IEICE TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2023EDP7228

Publicized:2024/04/30

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Enhanced Data Transfer Cooperating with Artificial Triplets for Scene Graph Generation

KuanChao CHU[†a)], **Satoshi YAMAZAKI**[††], ***Nonmembers***, ***and*** **Hideki NAKAYAMA**[†], ***Member***

SUMMARY This work focuses on training dataset enhancement of informative relational triplets for Scene Graph Generation (SGG). Due to the lack of effective supervision, the current SGG model predictions perform poorly for informative relational triplets with inadequate training samples. Therefore, we propose two novel training dataset enhancement modules: Feature Space Triplet Augmentation (FSTA) and Soft Transfer. FSTA leverages a feature generator trained to generate representations of an object in relational triplets. The biased prediction based sampling in FSTA efficiently augments artificial triplets focusing on the challenging ones. In addition, we introduce Soft Transfer, which assigns soft predicate labels to general relational triplets to make more supervisions for informative predicate classes effectively. Experimental results show that integrating FSTA and Soft Transfer achieve high levels of both Recall and mean Recall in Visual Genome dataset. The mean of Recall and mean Recall is the highest among all the existing model-agnostic methods.

key words: *scene graph, sgg, data transfer, feature space augmentation*

1. Introduction

Scene graphs have emerged as a pivotal representation for detailing semantic information within a visual scene, by specifying relationships between object pairs [1], [2]. This representation enables reasoning about visual content through the encoded spatial and logical details of object instances and their relations. In modern applications, scene graphs have become foundational for high-level visual tasks like activity parsing [3], image retrieval [1], visual understanding [4], and image captioning [5]. This paper delves into the scene graph generation (SGG) task, aiming to predict objects and their relations from visual input.

SGG models encounter two primary challenges when trained on common dataset [6]: first, the distinct long-tailed distribution of relations [7], [8], and second, the ambiguity caused by semantically similar relation classes (e.g., on/on back of/mounted on) [9]–[11]. The latter exacerbates the issue, as instances within a category may be annotated under multiple confusing classes. Such complexities often bias relation predictions in general SGG models, leading to low recall rates for rare predicate classes. While some unbiased SGG methods [7], [12]–[14] have addressed this, they often sacrifices performance on frequent classes. Hence, it is essential to consider these trade-offs to ensure the model performance on the majority of data is not compromised.

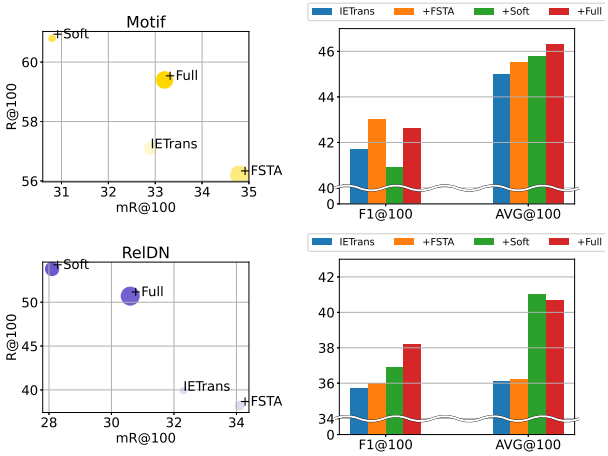

Fig. 1: Accuracy comparison between FSTA, Soft Transfer, Full, and the baseline IETrans on Motif (1^{st} row) and RelDN (2^{nd} row). In the scatter plots (left), a larger dot size and a darker color represent higher F1@100 and AVG@100 scores, respectively. As shown in the bar plots (right), increased scores in the overall metrics (F1@100 and AVG@100) indicate the alleviated performance trade-off in our full method, consisting of two complementary modules.

Recently, the training data modification approach has shown promising results for training an unbiased SGG model [9], [10]. Two major concepts for the modification are the addition of new predicate labels and reassignment of existing ones, which can efficiently improve rare class performance. We revisit these concepts through IETrans [10], a baseline data modification method. IETrans encompasses two modules: external transfer for label addition and internal transfer for label reassignment. Notably, the external transfer, while leveraging background triplets for augmentation, doesn't fully exploit the available data. Given the compositional nature of relational triplets, inter-triplet augmentation appears worthwhile. Additionally, predicate reassignments in the internal transfer are not uniformly reliable. A human evaluation study [10] reveals that only 76% of transferred triplets are deemed reliable. The inconsistency in the degree of semantic confusion, even among identical predicates, suggests that an "entire" transfer strategy might not be optimal. Guided by these findings, our system seeks to address these shortcomings by extending the modification concepts in two key ways: improving upon the data addition process and enhancing the reassignment efficiency.

[†]The authors are with the University of Tokyo, Tokyo, 113-8654 Japan.

[††]The author is with the NEC Corporation, Tokyo, 108-8001 Japan.

a) E-mail: kcchu@nlab.ci.i.u-tokyo.ac.jp

Our method introduces two novel modules: Feature Space Triplet Augmentation (FSTA) and Soft Transfer. FSTA dynamically creates artificial triplets during training. We can construct new data by enumerating triplet combinations `subject-predicate-object'` and `subject'-predicate-object` from a sampled mini-batch. Here, x' denotes data not from the original triplets. These artificial triplets serve to regularize the relation classification module in the SGG model. We undersample the frequent classes in artificial triplets to shape their predicate distribution. Further, a biased prediction-based sampler selects the class label for x'. This design aims to often sample combinations that are hard to be predicted correctly for a biased model. A pre-trained generator synthesizes the corresponding features based on class labels. Besides, Soft Transfer refines label reassignment by implementing an instance-wise ranking and mapping mechanism. We first compute a reliability score for each reassigned sample from biased model predictions, then select low-scoring triplets for Soft Transfer. Subsequently, a non-binary predicate label is calculated by mapping the reliability score, allowing for finer control over semantic confusion by using this label probability instead of an entire reassignment.

FSTA notably boosts performance on rare classes with increased sample quantity and diversity. Conversely, Soft Transfer alleviates performance loss in frequent classes, a typical compromise when elevating rare class performances. In essence, while FSTA contributes to mean recall (mR) gain, Soft Transfer leads to the recall (R) gain. Collectively, these modules bring reduced performance trade-off that shown in the improved overall metrics, F1@K and Avg@K. Our model-agnostic method was evaluated on the VisualGenome dataset [6], using two types of general SGG models: MOTIF [15] and RelDN [16] with IETrans. In the predcls task, our system outperforms the baseline IETrans by a 3.1% and 7.0% relative gain on the F1@100 metric for MOTIF and RelDN, respectively. Fig. 1 illustrates the balanced performance of our method.

To sum up, we make the following *contributions*:

1. We propose a novel, model-agnostic method for training a R/mR balanced SGG model. It integrates two complementary modules: FSTA and Soft Transfer, which enhance the baseline IETrans.
2. We conduct extensive experiments and discussions on VisualGenome and demonstrate the effectiveness of our system.

2. Related Work

2.1 Biased and Unbiased Scene Graph Generation

Scene Graph Generation (SGG) is first proposed as visual relation detection (VRD) [17], where each relation is independently detected, ignoring the rich contextual information. Later studies in SGG utilizes advanced techniques, e.g., message passing [18], recurrent sequential architectures [15] or contrastive learning [16]. However the accuracy of relationship detection is far from satisfaction due to the heavily biased data. Some authors [19], [20] point out that the predictions of current SGG models often collapse to several general and trivial predicate classes. Instead of only focusing on recall metric, hence they propose a new metric named mean recall, which is the average recall of all predicate classes, as the unbiased metric. Efforts towards developing unbiased SGG models have been noted. BGNN and DT2-ACBS [7], [21] proposes sophisticated re-sampling strategy. Some debiasing solutions [10], [12], [22] are categorized as biased-model based strategies that utilize predictions from biased SGG model. Especially, IETrans[10] adopts triplet-level data transfers over the less precise predicate-level manipulation. Our proposed method is inspired from IETrans, and focuses on data augmentation for inadequate training samples.

2.2 Compositional Learning

Recognition-By-Components theory [23] which illustrates that human representations of concepts are decomposable is especially influential in object recognition. Based on the theory, novel concepts from a few samples can be potentially learnable by composing known primitives. Some authors apply the compositional deep representation into few-shot learning for object recognition [24] and Human-Object Interaction (HOI) detection [25]–[27]. Visual compositional learning frameworks [26], [27] proposed for HOI detection compose HOI training samples from image-pairs and fake object representations to solve the open long-tail issue in HOI detection. Our proposed data augmentation method adapts the compositional learning to SGG task. To overcome the biased data issue in SGG, our sampling strategy of composed training samples plays an important role.

3. Methodology

A scene graph generation (SGG) model predicts a direct graph G for an input image $I \in \mathbb{R}^3$. $G = \{V, E\}$ contains a set of predicted objects $V = \{(\mathbf{b}_i, c_{e_i})\}_{i=1}^{N_V}$ and a set of predicted relationships $E = \{(s_j, c_{r_j}, o_j)\}_{j=1}^{N_E}$. $\mathbf{b}_i \in \mathbb{R}^4$ denotes the position of an object using bounded box coordinates. $c_{e_i} \in C_{objects}$ and $c_{r_j} \in C_{relations}$ belong to the known object and relation classes, respectively. $s_j \in V$ and $o_j \in V$ are nodes connected by the relation c_{r_j}. Each element in E can also be depicted as a *subject-predicate-object* triplet to convey the intrinsic semantic information.

Conceptually, an SGG model can be seen as a sequence of modules that includes an object detection backbone followed by a relation prediction head. The detection backbone first outputs a set of Region of Interest (RoIs) containing the detected object information. These results are then forwarded to the relation prediction head to refine these detections and predict the relation between the RoI pairs. This work mainly focuses on the scenario where a maximum of one predicate, with the highest score, can be predicted between each RoI pair. This principle aligns with the *graph constraint* mode

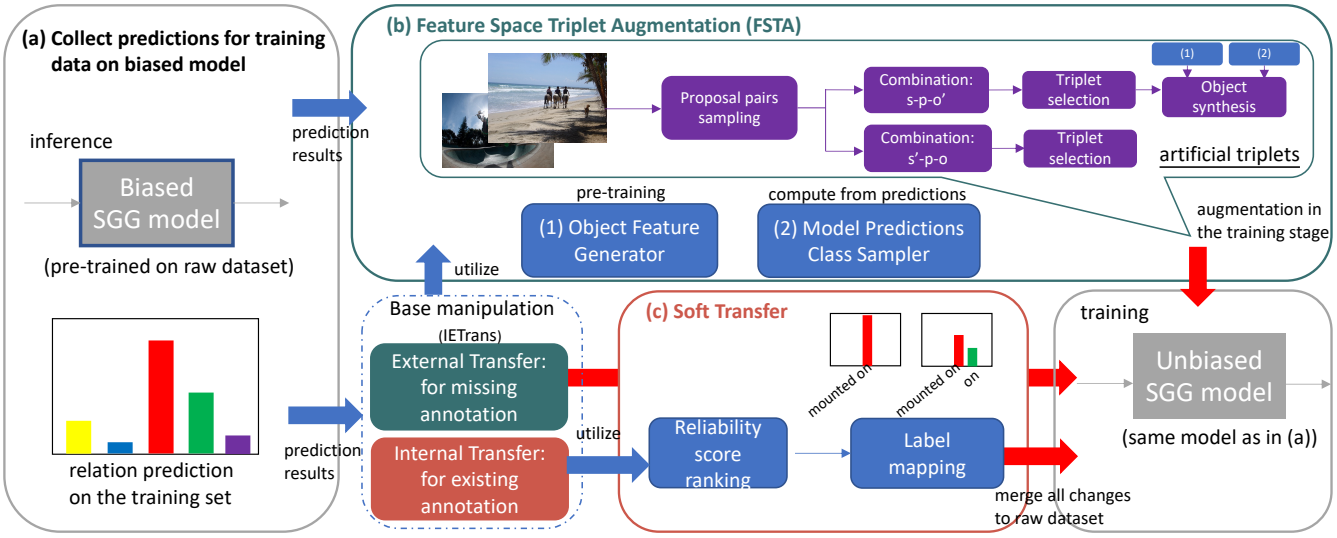

Fig. 2: The system overview of our proposed method. The FSTA and Soft Transfer modules are designed to introduce new concepts to enhance the baseline dataset manipulation module, IETrans. Blocks indicated in blue are prepared during the pre-processing stage, whereas the blocks in purple are designated for the unbiased SGG model training stage.

described in other research.

Figure 2 illustrates the system overview. In our enhanced data modification approach, we further leverage predictions from a pre-trained yet biased SGG model. Sec. 3.1 gives a brief overview of the baseline modification, IETrans. Sec. 3.2 details our FSTA module, elucidating a strategy for triplet augmentation in the feature space during the unbiased **training phase**. Sec. 3.3 explains Soft Transfer, a method offering precise control over reassigning predicate labels during the **pre-processing stage**, ensuring better handling of per-sample semantic confusion. Sec. 3.4 has our implementation details.

3.1 Preliminary Introduction: IETrans

IETrans constructs a modified training dataset during pre-processing. It has two steps: external transfer and internal transfer. In the external transfer, it acquires new labels from those no-relation object pairs. By ranking these no-relation prediction scores from the biased model, some object pairs are assigned new predicate labels that have the highest probability. This approach, however, may not fully leverage the available data. On the other hand, internal transfer shifts general predicates to informative ones using a ranking and affinity score filtering method through biased prediction results. For example, "man-on-horse" becomes "man-sitting on-horse". Nevertheless, the level of ambiguity is context-sensitive, and a binary transfer decision might not effectively capture semantic confusion across all samples.

These transfer steps explicitly adjust the balance of the dataset distribution, based on the property that a rare predicate class is often a more informative version of a frequent class. Linking general to informative predicate pairs reduced semantic ambiguity by discovering the confusion in biased model predictions. The raw frequently prior is employed to compensate the largely sacrificed performance on general predicates. We refer readers to the original publication for more details.

3.2 Feature Space Triplet Augmentation

Given the compositional nature of a relation triplet, it's possible to construct a new sample from multiple existing ones. The interaction between object-predicate representations in the feature space for SGG models is pivotal. Even though they can be combined in various ways—be it addition [16], concatenation [16], or element-wise multiplication [15]—the upstream feature extractor processes the elements in a triplet independently. As such, when the object representation in a triplet is partially changed to form a new semantically reasonable combination, the relation predictor in the relation head should be encouraged to produce similar outputs. This can be represented as:

$$M(F(\mathbf{f}_{s_i}, \mathbf{f}_{p_i}, \mathbf{f}_{o_i}; \theta_F); \theta_M) \approx M(F(\mathbf{f}_{s_i}, \mathbf{f}_{p_i}, \mathbf{f}_{o_j}; \theta_F); \theta_M) \quad (1)$$

where $(\mathbf{f}_{s_i}, \mathbf{f}_{p_i}, \mathbf{f}_{o_i})$ denotes the subject-relation-object intermediate representations of the i^{th} sample. Given that $i \neq j$ and $(c_{s_i}, c_{p_i}, c_{o_j})$ is a semantically reasonable triplet, $M(\cdot; \theta_M)$ is the final predicate classification module, and $F(\cdot; \theta_F)$ symbolizes the transitional layers in between. In light of this, artificial triplets can serve as augmented data to regularize the relation predictor during training.

We present feature space triplet augmentation (FSTA) via artificial triplets. Compared with generating new image samples, features are more tractable and computationally efficient without using external knowledge [28], [29]. For an input mini-batch of size N_B, the detector backbone yields RoIs of varying numbers with their object prediction results. We sample N_t RoI pairs per image from the pool of RoI

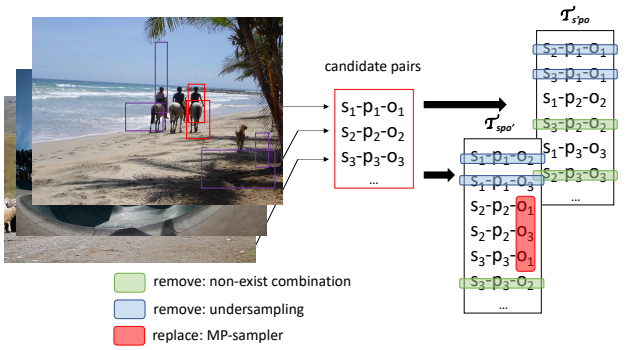

Fig. 3: Building combinations from batch input proposals. Purple box pairs are excluded for low IoU with ground-truth relations and red box pairs are selected as candidates.

pairs that have an overlap score exceeding s_{iou} with any of the ground-truth triplets in the image. Next, we enumerate a set $\mathcal{T}_{spo'}$ of combinations from these $N_B \times Nt$ triplets as `subject-predicate-object'`, where `object'` represents the object features from all other sampled triplets. After eliminating pairs that are absent from the training set label space, the remaining feature combinations—deemed to be reasonable—are forwarded to the same modules in the relation head. The resulting outputs are employed to compute a regularization term $\alpha\mathcal{L}_{at}$ to foster consistent predicate predictions on artificial triplets. We use the same loss function (i.e., cross entropy) as in the origin relation head for computing $\mathcal{L}_{at}$. Figure 3 visualizes our approach to building combinations for artificial triplets.

Besides generic artificial triplet synthesis, our FSTA module incorporates two novel features: (1) bi-directional resampling, and (2) Model prediction-based class sampler.

The bi-directional resampling further expands the volume of artificial triplets by enumerating `subject'-predicate-object` into a new set of combinations $\mathcal{T}_{s'po}$, where `sbject'` is sourced from other sampled triplets. As $\mathcal{T}_{s'po} \cap \mathcal{T}_{spo'} = \emptyset$, this enhances the richness and diversity of the artificial triplets. We define an undersampling parameter $U_h \in [0, 1]$ to govern the predicate distribution in artificial triplets. For triplets of frequent relations (termed as "head group"), we retain a random U_h fraction of them. This can effectively achieve a distribution shift toward rare relations in artificial triplets. Overall, We combine artificial triplets built from $\mathcal{T}_{spo'}$ and $\mathcal{T}_{s'po}$.

Moreover, we propose a new sampler based on biased model predictions (hence, MP-sampler) on training data. It targets to generate suitable `object'` classes rather than mere swaps. The motivation is straightforward: **Combinations that are difficult to be predicted correctly ought to be sampled more frequently**. To begin, we enumerate the candidate `object'` classes as $\mathcal{O}_{cand}$ from the dataset label space for a given `sbject-predicate` class label pair, (c_s, c_p). Then, we define a difficulty score function $d(\cdot)$ which computes the mean score discrepancy between the top-1 prediction and the ground-truth predicate class:

$$d(c_s, c_p, c_{o_i}) = max(l(c_s, c_p, c_{o_i})) - v(l(c_s, c_p, c_{o_i}), c_{o_i}) \quad (2)$$

where $o_i \in \mathcal{O}_{cand}$, and $l(\cdot) \in \mathbb{R}^{|\mathcal{C}_{relations}|}$ returns the average post-softmax predicate prediction vector for the input combination in MP. $v(l, i)$ obtains the value of l at element i. If the correct relation for a combination is often mispredicted, the difficulty score is positive; otherwise, it is zero. The MP-sampler then can generate an `object'` class following the probability:

$$p(c_{o_i}|(c_s, c_p)) = \frac{d(c_s, c_p, c_{o_i})}{\sum_{j=1}^{|\mathcal{O}_{cand}|} d(c_s, c_p, c_{o_j})} \quad (3)$$

In short, our emphasis primarily rests on those hard-to-predict combinations when building artificial triplets.

With MP-sampler, we use a generator to synthesize features for `object'`, as sampled classes are not assured to align with the classes from the batch's swapped features. We collect the ground-truth object features to train a conditional-GAN [30]. Following [31], [32], we define its adversarial loss function as:

$$\min_G \max_D \mathcal{L}_{wgangp} + \beta\mathcal{L}_{cls} + \gamma\mathcal{L}_{recon} \quad (4)$$

where $\mathcal{L}_{cls}$ and $\mathcal{L}_{recon}$ regularize the generator output via an object classifier and an reconstructor respectively, of both pre-trained on real data. Having the trained generator G, it is capable of yielding synthetic object features with MP-sampler, and we can construct artificial triplets for $\mathcal{T}_{spo'}$. The GAN model detail can be found in the appendix. Algorithm 1 outlines the complete procedures of FSTA.

3.3 Soft Transfer

The IETrans internal transfer reassigns relation labels from the general (source) ones to the informative (target) ones. However, some transfers are suboptimal: human evaluation deems only 76% of general-informative pairs as "reliable" [10]. While tail performance can benefit from these transfers, the cost of head performance drop is a concern.

A finer control on individual transfers could improve the transfer efficiency and thus alleviate the tail-head performance trade-off. Instead of a complete label transfer from a general ($p \to 0$) to an informative predicate ($p \to 1$), we propose the Soft Transfer that assigns non-binary probabilities to source and target predicate classes. Soft Transfer consists of two steps. Firstly, we rank all the reassigned pairs using a triplet-wise reliability score, from which pairs are selected for Soft Transfer. Second, a mapping function converts the reliability score into probabilities for source and target labels.

Based on the observation that transfer reliability varies from one combination to another, we define a preliminary function $r_{int}(\cdot)$ to estimate the degree of reliability. Given an transfer decision list, each list item includes a triplet index i, a source class $c_p^{src,i}$, and a target class $c_p^{tar,i}$. To determine the reliability score, we use the prediction output difference

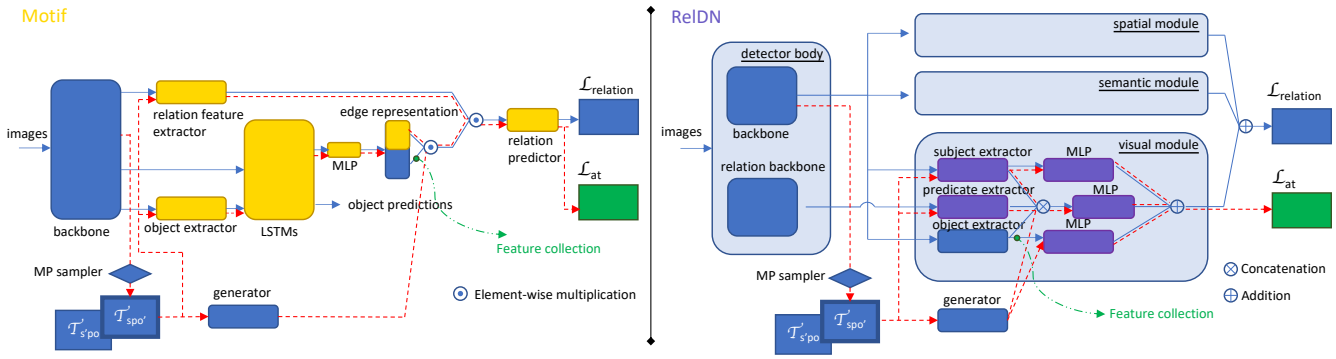

Fig. 4: The schematic view illustrates the combination of FSTA and SGG models. We visualize only the flow of $\mathcal{T}_{spo'}$ with red dotted lines for readability. The green dotted line indicates the point at which features are collected in the preparation stage.

Algorithm 1: FSTA

Input : Biased model predictions MP, pre-trained extracted object features and labels $\{\mathbf{f}_t, y_t\}_{t=1}^{T}$, batch size N_B, sampled pair size N_t, loss coefficient α

```
MPsampler ← build from MP based on Eq. (2) and (3);
{G, D} ← initialize adversarial modules;
for i = 1, ..., k_adv do
    {G, D} ← update as Eq. (4) using {f_t, y_t};
end
for i = 1, ..., k_SGG do
    {...}; // Do the general training step for the SGG model
    cand_triplets ← [];
    for j = 1, ..., N_B do
        pps ← sample N_t valid RoI pairs as shown in Fig. 3;
        cand_triplets ← cand_triplets + pps;
    end
    T_spo' ← EnumValidCombination (cand_triplets);
    T_s'po ← EnumValidCombination (cand_triplets);
    T_spo' ← Undersample (T_spo');
    T_s'po ← Undersample (T_s'po);
    gen_obj_labels ← MPsampler (T_spo');
    gen_obj_features ← G(z, gen_obj_labels);
    T_spo' ← replace object part with {gen_obj_labels,
      gen_obj_features};
    L_at ← CrossEntropy (M(F(ConCat (T_spo', T_s'po))),
      ground_truth_relations);
    update model from α L_at;
end
```

following

$$r_{int}(i) = v(l_{triplet}(i), c_p^{tar,i}) - v(l_{triplet}(i), c_p^{src,i}) \quad (5)$$

where $l_{triplet}(i) \in \mathbb{R}^{|C_{relations}|}$ returns the post-softmax model prediction of triplet i, and $v(l_{triplet}(\cdot), j)$ retrieves the value of $l_{triplet}(\cdot)$ at class j. We rank the score in ascending order and pick the top k_s% triplets for Soft Transfer while the other remains.

For those triplets with low reliability scores, we consider them as over-transferred. Thus, a positive probability should be assigned to the source class as the ground-truth label instead of zero. While achieving this and ensuring that the sum of the classes for the label is 1, we map the reliability scores to values within the range $[0, 1]$. Given a mapping function $Q(\cdot)$, the post-transferred result for triplet i can be represented as its label probability:

$$label_i(c) = \begin{cases} \frac{1}{1+Q(r_{int}(i))}, & \text{if } c = c_p^{tar,i} \\ \frac{Q(r_{int}(i))}{1+Q(r_{int}(i))}, & \text{if } c = c_p^{src,i} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

we set $Q(\cdot) = 1 - Q'(\cdot)$, where $Q'(\cdot)$ is a linear min-max scaling for the reliability scores.

Soft Transfer is applied on the original relation loss and needs no changes to it. Table 1 shows an example of post-transferred annotations. In IETrans, predicates of the selected triplets are reassigned to more informative ones (red). Our Soft Transfer evaluates the reliability of these reassigned predicates and converts them to non-binary values (blue).

Fig. 5: An example training image in VisualGenome.

Table 1: The differences in relation annotation among the raw dataset, the baseline IETrans (excluding External Transfer), and our proposed Soft Transfer for Figure 5.

config	relation annotations	
raw	man-sitting on-chair	laptop-on-table
	plant-in-pot	person-on-laptop
+IETrans	man-sitting on-chair	laptop-above-table
	plant-in-pot	person-looking at-laptop
++SoftTrans (ours)	man-sitting on-chair	
	laptop-**(above:0.76, on:0.24)**-table	
	plant-in-pot	
	person-**(looking at:1.0, on:0.0)**-laptop	

Table 2: The performance comparison for the predcls task on VG150. Scores for models listed in the first section are cited from their original papers, while models in subsequent sections use our implementation. "Model++X" is shorthand for "Model+IETrans+X". The best overall scores within each section are highlighted in bold. (Unit: %)

	Predicate Classification (Predcls)							
models	R@50	R@100	mR@50(h/b/t)	mR@100(h/b/t)	F1@50	F1@100	A@50	A@100
Motif+TDE [12]	46.2	51.4	25.5(-)	29.1(-)	32.9	37.2	35.9	40.3
Motif+DLFE [13]	52.5	54.2	26.9(-)	28.8(-)	35.6	37.6	39.7	41.5
Motif+NICE [9]	55.1	57.2	29.9(-)	32.3(-)	38.8	41.3	42.5	44.8
Motif+IETrans [10]	54.7	56.7	30.9(-)	33.6(-)	39.5	42.2	42.8	45.2
Motif+IETrans+rwt [10]	48.6	50.5	35.8(-)	39.1(-)	41.2	44.1	42.2	44.8
Motif+Inf [33]	51.5	55.1	24.7(-)	30.7(-)	33.4	39.4	38.1	42.9
Motif	65.0	67.2	16.1(39.3/9.3/1.2)	17.8(42.3/11.1/1.3)	25.8	28.1	40.6	42.5
Motif+IETrans†	54.8	57.1	29.6(42.2/33.4/14.0)	32.9(45.8/37.2/16.4)	38.4	41.7	42.2	45.0
Motif++FSTA (ours)	54.0	56.2	31.0(42.4/33.2/18.2)	34.8(45.8/37.4/22.0)	**39.4**	**43.0**	42.5	45.5
Motif++SoftTrans (ours)	58.6	60.8	28.0(42.5/31.8/10.6)	30.8(46.0/35.1/12.3)	37.9	40.9	43.3	45.8
Motif++Full (ours)	57.1	59.4	29.8(41.6/32.0/16.5)	33.2(45.1/35.8/19.5)	39.2	42.6	**43.5**	**46.3**
Motif+IETrans+rwt†	51.5	53.7	34.4(43.2/37.3/23.4)	38.8(46.3/40.5/30.2)	41.3	45.1	43.0	46.2
Motif++FSTA+rwt (ours)	49.0	51.1	35.9(42.0/36.8/29.1)	40.6(45.1/40.1/36.8)	41.4	45.2	42.4	45.8
Motif++SoftTrans+rwt (ours)	55.6	57.8	33.1(43.3/36.1/20.5)	38.3(46.4/39.9/28.9)	41.5	46.0	**44.4**	**48.0**
Motif++Full+rwt (ours)	53.4	55.5	34.7(42.4/35.7/26.6)	39.5(45.4/39.1/34.2)	**42.1**	**46.1**	44.1	47.5
RelDN	60.7	62.2	13.8(38.5/4.2/0.0)	14.9(40.9/5.3/0.1)	22.5	24.0	37.3	38.6
RelDN+IETrans†	38.6	39.9	29.6(35.4/33.1/20.5)	32.3(37.7/36.2/23.3)	33.5	35.7	34.1	36.1
RelDN++FSTA (ours)	37.0	38.2	31.3(33.7/33.0/27.3)	34.1(35.8/36.0/30.5)	33.9	36.0	34.2	36.2
RelDN++SoftTrans (ours)	52.4	53.8	26.1(37.6/27.6/13.9)	28.1(40.0/29.7/15.3)	34.8	36.9	**39.3**	**41.0**
RelDN++Full (ours)	49.2	50.7	28.2(35.8/28.2/21.0)	30.6(37.9/30.8/23.5)	**35.9**	**38.2**	38.7	40.7
RelDN+IETrans+rwt†	25.2	26.3	32.1(28.7/35.5/32.0)	34.6(30.7/37.8/35.1)	28.2	29.9	28.7	30.5
RelDN++FSTA+rwt (ours)	24.2	25.3	32.5(28.2/35.6/33.5)	35.8(30.1/38.1/38.8)	27.7	29.6	28.4	30.5
RelDN++SoftTrans+rwt (ours)	36.1	37.4	31.6(34.7/32.7/27.8)	34.8(36.8/34.9/32.7)	**33.7**	**36.1**	**33.9**	**36.1**
RelDN++Full+rwt (ours)	33.6	34.9	31.7(32.6/32.3/30.2)	35.1(34.6/35.3/35.4)	32.6	35.0	32.7	35.0

3.4 Implementation Details

We build our work upon an open source SGG model implementation† [34]. We integrate our system into two prevalent SGG model of distinct types: Motif [15] (which employs LSTM) and RelDN [16] (which utilizes CNN, multi-modality fusion, and contrastive losses). These were selected because they represent a variety of design elements commonly found in popular models. We use a ResNet50-FPN [35] Faster-RCNN [36] as the common detector backbone. The detector backbone is pre-trained on VisualGenome [6] and kept frozen. Fig. 4 illustrates how to combine our module with these SGG models. We implement IETrans with the default parameters: $k_i = 70$ and $k_e = 100$. In the FSTA module, N_t is set to 2 for Motif and 5 for RelDN to balance the number of artificial triplets in a mini-batch, considering the smaller batch size for RelDN. We set $s_{iou} = 0.7$††, $U_h = 0.2$ for Motif, and $s_{iou} = 0.5$, $U_h = 0.8$ for RelDN. Both models have a loss coefficient of $\alpha = 0.1$. We omit artificial triplets from $\mathcal{T}_{s'po}$ if their predicates are not in the tail group. For the soft transfer module, we set k_s to 10 for Motif and 70 for RelDN. In the experiments with a "reweighting" setting, only the original loss function is applied with reweighting, while $\mathcal{L}_{at}$, the loss for FSTA, does not apply as a standalone module. k_s also changes to 30 for Motif and 90 for RelDN.

†https://github.com/microsoft/scene_graph_benchmark

††We follow the implementations in † to compute s_{iou}.

4. Experiments

4.1 Dataset and Evaluation Protocol

We evaluated our system on the benchmark VG150 split of the VisualGenome dataset. This dataset consists of 60,784 training images and 26,446 testing images. It contains 150 object classes and 50 relation classes. Following the approach of [7], we sorted the predicates by cardinality, grouping the top 16, middle 17, and bottom 17 into **head**, **body**, and **tail** groups, respectively.

Our analysis focused on standard SGG tasks: **predcls**, **sgcls**, and **sgdet** [15], [16], [19]. These tasks evaluate the model with incrementally higher demands. For instance, "predcls" only assesses the model's ability in classifying relations using given object locations and categories. In contrast, "sgdet" evaluates both relation classification and object detection simultaneously. Our primary attention was on predcls since our proposed modules target improving predicate classification performance. We used the **Recall(R)@K** and **mean Recall(mR)@K** metrics for both full test set and per-class averaged recall evaluations. It is noteworthy that Recall@K is dominated by the performance of the top frequent classes due to the skewed predicate distribution, whereas mean Recall@K treats all classes equally. Given the

Table 3: The performance comparison for the sgcls and sgdet task on VG150. "Model++X" is shorthand for "Model+IETrans+X". Best overall scores in the section are highlighted in bold. Full results can be found in the appendix. (Unit: %)

	Scene Graph Classification (Sgcls)				Scene Graph Detection (Sgdet)			
models	R@100	mR@100(h/b/t)	F1@100	A@100	R@100	mR@100(h/b/t)	F1@100	A@100
Motif	38.9	10.7(25.2/7.0/0.8)	16.8	24.8	37.7	9.3(23.0/5.6/0.2)	14.9	23.5
Motif+IETrans†	30.1	20.9(25.9/21.2/15.9)	**24.7**	25.5	29.2	16.5(24.9/18.4/6.8)	21.1	22.9
Motif++FSTA (ours)	30.5	20.6(26.0/21.0/15.1)	24.6	25.6	28.8	17.1(25.1/18.3/8.2)	21.5	23.0
Motif++SoftTrans (ours)	34.1	18.7(26.4/20.8/9.3)	24.2	**26.4**	32.2	15.8(25.2/18.6/4.1)	21.2	24.0
Motif++Full (ours)	33.3	19.2(25.7/20.5/11.8)	24.4	26.3	32.2	17.0(24.6/18.5/8.3)	**22.3**	**24.6**
RelDN	36.9	7.9(22.9/1.6/0.0)	13.0	22.4	38.0	8.2(23.7/1.9/0.0)	13.5	23.1
RelDN+IETrans†	23.3	19.0(22.0/21.2/14.1)	20.9	21.2	22.0	18.4(22.0/20.9/12.4)	20.0	20.2
RelDN++FSTA (ours)	22.8	19.3(21.6/21.2/15.2)	20.9	21.1	21.1	19.1(21.3/21.0/15.1)	20.1	20.1
RelDN++SoftTrans (ours)	32.8	15.5(23.4/16.9/6.7)	21.1	**24.2**	32.3	15.4(23.8/16.5/6.3)	20.9	**23.9**
RelDN++Full (ours)	31.1	16.8(22.9/17.4/10.6)	**21.8**	24.0	29.3	17.2(22.5/17.8/11.5)	**21.7**	23.3

observed trade-off between Recall@K and mean Recall@K from earlier studies, we also reported the **F1@K** (their harmonic mean) and **Avg(A)@K** (their arithmetic mean) [9], [10] as the "overall" metrics in our comprehensive evaluation. **All metrics are the higher the better**.

4.2 Comparing to Other Methods

We compared our results with IETrans and several other recent model-agnostic SGG methods (first section of Table 2). IETrans serves as the baseline and is currently one of the best model-agnostic methods available.

Original baseline and our re-implementation. We compared our method with the reproduced baselines (denoted as †). For Motif+IETrans in the predcls task, the reproduced version yielded similar scores to those of the original, with a slightly higher R@100 and lower mR@100. These differences may due to some implementation variations in the base SGG model. Therefore, we use the reproduced version as our standard because it maintains identical implementation settings and IETrans transfer lists, consistent with our proposed methods. The original IETrans paper did not present results for RelDN; therefore, we also compared our results with a reproduced version. In summary, all our implementations share the basic settings to ensure a fair comparison.

Improved relation prediction over the baseline. Table 2 summarizes the scores for predcls. The results reveal that our method substantially outperformed the baseline for both the Motif and RelDN models. Specifically, the F1@100 score rose from 41.7 to 43.0 (a 3.1% relative gain) for Motif, and from 35.7 to 38.2 (a 7.0% relative gain) for the RelDN model. The A@100 score also increases. With FSTA, the standout feature was the mR enhancement in tail classes (e.g., from 16.4 to 22.0 for Motif). The artificial triplets generated in FSTA enriched the variation of triplets available for the relation predictor, aiding especially the sparse classes. As for frequent classes, the score decline is minor. On the other hand, Soft Transfer was intended to reduce the degree of label reassignment for less reliable transfers. This led to a score trend the opposite of the original IETrans: while recall scores raised, the tail mean recall scores decreased (e.g., R@100 increases from 57.1 to 60.8 for Motif, and 39.9 to 53.8 for RelDN). In certain cases, Soft Transfer can slightly reduce the F1 score, because the harmonic mean prioritizes enhancements in the smaller one. Nonetheless, the Avg@100 witnessed a notable boost with Soft Transfer. Combining both modules, the full system leveraged their complementary benefits, consistently delivering among the top F1/Avg@100 results for both models, indicating an effective balance of trade-offs.

Compatible with the reweighting setting. We also adhere to the original settings described in the IETrans paper to compare the models when integrated with the "reweighting" technique (+rwt) [10]. Our method proved efficacious even under this setting. Both the FSTA and Soft Transfer modules served their intended purposes, driving improvements across rare and frequent classes alike. The "Motif++Full+rwt" method achieves an increase in F1@100 from 45.1 to 46.1, and in Avg@100 from 46.2 to 47.5, thereby demonstrating the mitigation of the performance trade-off. For RelDN with reweighting, the performance was not as beneficial as for the Motif models. Although the mR@100 for the tail group further increased, the impact on R@K cannot be overlooked, leading to a dip in the overall scores. One possible explanation for this could be the architecture of the RelDN model, which already incorporates a frequency prior branch. Consequently, we did not add the frequency prior values during inference for the RelDN models, while we did so for the Motif models following [10]. This leads to a more serious degradation in head classes, despite having the best performance on tail classes. However, our method still consistently surpassed the baseline in the RelDN +rwt setting.

Similar trends observed for sgcls and sgdet. Table 3 showcases the digested results for sgcls and sgdet. Here, we noticed trends analogous to those in predcls. For RelDN, the full version achieves the best F1@100 and the second-best A@100. For the Motif sgdet, the full version outperforms all the others on both overall metrics (i.e., a 5.7% and 7.4% relative gain for F1@100 and A@100 over the baseline method, respectively). However, the FSTA module yields some unexpected results in the sgcls task. One potential cause is that we applied identical FSTA settings across all

tasks. However, sgcls uniquely relies on ground-truth boxes only for input proposals, which is different from the other tasks. This difference might result in distinct regularization effects, as the artificial triplets are constructed from sampled proposal pairs.

5. Discussions

In this section, we focus on the predcls task results for the Motif model[†] to gain a deeper understanding of our methods. Results for RelDN can be found in appendix.

5.1 Ablation Study

Table 4 presents the components ablated from FSTA to demonstrate their contributions to the module. The results indicate that all components positively influence the improvement of F1@100. Among these, undersampling has the greatest impact on F1@100, adjusting the proportions of artificial triplets in the head, body, and tail predicate groups from 0.70, 0.14, and 0.15 to 0.33, 0.32, and 0.35 respectively. Additionally, incorporating $\mathcal{T}_{s'po}$ effectively introduces new training combinations. The MP-sampler also plays a crucial role, further boosting R@100.

Table 5 summarizes the ablation results with reweighting. A similar trend is observed that the components in FSTA contribute to the increase in scores for tail relation groups and the mR@100.

Table 4: The ablation study results for our FSTA module. For the components, "us" refers to undersampling, "+sbj" denotes adding artificial set $\mathcal{T}_{s'po}$, and MP indicates that the MP-sampler is applied. (Unit: %)

us	+sbj	MP	R@100	mR@100(h/b/t)	F1/Avg@100
			55.8	33.4(45.2/38.7/17.1)	41.8/44.6
✓			55.3	34.4(45.3/39.9/19.6)	42.4/44.9
✓	✓		54.8	35.0(44.8/39.1/21.6)	42.7/44.9
✓	✓	✓	56.2	34.8(45.8/37.4/22.0)	**43.0/45.5**

Table 5: The ablation study results for our FSTA module with Motif and "reweighting". Item descriptions are identical to Table 4. (Unit: %)

us	+sbj	MP	R@100	mR@100(h/b/t)	F1/Avg@100
			54.7	36.6(45.8/39.2/25.4)	43.9/45.7
✓			53.1	38.7(45.1/40.8/30.7)	44.8/45.9
✓	✓		52.5	39.7(44.5/41.4/33.4)	45.2/**46.1**
✓	✓	✓	51.1	40.6(45.1/40.1/36.8)	**45.2**/45.8

5.2 Sensitivity Analysis

We then investigate the choice of percentage k_s in Soft Transfer. A "Naïve" setting would simply apply soft transfer to all reassigned triplets without our ranking and mapping mechanisms, where both source and target labels are assigned a value of 0.5. The results are summarized in Table 6.

[†]Unless specified otherwise, this means "Model+IETrans+X".

As the number of entirely transferred triplets is reduced, R@100 recovers as k_s grows, yet mR@100 decreases. The "Naïve" setting consistently performs worse than the others in the F1@100 or even Avg@100 metrics and only be on par with the baseline IETrans ($Avg@100 = 45.0$). This highlights the significance of our devised method.

Table 6: The sensitivity results for our Soft Transfer module. "$*$" indicates the setting applied in our method. (Unit: %)

settings:	R@100	mR@100(h/b/t)	F1/Avg@100
$k_s = 0.1*$	60.8	30.8(46.0/35.1/12.3)	**40.9**/45.8
$k_s = 0.3$	63.1	30.1(45.4/34.5/11.3)	40.7/**46.6**
$k_s = 0.5$	64.5	28.4(44.7/31.6/9.8)	39.4/46.4
Naïve	66.5	23.4(43.6/22.6/5.0)	34.6/45.0

5.3 Comparison with Real Data Resampling

We compare the effects of resampling real data with our FSTA. Although both approaches augment the number of rare predicates, their motivations and methods differ. FSTA aims to steer the predicate classification layers towards understandind the inherent concept of the predicate by leveraging combinative, yet semantically plausible, artificial triplets. This also introduces new variations to the training data. In contrast, resampling merely duplicates samples from rare classes to mitigate dataset imbalance; however, it is susceptible to overfitting.

For our real data resampling implementation, we altered the training set by duplicating the image n times if it contained more than one triplet of tail group predicates. We analyzed the performance difference when applied independently and the combined for Motif+IETrans on the predcls task. The scores are listed in Table 7.

Our "+FSTA" is more effective than "+resampling", as it yields superior overall metrics for both F1 and Avg. Combining both can boost the mean recall of the tail group. Intensive resampling improved tail classes but reduced frequent class recall. We did not observe positive effects when n was larger than 4.

Table 7: The results of comparing our FSTA with resampling. (Unit: %)

settings:	R@100	mR@100(h/b/t)	F1/Avg@100
IETrans	57.1	32.9(45.8/37.2/16.4)	41.7/45.0
+resample($n = 1$)	57.2	33.5(45.5/37.8/18.1)	42.3/45.4
+resample($n = 2$)	55.6	33.5(45.1/37.4/18.8)	41.8/44.5
+resample($n = 3$)	55.5	33.5(44.6/36.8/19.9)	41.8/44.5
+FSTA (ours)	56.2	34.8(45.8/37.4/22.0)	43.0/**45.5**
+resample+FSTA	54.9	35.8(44.7/37.2/26.1)	**43.3**/45.3

5.4 Parameter Choices for FSTA

To explore the quality of our s_{iou} and U_h choices, which are applied across settings, we assess the performance within the reweighting setting. Table 8 details the results.

Our observations are as follows: (1) Lower values of U_h

tend to result in higher tail group performance, attribute to the increased ratio of tail relations in the artificial triplets. (2) A higher s_{iou} threshold allows only the most precise feature representations, benefiting the data-sparse tail group while potentially harming the generalizability in frequent classes. Overall, we found that the parameters we selected perform reasonably well, even under such a different setting.

Table 8: The results of parameter choices for FSTA with Motif and "reweighting". "$*$" indicates the setting applied in our method. (Unit: %)

Param.	Value	R@100	mR@100(h/b/t)	F1/Avg@100
s_{iou}	0.5	51.6	39.9(45.8/40.6/33.9)	45.0/45.8
	0.6	50.9	39.2(45.8/41.0/31.2)	44.3/45.1
	0.7$*$	51.1	40.6(45.1/40.1/36.8)	**45.2/45.9**
	0.8	50.3	40.7(44.7/40.0/37.6)	45.0/45.5
	0.9	49.5	40.9(44.4/39.4/39.1)	44.8/45.2
U_h	0.2$*$	51.1	40.6(45.1/40.1/36.8)	45.2/45.9
	0.4	51.8	40.3(45.4/40.5/35.4)	**45.3/46.1**
	0.5	51.0	40.2(45.3/40.5/35.1)	45.0/45.6
	0.6	51.2	39.7(45.8/39.5/34.3)	44.7/45.5
	0.8	51.7	39.9(44.5/43.1/32.3)	45.0/45.8

5.5 A Study on MP-sampler

We examine the role of the MP-sampler. In this case, the count of artificial triplets per predicate class is invariant, but the distribution of combinations changes. We undertake a case study focusing on the mR@100 tail group, where FSTA has shown significant performance gains. Fig. 6 (left) portrays the per-class recall: 12 of 17 classes either tie or improve with the MP-sampler. Next, we study the rationale behind the signal designed in the MP-sampler. It uses the scores from $d(\cdot)$ as the sampling probability, which inversely correlates with recall. For example, $d(\cdot) = 0$ implies that the top-1 predicate prediction aligns with the ground-truth, whereas $d(\cdot) > 0$ does not. We hypothesize that sampling more object labels of higher $d(\cdot)$ scores can make correct predictions easier for the unbised model. Thus, we analyze the changes in accumulated $d(\cdot)$ scores between the pretrained and unbiased models, from those object classes where counts have increased. Fig. 6 (right) visualizes the results computed on test data. Out of the 17 classes, 14 show non-negative total reductions scores, confirming that our designed signal is evident in the test data. We also inspect the relationship between the reduced score and class recall. The majority of classes follow a similar trend, with only four exhibiting the converse pattern (e.g., recall increases while the reduced score is negative).

5.6 Feature Visualization

We visualize the similarity between the synthesized object features in artificial triplets and the real ones in the given classes. Fig. 7 illustrates the sampled classes within the body, tail, and head groups, separated by different colors. The neighborhood identity between the real and synthetic features from the same class suggest the effectiveness of the generator.

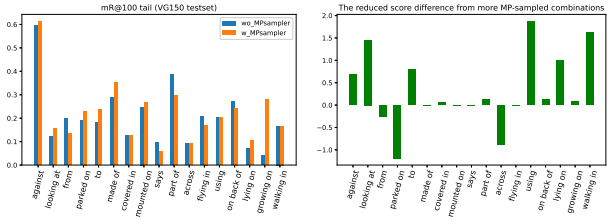

Fig. 6: A case study examining the effects of the MP-sampler. (Left) The tail group mR@100 comparison between setups without and with MP-sampler. (Right) The accumulated $d(\cdot)$ reduction contributed by objects that are sampled more frequently.

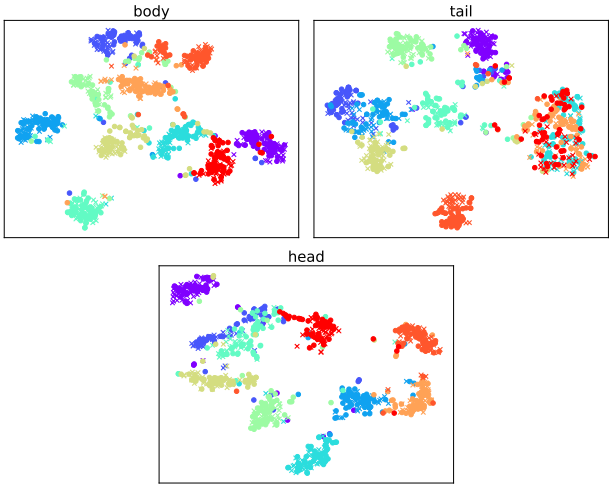

Fig. 7: The t-SNE plots for object features in the artificial triplets. We select 10 classes from each group. Real features are plotted in dots and generated in crosses.

6. Conclusion

In this paper, we introduce two key concepts to enhance the dataset modification approach for unbiased SGG: a novel data augmentation strategy via our FSTA module, and improved predicate reassignment efficiency through Soft Transfer. The FSTA module substantially boosts tail class recall by generating additional artificial triplets, while Soft Transfer offers a more nuanced evaluation of the reliability of individual transfers, allowing for a continuous degree of transfer and mitigating the typical decline in frequent class recall during reassignment. Experimental results confirm that integrating the complementary modules improves overall performance, surpassing the baseline IETrans.

Acknowledgments

This work was supported by the commissioned research (No. 225) by National Institute of Information and Communications Technology (NICT), JSPS/MEXT KAKENHI Grant Numbers JP23H03449 and JP22H05015.

References

[1] J. Johnson, R. Krishna, M. Stark, L.J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.3668–3678, 2015.

[2] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.45, no.1, pp.1–26, 2021.

[3] Z. Luo, W. Xie, S. Kapoor, Y. Liang, M. Cooper, J.C. Niebles, E. Adeli, and F.F. Li, "Moma: Multi-object multi-actor activity parsing," Advances in Neural Information Processing Systems, ed. M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J.W. Vaughan, pp.17939–17955, Curran Associates, Inc., 2021.

[4] S. Aditya, Y. Yang, C. Baral, Y. Aloimonos, and C. Fermüller, "Image understanding using vision and reasoning through scene description graph," Computer Vision and Image Understanding, vol.173, pp.33–45, 2018.

[5] J. Gu, S. Joty, J. Cai, H. Zhao, X. Yang, and G. Wang, "Unpaired image captioning via scene graph alignments," Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.10323–10332, 2019.

[6] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.J. Li, D.A. Shamma, M.S. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," Int. J. Comput. Vision, vol.123, no.1, p.32–73, may 2017.

[7] A. Desai, T.Y. Wu, S. Tripathi, and N. Vasconcelos, "Learning of visual relations: The devil is in the tails," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp.15404–15413, October 2021.

[8] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," International Conference on Learning Representations, 2020.

[9] L. Li, L. Chen, Y. Huang, Z. Zhang, S. Zhang, and J. Xiao, "The devil is in the labels: Noisy label correction for robust scene graph generation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.18869–18878, June 2022.

[10] A. Zhang, Y. Yao, Q. Chen, W. Ji, Z. Liu, M. Sun, and T.S. Chua, "Fine-grained scene graph generation with data transfer," European conference on computer vision, pp.409–424, Springer, 2022.

[11] X. Lyu, L. Gao, Y. Guo, Z. Zhao, H. Huang, H.T. Shen, and J. Song, "Fine-grained predicates learning for scene graph generation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.19467–19475, June 2022.

[12] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.3716–3725, 2020.

[13] M.J. Chiou, H. Ding, H. Yan, C. Wang, R. Zimmermann, and J. Feng, "Recovering the unbiased scene graphs from the biased ones," Proceedings of the 29th ACM International Conference on Multimedia, MM '21, New York, NY, USA, p.1581–1590, Association for Computing Machinery, 2021.

[14] S. Yan, C. Shen, Z. Jin, J. Huang, R. Jiang, Y. Chen, and X.S. Hua, "Pcpl: Predicate-correlation perception learning for unbiased scene graph generation," Proceedings of the 28th ACM International Conference on Multimedia, MM '20, New York, NY, USA, p.265–273, Association for Computing Machinery, 2020.

[15] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.5831–5840, 2018.

[16] J. Zhang, K.J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical contrastive losses for scene graph parsing," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.11535–11543, 2019.

[17] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp.852–869, Springer, 2016.

[18] D. Xu, Y. Zhu, C.B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.5410–5419, 2017.

[19] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu, "Learning to compose dynamic tree structures for visual contexts," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.6619–6628, 2019.

[20] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-embedded routing network for scene graph generation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.6163–6171, 2019.

[21] R. Li, S. Zhang, B. Wan, and X. He, "Bipartite graph network with adaptive message passing for unbiased scene graph generation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.11109–11119, June 2021.

[22] J. Yu, Y. Chai, Y. Wang, Y. Hu, and Q. Wu, "Cogtree: Cognition tree loss for unbiased scene graph generation," Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, ed. Z.H. Zhou, pp.1274–1280, International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[23] I. Biederman, "Recognition-by-components: a theory of human image understanding.," Psychological review, vol.94, no.2, p.115, 1987.

[24] P. Tokmakov, Y.X. Wang, and M. Hebert, "Learning compositional representations for few-shot recognition," Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.6372–6381, 2019.

[25] K. Kato, Y. Li, and A. Gupta, "Compositional learning for human object interaction," Proceedings of the European Conference on Computer Vision (ECCV), pp.234–251, 2018.

[26] Z. Hou, X. Peng, Y. Qiao, and D. Tao, "Visual compositional learning for human-object interaction detection," Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16, pp.584–600, Springer, 2020.

[27] Z. Hou, B. Yu, Y. Qiao, X. Peng, and D. Tao, "Detecting human-object interaction via fabricated compositional learning," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.14646–14655, June 2021.

[28] Y. Zhong, J. Shi, J. Yang, C. Xu, and Y. Li, "Learning to generate scene graph from natural language supervision," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp.1823–1834, October 2021.

[29] T. He, L. Gao, J. Song, and Y.F. Li, "Towards open-vocabulary scene graph generation with prompt-based finetuning," European Conference on Computer Vision, pp.56–73, Springer, 2022.

[30] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.

[31] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.5542–5551, 2018.

[32] R. Felix, I. Reid, G. Carneiro, *et al.*, "Multi-modal cycle-consistent generalized zero-shot learning," Proceedings of the European conference on computer vision (ECCV), pp.21–37, 2018.

[33] B.A. Biswas and Q. Ji, "Probabilistic debiasing of scene graphs," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.10429–10438, June 2023.

[34] X. Han, J. Yang, H. Hu, L. Zhang, J. Gao, and P. Zhang, "Image scene graph generation (sgg) benchmark," arXiv preprint arXiv:2107.12604, 2021.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in neural information processing systems, vol.28, 2015.
[37] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," arXiv preprint arXiv:1701.04862, 2017.
[38] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," CoRR, vol.abs/2103.00020, 2021.

Appendix A: Full Sgcls and Sgdet Tasks Results

The full results for the "sgcls" and "sgdet" tasks are listed in Table A5 (sgcls) and A6 (sgdet).

Appendix B: Qualitative Results

We visualize the results of predicate prediction in Fig. A1.

Appendix C: Results for Ablation Study (RelDN)

Table A1 and Table A2 summarize the ablation results for FSTA module under RelDN and RelDN with reweighting, respectively.

Table A1: The ablation study results for our FSTA module with RelDN. Item descriptions are identical to Table 4. (Unit: %)

us	+sbj	MP	R@100	mR@100(h/b/t)	F1/Avg@100
			38.9	33.4(37.0/38.8/24.5)	35.9/36.2
✓			39.1	33.9(37.0/38.2/26.7)	**36.3/36.5**
✓	✓		39.0	34.0(37.0/38.9/26.1)	36.3/36.5
✓	✓	✓	38.2	34.1(35.8/36.0/30.5)	36.0/36.2

Table A2: The ablation study results for our FSTA module with RelDN and "reweighting". Item descriptions are identical to Table 4. (Unit: %)

us	+sbj	MP	R@100	mR@100(h/b/t)	F1/Avg@100
			25.3	34.9(29.9/39.6/35.0)	29.3/30.1
✓			25.4	35.1(30.3/39.4/35.4)	29.5/30.3
✓	✓		25.5	35.1(30.3/39.8/35.0)	29.5/30.3
✓	✓	✓	25.3	35.8(30.1/38.1/38.8)	**29.6/30.5**

Appendix D: Results for Sensitivity Analysis (RelDN)

Table A3 lists the Soft Transfer sensitivity results of RelDN for the predcls task. $k_s = 0.7$ actually achieves a better score on **both** R@100 and mR@100 over the baseline method (an improvement of 39.9 to 40.8 for R@100 and 32.3 to 32.5 for mR@100). Nevertheless, modifying $Q(\cdot) = 1 - Q'(\cdot)$ to $Q(\cdot) = Q'(\cdot)$ leads to stronger overall performance, due to the larger space for the R@100 score recovery. Again, the "Naïve" case is inferior to the applied settings.

Table A3: The sensitivity results with RelDN for our Soft Transfer module. "$*$" indicates the setting applied in our method. "$\diamond$" stands for a modified $Q(\cdot)$. (Unit: %)

settings:	R@100	mR@100(h/b/t)	F1/Avg@100
$k_s = 0.5\diamond$	50.8	29.9(39.5/32.4/18.3)	**37.6**/40.4
$k_s = 0.7 \diamond *$	53.8	28.1(40.0/29.7/15.3)	36.9/**40.9**
$k_s = 0.7$	40.8	32.5(38.0/36.2/23.8)	36.2/36.7
Naïve	56.3	25.0(40.9/25.5/9.6)	34.7/40.7

Appendix E: Results for FSTA Parameter Choices (RelDN)

Table A4 describes the results of FSTA parameter study for RelDN.

Table A4: The results of parameter choices for FSTA with RelDN and "reweighting". "$*$" indicates the setting applied in our method. (Unit: %)

Param.	Value	R@100	mR@100(h/b/t)	F1/Avg@100
s_{iou}	0.5$*$	25.3	35.8(30.1/38.1/38.8)	**29.6/30.5**
	0.6	24.8	35.0(29.5/38.2/36.9)	29.0/29.9
	0.7	25.2	34.4(29.4/38.5/35.0)	29.1/29.8
	0.8	24.3	34.0(29.2/36.8/35.7)	28.3/29.2
	0.9	24.5	36.3(29.3/37.8/41.4)	29.3/30.4
U_h	0.2	24.6	35.7(29.5/38.1/39.0)	29.1/30.2
	0.4	24.3	36.8(29.1/37.5/43.3)	29.3/30.5
	0.5	25.4	36.1(30.0/37.7/40.1)	**29.8/30.8**
	0.6	24.5	35.8(29.3/38.1/39.5)	29.1/30.2
	0.8$*$	25.3	35.8(30.1/38.1/38.8)	29.6/30.5

Appendix F: Randomness of FSTA

The resource of randomness: Including the undersampling step and the generator pretraining. We selected a fixed checkpoint for the generator based on the classification accuracies observed in the validation data.
The reproducibility of randomness: In the SGG model training, we followed the open-source SGG model implementation† to set the seed for the libraries and switch to deterministic mode for the cudnn library.
The impact of randomness: We measured the standard deviation of R@100 and mR@100 under "Motif++FSTA+rwt" in the predcls task, using five different runs. The values are 0.23 for R@100 and 0.31 for mR@100. Note that these include randomness from both the Motif model and the FSTA module.

Appendix G: Object Generator

We exploit a conditional-GAN based model to synthesize `object'` features, due to its lightweight and low additional computational cost. In the pre-processing step, we collect the real features from model predictions on training data (See

Table A5: The full performance comparison for the sgcls task on VG150. Scores for models listed in the first section are cited from their original papers, while models in subsequent sections use our implementation. "Model++X" is shorthand for "Model+IETrans+X". The best overall scores within each section are highlighted in bold. (Unit: %)

	Scene Graph Classification (Sgcls)							
models	R@50	R@100	mR@50(h/b/t)	mR@100(h/b/t)	F1@50	F1@100	A@50	A@100
Motif+TDE [12]	27.7	29.9	13.1(-)	14.9(-)	17.8	19.9	20.4	22.4
Motif+DLFE [13]	32.3	33.1	15.2(-)	15.9(-)	20.7	21.5	23.8	24.5
Motif+NICE [9]	33.1	34.0	16.6(-)	17.9(-)	22.1	23.5	24.9	26.0
Motif+IETrans [10]	32.5	33.4	16.8(-)	17.9(-)	22.2	23.3	24.7	25.7
Motif+IETrans+rwt [10]	29.4	30.2	21.5(-)	22.8(-)	24.8	26.0	25.5	26.5
Motif+Inf [33]	32.2	33.8	14.5(-)	17.4(-)	20.0	23.0	23.4	25.6
Motif	38.1	38.9	10.0(23.9/6.3/0.6)	10.7(25.2/7.0/0.8)	15.8	16.8	24.1	24.8
Motif+IETrans†	29.1	30.1	18.0(24.5/19.7/10.3)	20.9(25.9/21.2/15.9)	22.2	**24.7**	23.6	25.5
Motif++FSTA (ours)	29.5	30.5	18.3(24.5/19.4/11.5)	20.6(26.0/21.0/15.1)	22.6	24.6	23.9	25.6
Motif++SoftTrans (ours)	33.0	34.1	17.2(24.9/19.1/8.1)	18.7(26.4/20.8/9.3)	22.6	24.2	**25.1**	**26.4**
Motif++Full (ours)	32.2	33.3	17.7(24.2/18.7/10.5)	19.2(25.7/20.5/11.8)	**22.8**	24.4	25.0	26.3
Motif+IETrans+rwt†	28.1	28.6	18.9(24.1/20.2/12.6)	21.0(25.0/20.9/17.4)	22.6	24.2	23.5	24.8
Motif++FSTA+rwt (ours)	26.1	26.6	19.6(23.3/20.2/15.4)	21.6(24.2/20.9/20.0)	22.4	23.8	22.9	24.1
Motif++SoftTrans+rwt (ours)	30.9	31.5	18.0(24.2/19.2/10.9)	20.4(25.0/20.0/16.3)	22.7	**24.8**	**24.5**	**26.0**
Motif++Full+rwt (ours)	29.3	29.9	18.5(23.6/19.3/13.1)	21.0(24.4/20.0/18.7)	**22.7**	24.7	23.9	25.5
RelDN	36.0	36.9	7.4(21.8/1.3/0.0)	7.9(22.9/1.6/0.0)	12.3	13.0	21.7	22.4
RelDN+IETrans†	22.4	23.3	17.8(20.8/20.0/12.7)	19.0(22.0/21.2/14.1)	19.8	20.9	20.1	21.2
RelDN++FSTA (ours)	21.9	22.8	17.9(20.4/19.9/13.6)	19.3(21.6/21.2/15.2)	19.7	20.9	19.9	21.1
RelDN++SoftTrans (ours)	31.8	32.8	14.6(22.2/15.9/6.2)	15.5(23.4/16.9/6.7)	20.0	21.1	**23.2**	**24.2**
RelDN++Full (ours)	30.1	31.1	15.6(21.7/16.3/9.4)	16.8(22.9/17.4/10.6)	**20.5**	**21.8**	22.9	24.0
RelDN+IETrans+rwt†	17.7	18.5	19.3(18.7/21.0/18.1)	20.6(19.9/22.1/19.7)	18.5	19.5	18.5	19.6
RelDN++FSTA+rwt (ours)	16.6	17.4	19.0(18.0/21.4/17.4)	20.8(19.1/22.5/20.5)	17.7	18.9	17.8	19.1
RelDN++SoftTrans+rwt (ours)	23.7	24.6	18.4(21.6/18.6/15.3)	21.1(22.8/19.5/21.0)	**20.7**	**22.7**	**21.0**	**22.9**
RelDN++Full+rwt (ours)	22.6	23.5	18.7(20.6/19.0/16.5)	21.6(21.8/20.0/23.1)	20.5	22.5	20.7	22.6

Fig.4 in the manuscript). The adversarial loss function for the GAN model consist of three parts: $\mathcal{L}_{wgangp}$, $\mathcal{L}_{cls}$, and $\mathcal{L}_{recon}$.

$\mathcal{L}_{wgangp}$ is a standard WGAN loss with gradient penalty [37] as Eq.(A1).

$$\begin{aligned}\mathcal{L}_{wgangp} =& \mathbb{E}_{\mathbf{x}\sim real}[D(\mathbf{x},\mathbf{s}_c)] - \mathbb{E}_{\tilde{\mathbf{x}}\sim gen}[D(\tilde{\mathbf{x}},\mathbf{s}_c)] \\ &- \lambda\mathbb{E}[(||\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}},\mathbf{s}_c)||_2 - 1)^2]\end{aligned} \tag{A1}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the feature sampled from real data, $\tilde{\mathbf{x}} = G(\mathbf{z},\mathbf{s}_c) \in \mathbb{R}^d$ is the synthesized feature from generator G. d is the size of object feature. $\hat{\mathbf{x}} = \alpha\mathbf{x}+(1-\alpha)\tilde{\mathbf{x}}$ is an interpolated feature with α sampled from a uniform distribution. $\mathbf{z}$ is an initial vector sampled from normal distribution, and $\mathbf{s}_c$ is a condition vector represents the object class. We collect $\mathbf{s}_c$ from the pre-trained CLIP [38] text encoder. We use the basic template "a photo of a [OBJECT NAME]." as the input prompt to text encoder, then the output vector as the class representation.

$\mathcal{L}_{cls}$ is a regularization loss for the generator G. It utilizes a softmax classifier pre-trained on real data to encourage the generator to output features with enhanced discriminability. That is, the synthetic features can be better classified. Eq.(A2) describes its loss function.

$$\mathcal{L}_{cls} = -\mathbb{E}_{\tilde{\mathbf{x}}\sim gen}[\log P(y|\tilde{x};\theta_{cls})] \tag{A2}$$

where θ_{cls} is the weights of the softmax classifier. y is the corresponding class label. During the adversarial training, the pre-trained classifier is frozen.

$\mathcal{L}_{recon}$ is another regularization term for the class consistency between generator output and its condition input. A reconstructor $R(\cdot)$ is pre-trained on real data to infer the class condition vector from the feature. Eq.(A3) describes its loss function.

$$\mathcal{L}_{recon} = \mathbb{E}_{\tilde{\mathbf{x}}\sim gen}[\|R(\tilde{\mathbf{x}}) - \mathbf{s}_c\|_2] \tag{A3}$$

the reconstructor is also frozen in the adversarial training.

The overall loss function is as below and identical to Eq.(4) in the main paper.

$$\min_G \max_D \mathcal{L}_{wgangp} + \beta\mathcal{L}_{cls} + \gamma\mathcal{L}_{recon} \tag{A4}$$

We list the model architecture for training the object generator in Table A7.

Appendix H: Hyperparameter Details

We list the parameter choices for training SGG models and the generator model in Table A8.

Table A6: The full performance comparison for the sgdet task on VG150. Scores for models listed in the first section are cited from their original papers, while models in subsequent sections use our implementation. "Model++X" is shorthand for "Model+IETrans+X". The best overall scores within each section are highlighted in bold. (Unit: %)

	Scene Graph Detection (Sgdet)							
models	R@50	R@100	mR@50(h/b/t)	mR@100(h/b/t)	F1@50	F1@100	A@50	A@100
Motif+TDE [12]	16.9	20.3	8.2(-)	9.8(-)	11.0	13.2	12.5	15.1
Motif+DLFE [13]	25.4	29.4	11.7(-)	13.8(-)	16.0	18.8	18.6	21.6
Motif+NICE [9]	27.8	31.8	12.2(-)	14.4(-)	17.0	19.8	20.0	23.1
Motif+IETrans [10]	26.4	30.6	12.4(-)	14.9(-)	16.9	20.0	19.4	22.8
Motif+IETrans+rwt [10]	23.5	27.2	15.5(-)	18.0(-)	18.7	21.7	19.5	22.6
Motif+Inf [33]	23.9	27.1	9.4(-)	11.7(-)	13.5	16.3	16.7	19.4
Motif	32.7	37.7	7.7(19.4/4.4/0.1)	9.3(23.0/5.6/0.2)	12.5	14.9	20.2	23.5
Motif+IETrans†	24.7	29.2	13.8(20.9/15.5/5.3)	16.5(24.9/18.4/6.8)	17.7	21.1	19.3	22.9
Motif++FSTA (ours)	24.3	28.8	13.9(21.2/15.7/5.3)	17.1(25.1/18.3/8.2)	17.7	21.5	19.1	23.0
Motif++SoftTrans (ours)	27.4	32.2	13.1(21.1/15.5/3.2)	15.8(25.2/18.6/4.1)	17.7	21.2	20.3	24.0
Motif++Full (ours)	27.5	32.2	14.0(20.8/15.7/5.8)	17.0(24.6/18.5/8.3)	**18.6**	**22.3**	**20.8**	**24.6**
Motif+IETrans+rwt†	23.8	28.5	15.6(22.5/18.3/6.3)	18.8(26.5/21.1/9.4)	18.8	22.7	19.7	23.7
Motif++FSTA+rwt (ours)	22.1	26.4	17.3(21.3/18.7/12.0)	20.1(25.2/21.1/14.2)	19.4	22.8	19.7	23.3
Motif++SoftTrans+rwt (ours)	27.0	31.8	15.0(22.1/17.6/5.8)	19.4(26.1/20.4/12.0)	19.3	**24.1**	**21.0**	**25.6**
Motif++Full+rwt (ours)	25.5	30.1	16.3(21.4/18.2/9.5)	19.5(25.2/20.8/12.9)	**19.9**	23.7	20.9	24.8
RelDN	32.7	38.0	6.7(19.7/1.1/0.0)	8.2(23.7/1.9/0.0)	11.1	13.5	19.7	23.1
RelDN+IETrans†	18.4	22.0	14.9(18.3/16.8/9.7)	18.4(22.0/20.9/12.4)	16.5	20.0	16.7	20.2
RelDN++FSTA (ours)	17.5	21.1	15.5(17.8/17.0/11.7)	19.1(21.3/21.0/15.1)	16.4	20.1	16.5	20.1
RelDN++SoftTrans (ours)	27.3	32.3	12.5(19.9/13.3/4.7)	15.4(23.8/16.5/6.3)	17.1	20.9	**19.9**	**23.9**
RelDN++Full (ours)	24.6	29.3	14.4(19.0/14.7/9.8)	17.2(22.5/17.8/11.5)	**18.2**	**21.7**	19.5	23.3
RelDN+IETrans+rwt†	12.2	14.7	16.5(14.9/19.8/14.8)	19.7(17.8/23.2/17.9)	14.0	16.8	14.4	17.2
RelDN++FSTA+rwt (ours)	11.2	13.8	16.6(14.8/19.0/15.8)	19.5(17.7/22.5/18.2)	13.4	16.2	13.9	16.7
RelDN++SoftTrans+rwt (ours)	18.0	21.6	15.9(18.6/18.3/11.1)	18.9(22.1/21.5/13.5)	**16.9**	**20.2**	**17.0**	**20.3**
RelDN++Full+rwt (ours)	16.0	19.5	16.5(17.3/18.0/14.2)	19.9(20.5/21.4/17.7)	16.2	19.7	16.3	19.7

Table A7: The model architecture.

module	input and the forward flow
G	**Input: ($\mathbf{z}$, $\mathbf{s_c}$)** out = concat(Input) out = linear1(in=1024+512, out=4096)(out) out = LeakyReLU(slope=-0.2)(out) $\tilde{\mathbf{x}}$ = linear2(in=4096, out=1024)(out)
D	**Input: ($\tilde{\mathbf{x}}$, $\mathbf{s_c}$) or ($\mathbf{x}$, $\mathbf{s_c}$)** out = concat(Input) out = linear1(in=1024+512, out=4096)(out) out = LeakyReLU(slope=-0.2)(out) out = linear2(in=4096, out=1)(out)
classifier	**Input: $\tilde{\mathbf{x}}$** out = linear(in=1024, out=150)(out) out = softmax(out)
reconstructor	**Input: $\tilde{\mathbf{x}}$** out = linear(in=1024, out=4096)(out) out = LeakyReLU(slope=-0.2)(out) out = linear(in=4096, out=512)(out)

Table A8: The parameter choices for training Motif-based SGG models (section 1), RelDN-based SGG models (section 2), and the genertor model (section 3).

Parameter	Value	Description
MOTIF_IMS_PER_BATCH	16	batch size
MOTIF_BASE_LR	0.015	learning rate
MOTIF_MAX_ITER	40,000	iterations
RELDN_IMS_PER_BATCH	2	batch size
RELDN_BASE_LR	0.005	learning rate
RELDN_MAX_ITER	150,000	iterations
d_z	1024	dim of input z
BATCH_FG	128	batch size (adv. training)
D_TRAIN_ITER	5	D-over-G update iters
MAX_ITER_FG	55,000	iterations
GAN_LR	0.0001	learning rate
λ	10.0	the coef. for gp
β	0.1	the coef. for loss $\mathcal{L}_{cls}$
γ	0.1	the coef. for loss $\mathcal{L}_{recon}$

KuanChao Chu He is currently pursuing the doctoral degree with the Nakayama Laboratory, Graduate School of Information Science and Technology, The University of Tokyo. His research interests include novel object detection, data augmentation, scene graph detection, and deep learning.

Satoshi Yamazaki The master's and Ph.D. degrees in physics from Tohoku University, Japan, in 2011 and 2014, respectively. He is researcher in NEC Corporation since April 2014. His research interests include object tracking, person re-identification, and scene graph generation, and deep learning.

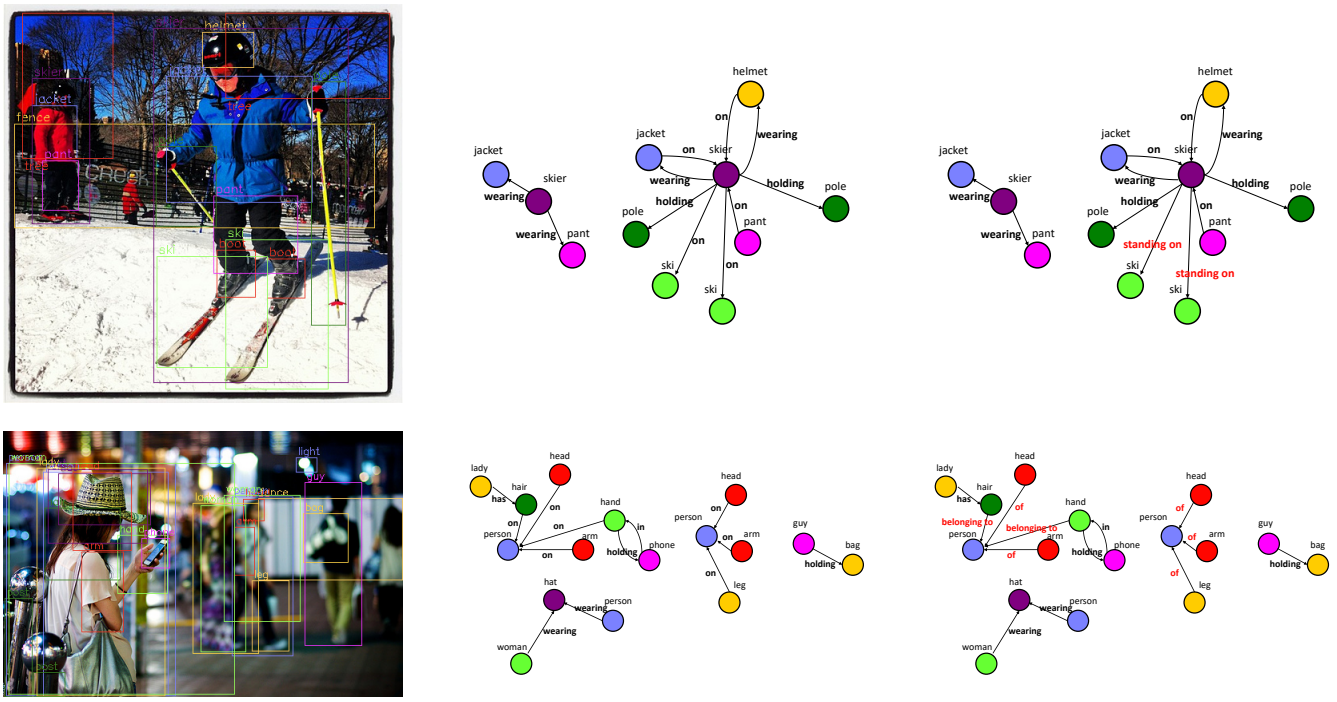

Fig. A1: Qualitative results of our method for the predcls task under the Motif+rwt setting: (Left) Images with bounding boxes, (Middle) Ground-truth scene graphs, and (Right) Predicted results. Isolated nodes have been omitted from the visualized scene graphs. The relations in red indicate discrepancies with the ground truth.

Hideki Nakayama received the master's and Ph.D. degrees in information science from the University of Tokyo, Japan, in 2008 and 2011, respectively. From 2012 to 2018, he was an Assistant Professor at the Graduate School of Information Science and Technology, The University of Tokyo, where he has been an Associate Professor, since April 2018. He is also a Faculty Member with the International Research Center for Neurointelligence (IRCN) and a Visiting Researcher with the National Institute of Advanced Industrial Science and Technology (AIST). His research interests include generic image recognition, natural language processing, and deep learning.