph and hierarchical knowledge graph**. Nodes and edges within the scene graph are orange, those within the knowledge graph are blue, and the bridge edges that connect the two graphs are green.

A.2. Scene Graph and Hierarchical Knowledge Graph

In Figure A2, we summarize the architecture and notations of our scene graph and hierarchical knowledge graph we construct in this work. Specifically, we have 6 different types of nodes, as well as 3 types of edges. Below, we detail each one individually.

We have 6 different types of nodes:

- Commonsense entity node $\mathcal{N}_{\mathrm{CE}}$ in the knowledge graph. We only consider 150 entities from the VG dataset.
- Commonsense predicate node $\mathcal{N}_{\mathrm{CP}}$ in the knowledge graph. We only consider 50 predicates from the VG dataset.



	along	and	behind	between	in	in front of	near	on	on back of	over	under
along	0.02	0.00	0.06	0.00	0.05	0.01	0.48	0.33	0.00	0.00	0.00
and	0.00	0.09	0.08	0.00	0.05	0.00	0.21	0.12	0.00	0.01	0.02
behind	0.00	0.00	0.60	0.00	0.03	0.01	0.14	0.07	0.00	0.00	0.01
between	0.00	0.00	0.10	0.09	0.05	0.02	0.34	0.19	0.00	0.00	0.01
in	0.00	0.00	0.04	0.00	0.35	0.01	0.06	0.25	0.00	0.00	0.01
in front of	0.00	0.00	0.18	0.00	0.04	0.13	0.30	0.17	0.00	0.00	0.01
near	0.00	0.00	0.17	0.00	0.05	0.02	0.38	0.22	0.00	0.00	0.01
on	0.00	0.00	0.02	0.00	0.03	0.00	0.03	0.72	0.00	0.00	0.01
on back of	0.00	0.00	0.03	0.00	0.03	0.00	0.03	0.74	0.01	0.00	0.00
over	0.00	0.01	0.08	0.00	0.04	0.01	0.09	0.32	0.00	0.16	0.02
under	0.00	0.00	0.06	0.00	0.02	0.01	0.10	0.19	0.00	0.00	0.30

	along	and	behind	between	in	in front of	near	on	on back of	over	under
along	0.64	0.00	0.03	0.03	0.00	0.01	0.01	0.02	0.00	0.01	0.00
and	0.01	0.60	0.02	0.01	0.00	0.01	0.02	0.00	0.00	0.02	0.03
behind	0.01	0.01	0.73	0.02	0.00	0.02	0.03	0.00	0.00	0.03	0.02
between	0.03	0.06	0.08	0.38	0.00	0.04	0.10	0.01	0.00	0.05	0.02
in	0.01	0.01	0.02	0.00	0.39	0.01	0.01	0.05	0.00	0.02	0.01
in front of	0.01	0.02	0.09	0.03	0.01	0.39	0.08	0.02	0.02	0.02	0.04
near	0.06	0.05	0.14	0.05	0.01	0.09	0.18	0.03	0.01	0.03	0.04
on	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.16	0.04	0.03	0.01
on back of	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.04	0.44	0.02	0.00
over	0.02	0.01	0.05	0.01	0.00	0.01	0.01	0.01	0.00	0.66	0.00
under	0.00	0.02	0.02	0.01	0.01	0.02	0.02	0.00	0.00	0.01	0.58

Figure A3. **An illustration of adaptive semantic adjustment**. Left: the initial confusion matrix $\mathcal{R}^0$; Right: the confusion matrix $\mathcal{R}^5$ after 5 training epochs.

- Commonsense superclass entity node $\mathcal{N}_{\mathrm{CXE}}$ in the knowledge graph. This includes a set of specialized entity nodes at various levels, corresponding to overarching categories of entities.
- Commonsense superclass predicate node $\mathcal{N}_{\mathrm{CXP}}$ in the knowledge graph. This includes a set of specialized predicate nodes at various levels, corresponding to overarching categories of predicates.
- Scene entity node $\mathcal{N}_{\mathrm{SE}}$ in the scene graph. Derived from the commonsense entity node, each scene entity (SE) node $\mathcal{N}_{\mathrm{SE}}$ is additionally linked with a bounding box, *i.e.*, $\mathcal{N}_{\mathrm{SE}} \subseteq [0,1]^4 \times \mathcal{N}_{\mathrm{CE}}$.
- Scene predicate node $\mathcal{N}_{\mathrm{SP}}$ in the scene graph. Originating from the commonsense predicate node, each scene predicate (SP) node $\mathcal{N}$SP connects a pair of SE nodes, *i.e.*, $\mathcal{N}_{\mathrm{SP}} \subseteq \mathcal{N}_{\mathrm{SE}} \times \mathcal{N}_{\mathrm{SE}} \times \mathcal{N}_{\mathrm{CP}}$.

We also have 3 types of edges to connect these nodes:

- Commonsense edges $\mathcal{E}_{\mathrm{C}}$ in the knowledge graph. These edges within the commonsense graph $\mathcal{E}_{\mathrm{C}}$ delineate relationships between each node pair in both sets, functioning as a reservoir of general knowledge about objects. Examples include connections like `man-wears-shirt` and `cat-is-animal`.
- Scene edges $\mathcal{E}_{\mathrm{S}}$ in the scene graph. These edges encapsulate the relationships within the scene graph, linking scene entities and predicates to denote interactions and spatial relationships in a given scene.
- Bridge edges $\mathcal{E}_{\mathrm{B}}$ connecting commonsense nodes and scene nodes. These bi-directional bridge edges link an entity or predicate from the scene graph to its corresponding labels in the commonsense graph. Given the symmetric nature of the relation, the bridge edges are implemented as bi-directional directed edges with shared weights.

A.3. Adaptive Refinement

To provide insights into the adaptive refinement process introduced in Section 3.7, we present a visualization of the initial confusion matrix $\mathcal{R}^0$, along with the updated confusion matrix $\mathcal{R}^5$ after 5 training epochs in Figure A3. The initial confusion matrix $\mathcal{R}^0$ exhibits strong performance in general classes with more samples. In contrast, the updated version $\mathcal{R}^5$ strives for balanced accuracy across all predicate classes. Simultaneously, rather than the initial matrix which aims to reveal surface-level biases, the updated $\mathcal{R}^5$ shifts its focus to uncovering deeper and subtle correlations between predicate classes, as indicated by more minor values off the diagonal (*e.g.*, `near` with `along`/`and`/`between`).

B. More Experimental Results

B.1. Experiment Settings

Tasks. Following previous work [9, 85], we assess the effectiveness of our proposed approach in the context of two standard SGG tasks: Predicate Classification (PredCls) and Scene Graph Classification (SGCls). In the PredCls scenario, our model is provided with ground-truth bounding boxes and their associated object classes, with the sole task of predicting the predicate class. In the SGCls scenario, the model is only provided with known bounding boxes while the object classes are treated as unknown, and our SGG model is required to predict both the object and predicate classes.

Evaluation Metrics. We evaluate the performance of the SGG models by top-k mean triplet recall (mR@k) metric on both the PredCls and SGCls tasks. In specific, mR is the average recall score between the top-k predicted triplets and ground-truth ones across all 50 predicate categories, which promotes unbiased prediction for less frequently occurring predicate classes. A subject-predicate-object triplet is considered a match when all three components are correctly classified, and the subject and object bounding boxes align with an IoU (Intersection over Union) score of at least 0.5. In our experiments, we report the mean recall on $k = 20, 50, 100$ to comprehensively evaluate the effectiveness of our method. We also report the constrained (C) and unconstrained (UC) performance results, depending on the presence or absence of the graph constraint. This constraint restricts our SGG model to predict only a single relation between each pair of objects.

Implementation Details. We use the Faster-RCNN [56] as the object detector, which is based on VGG-16 [58] backbone provided by Zellers *et al.* [87]. In our experiments, we train our model for 30 epochs, initializing the learning rate at 1×10^{-4}. This learning rate will decrease to $1/10$ of its value after every 10 epochs. A single NVIDIA Quadro RTX 6000 GPU is used for all the experiments.

Fairness. To the best of our knowledge, our work is the first to tackle the robustness challenge in SGG, therefore there are no other established baselines for this task available. However, we do our best to ensure a fair comparison: all models rigorously follow the ***same*** evaluation protocol stated in Section 4. Our experiments are designed to highlight: (1) Compared to GB/EB-Net, HiKER-SGG enables a more comprehensive and efficient exploitation of KG information. (2) Compared to other methods, the performance gain demonstrates the effectiveness of KG in enhancing SGG. To ensure a fair comparison with non-graph-based methods, we also conduct an experiment that set the message propagation steps as $t = 0$ to isolate the effect of KG. In the PredCls tasks, the mR@50/100 accuracy remains competitive at 34.9%/37.1%.

Corruption Type	Abbreviation
Gaussian Noise	gaus
Shot Noise	shot
Impulse Noise	imp
Defocus Blur	dfcs
Glass Blur	gls
Motion Blur	mtn
Zoom Blur	zm
Snow	snw
Frost	frst
Fog	fg
Brightness	brt
Contrast	cnt
Elastic	els
Pixelate	px
JPEG Compression	jpg
Sunlight glare	sun
Water drop	wtd
Wildfire Smoke	smk
Rain	rain
Dust	dust

Table B1. **Abbreviations** of the 20 corruption types in our created corrupted Visual Genome (VG-C) benchmark.

B.2. Corrupted Visual Genome Benchmark

In addition to the clean Visual Genome dataset, we also evaluate our method on the corrupted Visual Genome [38] (VG-C) dataset, which comprises 20 versions of corrupted images designed to simulate realistic corruptions that may occur in real-world scenarios, thereby providing insights into the models' robustness under various corruption conditions. Of these corruptions, the first 15 types of corruption introduced by Hendrycks *et al.* [28] are widely recognized as standard benchmarks for evaluating robustness within the research community. To further align with real-world deployment scenarios, we introduce 5 additional types of *natural* corruptions to our evaluation:

- **Sunlight glare**: Sunlight glare refers to the interference caused by excessive sunlight or bright light sources in an image. It typically results in overexposed or washed-out areas in the photo, making it difficult to discern details and colors.

Table B2. **Multi-hop accuracy on the PredCls task using the Visual Genome [38] dataset**. We compare our method with EB-Net [9] method, assessing performance based on both level-1/2 superclass and final subclass accuracy.

	Setting	mR@20: UC/C	mR@50: UC/C	mR@100: UC/C
EB-Net	1-hop	51.6 / 50.5	68.2 / 63.7	79.4 / 68.1
	2-hop	45.4 / 40.2	62.8 / 48.9	73.7 / 52.0
	3-hop	39.8 / 30.8	54.9 / 36.7	66.3 / 39.2
Ours	1-hop	59.6 / 57.8	75.6 / 69.1	87.7 / 75.3
	2-hop	50.8 / 45.2	67.7 / 53.8	79.6 / 57.2
	3-hop	42.1 / 33.4	57.9 / 39.3	69.2 / 41.2

- **Water drop**: Water drop corruption occurs when water droplets or condensation obstruct the camera lens or affect the image sensor. This can create blurry or distorted portions of the image and often results in a hazy or unfocused appearance.
- **Wildfire smoke**: Wildfire smoke corruption pertains to images taken in areas affected by heavy smoke. It causes reduced visibility, a haze or smoky appearance, and can obscure objects in the frame.
- **Rain**: Rain refers to the presence of falling raindrops in an image. Rain can cause blurriness and distortions, making it difficult to see objects clearly.
- **Dust**: Dust corruption results from particles or dust settling on the camera lens or sensor. This can lead to the appearance of dark spots or specks in the image, which may obscure details and reduce clarity.

We establish 5 distinct severity levels for each corruption, following Hendrycks *et al.* [28] to facilitate future benchmarking. Table B1 presents a summary of the abbreviations used for the various types of corruption. To illustrate the effects of these corruptions, we present the corrupted versions of two example images in Figure B4.

We have already made the processing code for generating these corruptions available (VG-C benchmark) at https://github.com/zhangce01/HiKER-SGG. This benchmark offers a comprehensive evaluation platform to assess the robustness of SGG models in adverse conditions, and we encourage the formulation of new SGG models to be evaluated using this benchmark, emphasizing the real-world applications of the SGG task.

B.3. Multi-Hop Accuracy

To further illustrate the robustness of our method, we compare HiKER-SGG with EB-Net [9] by multi-hop mean recall metrics on the Visual Genome dataset in Table B2. Our evaluation criterion is as follows: a 1/2-hop prediction is considered correct if any of the final predicted predicate classes in the triplets correspond to the true level-1/2 superclass. By designing our model to predict from higher to lower levels, our HiKER-SGG not only achieves state-of-the-art performance in final subclass prediction, but also exhibits superior performance in 1/2-hop superclass prediction, outperforming the baseline method by an average of 8% and 5% in mean recall, respectively. This performance highlights that when unable to classify to the final subclass, HiKER-SGG tends to more accurately predict the superclass, illustrating the robustness of our hierarchical prediction approach.

Table B3. Training time, testing time, and parameter count of HiKER-SGG compared with other methods.

Method	Training (/epoch)	Inference (/image)	# params
KERN [8]	179.1 min	0.32 s	405.2M
GB-Net [85]	84.6 min	0.20 s	444.6M
EB-Net [9]	89.7 min	0.22 s	448.8M
HiKER-SGG	101.3 min	0.24 s	455.9M

B.4. Inference Time

In Section 4.3, we have shown that our HiKER-SGG exhibits significantly enhanced robustness with both clean and corrupted images with only about 10% training costs. In Table B3, we also include inference time form comparisons. A single NVIDIA Quadro RTX 6000 GPU is used for all the experiments. When compared to state-of-the-art methods such as GB-Net [85] and EB-Net [9], the HiKER-SGG model only extends the inference time by a slight 0.02-0.04 seconds. This minor increase is likely negligible in practical real-world deployment scenarios.

C. Limitations

We identify two potential limitations of our HiKER-SGG method: (1) For each new dataset, a hierarchical structure must be re-discovered, potentially increasing complexity. Additionally, the selection of similarity metrics also includes bias or the prior incorporation by humans. We acknowledge that the choice of measures does reflect a one-time prior human incorporation. However, once determined, the process becomes systematic. This is fundamentally different from the continuous, subjective interventions that characterize the human bias we aim to avoid. (2) Our method is tested in corrupted experiments on PredCls and SGCls tasks, assuming the accuracy of detected bounding boxes. However, in cases of severely corrupted images where the object detector fails to recognize objects, our HiKER-SGG method may not perform effectively. However, in our experiments, a simple Faster-RCNN is able to identify nearly 50% of the GT boxes even under corrupted scenarios; In contrast, given the GT boxes, SGG models can only achieve about 11% mR@100 in SGCls task. This highlights the practical significance of enhancing the robustness of SGG models. Besides, we also notice that there is another line of work and benchmarks (*e.g.*, Foggy Cityscapes) focusing on designing robust detectors. Combining our approach with them could further enhance the overall reliability of the system.

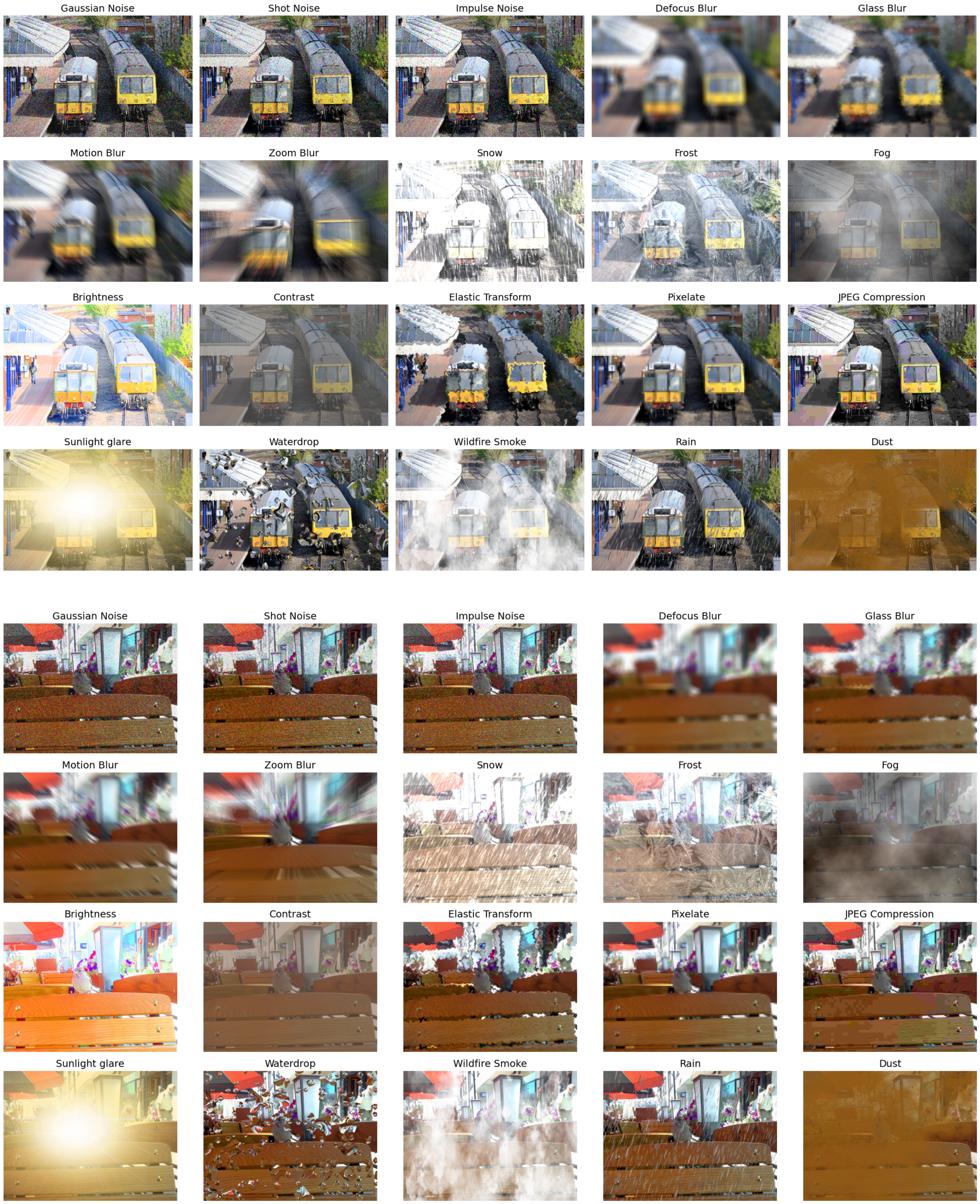


Figure B4. **All the 20 corruption types we used in our corrupted experiments**. The first 15 types of corruption are introduced by Hendrycks *et al.* [28], and we introduce 5 additional types of *natural* corruptions for a more comprehensive and practical evaluation.