

# DisPositioNet: Disentangled Pose and Identity in Semantic Image Manipulation

Azade Farshad<sup>\*1,2</sup>  
azade.farshad@tum.de

Yousef Yeganeh<sup>\*1</sup>  
y.yeganeh@tum.de

Helisa Dhamo<sup>‡3</sup>  
helisa.dhamo@huawei.com

Federico Tombari<sup>1,4</sup>  
tombari@in.tum.de

Nassir Navab<sup>1,5</sup>  
nassir.navab@tum.de

<sup>1</sup> Technical University of Munich  
Germany

<sup>2</sup> Munich Center for Machine Learning  
Germany

<sup>3</sup> Huawei Noah's Ark Lab  
United Kingdom

<sup>4</sup> Google  
Switzerland

<sup>5</sup> Johns Hopkins University  
USA

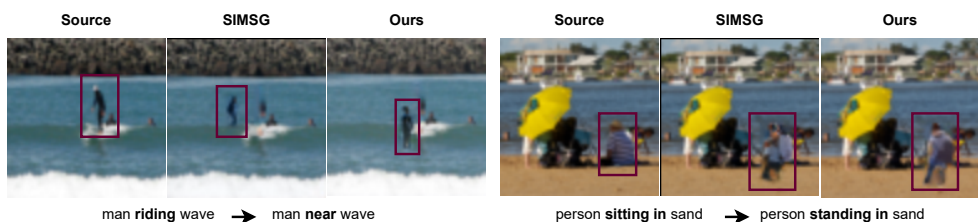


Figure 1: Our method is able to preserve the object features required for image manipulation and generate more realistic objects compared to SIMSG [5].

## Abstract

Graph representation of objects and their relations in a scene, known as a scene graph, provides a precise and discernible interface to manipulate a scene by modifying the nodes or the edges in the graph. Although existing works have shown promising results in modifying the placement and pose of objects, scene manipulation often leads to losing some visual characteristics like the appearance or identity of objects. In this work, we propose DisPositioNet, a model that learns a disentangled representation for each object for the task of image manipulation using scene graphs in a self-supervised manner. Our framework enables the disentanglement of the variational latent embeddings as well as the feature representation in the graph. In addition to producing more realistic images due to the decomposition of features like pose and identity, our method takes advantage of the probabilistic sampling in the intermediate features to generate more diverse images in object replacement or addition tasks. The results of our experiments show that disentangling the feature representations in the latent manifold of the model outperforms the previous works qualitatively and quantitatively on two public benchmarks. **Project Page:** <https://scenegenie.github.io/DispositioNet/>

# 1 Introduction

Image manipulation is a task of interest in computer vision, which consists of the partial synthesis, change, or removal of the content in a given image. In recent years, this task has been explored using deep generative models, in particular utilizing Generative Adversarial Networks (GANs) [14]. There have been different approaches towards image manipulation, in which the interface used to induce these changes by a user is an important choice. The utilization of segmentation maps for image modification [15, 65] requires direct manipulation of a semantic segmentation map at the pixel level. Recently, motivated by a more user-friendly interface, SIMSG [6] proposed a semantic manipulation framework using scene graphs. Scene graphs define a scene by considering the objects in the scene as the nodes in the graph, and the edges as the relationships between the objects. In semantic image manipulation using scene graphs, the user simply needs to change the nodes or edges in a graph that represents the scene. The manipulation of scenes in SIMSG [6] is performed by masking specific parts of data based on the manipulation mode, e.g., the object features or the bounding box information.

Despite the encouraging results, this model comes with a pitfall that the learned object features used for manipulation are intertwined, i.e., they encode both the pose and appearance features simultaneously. This becomes particularly evident when we want to preserve one of the aspects of an object while changing the other. For instance, in Fig. 1 we observe a relationship change setup. When the man changes from *riding* to *near the wave* (left), or from *sitting* to *standing in the sand* (right), SIMSG [6] (middle column) will lose some visual features of the man, in the process of adapting to the new pose, i.e. change of body shape or outfit color.

In this work, we propose **DisPositioNet**, a network for **Disentangled Pose** and **Identity** in Semantic Image Manipulation, which disentangles the object features using a self-supervised variational approach by employing two branches for encoding the pose and appearance features in the latent space. We hypothesize that, by disentangling the features in the image manipulation framework, the model would preserve features more reliably, and therefore generate more meaningful results. To disentangle the features further and make the extracted features from the scene graph more compatible with the variational embedding, we propose DSGN, a disentangled scene graph neural network for disentangled feature extraction from the scene graphs. We evaluate our model on standard benchmarks for image manipulation (Visual Genome [23], and Microsoft COCO [27]), showing superior performance compared to the baseline [6] both quantitatively and qualitatively. The qualitative results show that our proposed method specifically outperforms SIMSG [6] in cases where the appearance of the object should be preserved while changing its pose.

To summarize our contributions, we propose: 1) a self-supervised approach for disentanglement of pose and appearance for semantic image manipulation, that does not require label information for the disentanglement task, 2) a disentangled scene graph neural network, 3) a variational latent representation that provides higher diversity in image manipulation, 4) superior quantitative and qualitative performance compared to the state of the art on two public benchmarks. The source code of this work is provided in the supplementary material, and it will be publicly released upon its acceptance.

## 2 Related Work

**Scene Graphs** Scene graphs define a directed graph representation that describes an image [19], where objects are the nodes and their relationships are the edges. A broad line of works explores the generation of scene graphs from an image [12, 25, 33, 40, 47, 49, 50] and recently also point clouds [51, 54]. The task boils down to identifying the underlying objects in a scene and their visual relationships. A diverse set of approaches has been explored for this purpose, such as iterative message-passing [56], decomposition of the graph into sub-graphs [76] and attention mechanisms [57]. Recently, SceneGraphGen [10] explored this task unconditionally by learning an auto-regressive model on scene graphs. Scene graphs have shown to be a powerful alternative in conditional scene generation [6, 70, 79], and manipulation [8], which we will review as it follows.

**Image Generation** The recent advances in image generation, for the most part, emerged from Generative Adversarial Networks [11] and diffusion models [34]. In particular, the community has explored conditional variants [31] which enable image generation conditioned on various modalities. Pix2Pix [17] represents a model for general translation between different image domains. Further, CycleGAN [65] attempts this task by relaxing the need for image pairs for training. Other works [21, 22] explore unconditional generation, typically focused on a specific domain, such as faces. A line of methods [3, 37, 52] propose semantic image generation, where an image results from an input semantic map. Other works propose image generation from layout [48, 54], as a set of bounding boxes and class labels for each scene instance. More related to ours are methods that generate an image conditioned on a scene graph [11, 8, 18, 70], where the layout arises as an intermediate step to translate the graph structure into image space. Johnson et al. [70] introduced Sg2im, the first method that tackles this task supervised via a combined object-level and image-level GAN loss. Following work further improve the performance in this challenging task by utilizing per-object neural image features to increase diversity [10], exploiting meta-learning to better learn the highly diverse datasets (MIGS) [8], and employing contextual information to refine the layout (CoLoR) [18].

**Interactive Image Manipulation** This task represents a form of partial image generation, which usually comes with a user interface to indicate the subject of change [74]. Early works perform scene-level image editing in a hand-crafted manner, which replaces some image parts with sample patches from a database [16]. One form of manipulation is image inpainting, where a user can indicate a mask for removing and automatically filling an image area [39], that can be further extended with semantics [59] or edges [62, 60] to guide the missing area. Hong et al. [15] employ a learned model on a semantic layout representation, in which the user can make changes in the image by adding, moving, or removing bounding boxes. SESAME [35] allows the user to draw a mask with semantic labels on an image to indicate the category of the changed pixels. Similarly, in EditGAN [75] the user can modify a detailed object part segmentation map to alter object appearance. SIMSG [8] explores scene graphs as the interface, where the user can make changes in the nodes or edges of a graph to manipulate the image. Recently, Su et al. [66] proposed an improvement to this model by relying on masks instead of bounding boxes for the object placement. Different from these models, we want to model an object representation with disentangled appearance and pose.

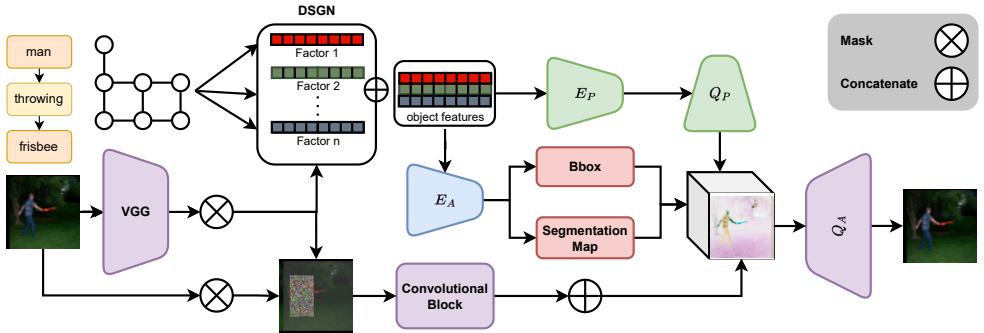


Figure 2: **DisPositionNet Overview.** Our disentangled image manipulation framework performs by disentangling the graph representations, as well as the variational embeddings through learning the feature transformations.

**Disentangled Representation Learning** Learning disentangled representation has been explored in many works using variational autoencoders [7, 50], e.g., for changing the digit and handwriting in the MNIST dataset. Initial works on disentangled representation learning [4] focused on variational mutual information maximization or decreasing the channel capacity of the variational autoencoder [14]. Some recent works focus on disentanglement using deformable networks [43, 55], contrastive learning [2, 58] or disentangling identity and pose for face manipulation [62]. Many of these works require labeled information for conditioning the model on the specific attributes for the disentangling. They also disentangle the data without knowing which feature belongs to which factor. Recently, [45] was proposed as an unsupervised way to disentangle pose and appearance. They used two branches to predict the image’s transformation parameters and apply the learned transformation to the appearance features. Disentanglement in graph neural networks (GNN) has been previously explored in [30, 53] where the graph features are divided into different factors that help disentangle the latent representation. These approaches, however, operate on regular graphs, which only consider neighboring nodes when computing the features. In contrast, GNNs designed for scene graphs also modulate edge features into the GNN network.

### 3 Method

Our goal is to learn a disentangled representation for the appearance and pose of the objects in the latent space for the semantic image manipulation task to preserve the features of specific attributes. As it follows, we first discuss the semantic image manipulation framework. Then, we describe our proposed disentangled graph model and our variational disentanglement approach in detail. Fig. 2 shows an overview of our method.

**Semantic Image Manipulation** Given an image  $I$  and its corresponding scene graph  $\mathcal{G} = \{\mathcal{O}, \mathcal{R}\}$ , where  $\mathcal{O}$  is the set of objects (nodes) in the scene and  $\mathcal{R}$  represents the set of relationships (edges) between the objects, the goal is to obtain a modified image  $I^*$  based on an altered version of the scene graph  $\mathcal{G}^*$ . Inspired by SIMSG [5] we formulate this image manipulation task via a reconstruction proxy objective, such that we do not need to rely on

image pairs with changes for the training. To enable control on specific object attributes, the semantic graph representation is extended to obtain an augmented graph, where each node contains a semantic class embedding, a bounding box  $x$ , and a neural visual feature  $z_I$ . During training, object regions in the image, visual features, or bounding boxes are randomly masked using a noise vector, and the model’s objective is to reconstruct the masked parts using the information from the scene graph  $\mathcal{G}$  and the remaining regions in the image.  $\mathcal{G}$  is defined as a set of triplets  $\mathcal{G}_i = (s_i, r_i, o_i)$ , where  $s_i, r_i, o_i$  are the subject, predicate and object respectively. Each object in the graph belongs to a class of object categories  $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ . Image features  $z_I$  are extracted from the input image using a pre-trained classifier network such as VGG16 [44]. The graph triplets are fed to a scene graph neural network (SGN) for message passing between the nodes.  $z_G$  is obtained from SGN with parameters  $\Phi$ , which processes the scene graph  $\mathcal{G}$ , the input bounding boxes  $x$ , and visual features  $z_I$ .

To disentangle the features based on the pose and appearance, we harness two encoder networks, namely  $E_A$  and  $E_P$ , that receive the per-object features  $z_G$  as input and produce appearance features  $z_{GA}$  and pose features  $z_{GP}$  respectively. The object bounding boxes and pseudo-segmentation maps are predicted utilizing two networks that receive  $z_{GA}$  as input. Further, the scene layout  $z_l$  is constructed by projecting the appearance features  $z_{GA}$  of each object in the image space, in the regions indicated by the respective predicted bounding boxes and segmentation maps. We further employ a pose decoder network  $Q_P$  to predict a set of transformation parameters  $\gamma$ . These parameters are used to construct a transformation function  $\tau$ , which is applied to the pooled object features from the scene layout  $z_l$ . The object feature pooling is performed by cropping  $z_I$  using the bounding boxes  $x$  and applying the per-object transformations on the cropped feature vectors. Finally, the reconstructed image  $\tilde{I}$  is generated by passing the transformed layout  $\tau(z_l)$  to the image decoder network  $Q_A$ .

**Disentangled Graph Neural Network** One main limitation in the SIMSG formulation is that the object features extracted by the SGN are entangled. Therefore, to increase the disentanglement between pose and appearance even further, we propose using a disentangled graph neural network. Inspired by [45], we propose *DSGN*, a disentangled scene graph network. DSGN not only considers the nodes in the graph as in [45], but it also combines the edge features (here predicates) in the disentangled feature extraction, as this provides crucial information for the task at hand. Our network is thus adapted for triplets of the form  $\mathcal{G}_i = (s_i, r_i, o_i)$ . The DSGN utilizes disentangled convolutional layers combined with neighborhood routing mechanism [45] for projection of the features into different subspaces. The neighborhood routing mechanism actively distinguishes the latent factor that could have caused the edge between a node and its neighbors. This would assign the neighbor to another channel to extract the features for that specific factor. The DSGN receives the triplets  $\mathcal{G}_i = (s_i, r_i, o_i)$  as input, where  $o_i, s_i \in \mathcal{O}$  and  $r_i \in \mathcal{R}$ . Each layer  $e$  in the DSGN represents a function  $f_e(\cdot)$  which applies the edge features to the nodes in the graph and their neighbours:

$$(\alpha_{ij}^{(t+1)}, \rho_{ij}^{(t+1)}, \beta_{ij}^{(t+1)}) = f_e \left( v_i^{(t)}, \rho_{ij}^{(t)}, v_j^{(t)} \right), \quad (1)$$

with  $v_i^{(0)} = o_i$ , where  $t$  represents the layers of the DSGN, which has a total of  $T$  layers. The input is first processed by a Sparse Input Layer [45] to decompose the node features  $v_i^{(t)}$  into  $k$  factors. The  $k$  node features are then passed through  $k$  separate neighbourhood routing layers. Finally, the object features  $v_i^{(T)}$  are computed by concatenating the object features

from all factors  $k$ .

**Disentangled Variational Embedding** Our model is enforced to disentangle the perspective and appearance features in the latent embedding, through modelling and predicting the transformation in the features, based on [45]. The variational embedding disentanglement happens by employing two encoder and two decoder networks. The object features  $z_G$  are passed to the variational encoders  $E_A, E_P$ , that are composed of two subnetworks that model the mean  $\mu$  and variance  $\sigma$  of the data. They both output the latent representation  $z$ , obtained by applying the reparameterization trick  $z = \mu + \sigma\epsilon$ , where  $\epsilon$  is a random noise vector.

$E_A$  encodes the appearance features, while  $E_P$  encodes the perspective information. The transformation  $\gamma$  for each object is predicted by a simple MLP network  $Q_P$ , which is then applied to the pooled object patches from the scene layout  $z_l$ . The intention behind predicting and applying the transformation  $\gamma$  by  $Q_P$  is to separate the pose information in the pose branch and enforce the model to only learn the appearance features in  $E_A$ . Finally, the image is reconstructed from the scene layout  $z_l$  by  $Q_A$ , which is the SPADE [46] generator here.

We define the affine transformation function  $\tau$  given the input  $z$  as follows:

$$\tau_{\gamma, \text{affine}}(z) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_{z_a} & 0 \\ 0 & \delta_{z_b} \end{bmatrix} + \begin{bmatrix} t_{z_a} \\ t_{z_b} \end{bmatrix} \quad (2)$$

where  $\alpha$  is the rotation angle,  $m$  is the shear value,  $\delta_{z_a}$  and  $\delta_{z_b}$  are the scaling factors and  $t_{z_a}, t_{z_b}$  are the translation parameters. These parameters, defined by  $\gamma$  are modelled by an MLP represented by  $Q_P$  that outputs these values per object.

**Objective Functions** The loss terms used for training our model are a combination of original losses used in [46] and variational terms. The generative adversarial objective is:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{q \sim p_{\text{data}}} \log D(q) + \mathbb{E}_{q \sim p_g} \log(1 - D(q)), \quad (3)$$

where  $p_g$  denotes the distribution of fake / generated images or object patches,  $p_{\text{data}}$  is the distribution of the ground truth images or objects, and  $q$  defines the input to the discriminator network  $D$  sampled from the ground truth or generated data distributions. In addition to the global image discriminator  $D_{\text{image}}$ , an object discriminator  $D_{\text{obj}}$  is used for cropped patches of objects in the image to improve the appearance and realism of the objects. The bounding box prediction loss is defined as  $\mathcal{L}_{\text{bbox}} = \lambda_b \|x_i - \hat{x}_i\|_1$ , while the generative objective is:

$$\begin{aligned} \mathcal{L}_{\text{generative}} = & \lambda_g \min_G \max_D \mathcal{L}_{\text{GAN, image}} + \lambda_o \min_G \max_D \mathcal{L}_{\text{GAN, obj}} \\ & + \lambda_a \mathcal{L}_{\text{aux, obj}} + \mathcal{L}_{\text{rec}} + \lambda_p \mathcal{L}_p + \lambda_f \mathcal{L}_f, \end{aligned} \quad (4)$$

where  $\lambda_g, \lambda_o, \lambda_a$  are the constant weight multipliers.  $\mathcal{L}_{\text{aux, obj}}$  is an auxiliary object classifier loss [46], and  $\mathcal{L}_p, \mathcal{L}_f$  are respectively the perceptual and GAN feature loss terms borrowed from the SPADE generator [46], and  $\mathcal{L}_{\text{rec}} = \|I - \tilde{I}\|_1$  is the image reconstruction loss.

The variational objective for feature disentanglement tries to minimize the evidence lower bound (ELBO):

$$\begin{aligned} \mathcal{L}_{\text{var}} = & \mathbb{E}_{q_A, q_P} [\log(p(I|z_{GA}, z_{GP}))] - D_{\text{KL}}(q_P(z_{GP}|I) || p(z_{GP})) \\ & - \mathbb{E}_{q_P} [D_{\text{KL}}(q_A(z_{GA}|I) || p(z_{GA}))]. \end{aligned} \quad (5)$$

Then, the final objective becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{var} + \mathcal{L}_{generative} + \mathcal{L}_{bbox} \quad (6)$$

**Table 1: Image reconstruction on Visual Genome.** We compare the results of our method to previous works using ground truth (GT) and predicted scene graphs. In the experiments denoted by (Generative), the whole input image is masked. N/A: Not Applicable.

Method	Decoder	All pixels					RoI only	
		MAE ↓	SSIM ↑	LPIPS ↓	FID ↓	IS ↑	MAE ↓	SSIM ↑
Generative, GT Graphs								
ISG [10]	Pix2pixHD	46.44	28.10	0.32	58.73	6.64±0.07	N/A	N/A
SIMSG [10]	SPADE	41.88	34.89	0.27	44.27	7.86±0.49	N/A	N/A
DisPositioNet (Ours)	SPADE	<b>41.62</b>	<b>35.30</b>	<b>0.26</b>	<b>40.75</b>	<b>7.93±0.36</b>	N/A	N/A
GT Graphs								
Cond-sg2im [10]	CRN	14.25	84.42	0.081	13.40	11.14±0.80	29.05	52.51
SIMSG [10]	SPADE	8.61	87.55	0.050	<b>7.54</b>	<b>12.07±0.97</b>	<b>21.62</b>	<b>58.51</b>
DisPositioNet (Ours)	SPADE	<b>8.41</b>	<b>87.56</b>	<b>0.048</b>	7.66	11.65±0.58	21.76	58.18
Predicted Graphs								
SIMSG [10]	SPADE	13.82	83.98	0.077	16.69	10.61±0.37	28.82	49.34
DisPositioNet (Ours)	SPADE	<b>9.39</b>	<b>86.91</b>	<b>0.052</b>	<b>14.42</b>	<b>10.69±0.33</b>	<b>25.40</b>	<b>51.85</b>

## 4 Experiments

In this section, we first discuss our framework’s setup, including the hyperparameters and the metrics. Then we ablate the different components of our model, and report the quantitative and qualitative results of our experiments compared to previous work. We also provide the results of our performed user study. Finally, we discuss the results and the limitations of our work.

### 4.1 Experimental Setup

We evaluate our method on Visual Genome (VG) [23] and Microsoft COCO [27] datasets which are commonly used in the image generation using scene graphs literature. The qualitative results on COCO are included in the supplementary material due to the space limitations.

**Evaluation Metrics** To evaluate the quality of generated images by our model, we use common similarity metrics used for GANs such as Inception Score (IS) [14], Frechet Inception Distance (FID) [14], structural similarity metric (SSIM) [14], Perceptual Similarity (LPIPS) [14] and the Mean Absolute Error (MAE). We also measure the MAE and SSIM for the Region of Interest (RoI) where the change or reconstruction happens.

**Implementation Details** All models were trained on  $64 \times 64$  images with batch size of 32. The visual feature extraction model is a VGG-16 pretrained on ImageNet. The learning rate for all models is  $2e - 4$ , and the disentangling factor  $k$  in the Disentangled SGN is equal to 16. All models were trained for 300k iterations on VG and COCO. The architectures of



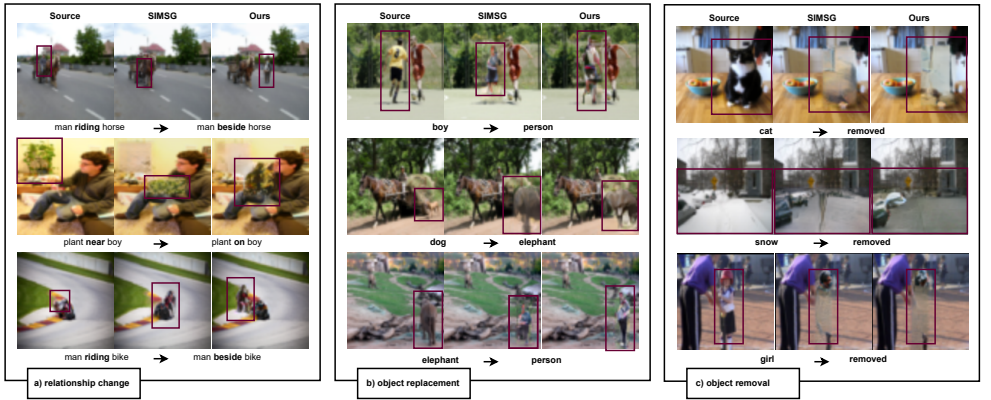


Figure 3: **Qualitative comparison to SIMSG [5] on VG.** It can be seen that in a) the plant has a more realistic appearance and a more similar shape to the original object, the same applies to b) where the boy and the elephant are changed to person. For the object removal in c) there are some artifacts visible after the removal of the cat and snow, however the images generated by our method do not have these artifacts and look more realistic.

$E_P$ ,  $E_A$ , and  $D_P$  are MLPs with two FC layers with 64 filters, 1 BN layer, and a LeakyReLU activation function. The decoder and discriminator architectures follow [5]. The values of the hyperparameters were obtained empirically or based on previous works. The slight difference between the reported values and the ones in [5] could be due to library version differences. We report the details of network architectures in the supplementary material.

**Modification Modes** During the testing phase, four modification modes are supported, i.e. relationship change, object replacement, object removal, and object addition. The model receives the source image, and the desired modification on the graph as input. Specific features are masked based on the modification mode and the target image is generated by the decoder. E.g., for relationship changes, the object features are retained while the bounding box features are masked. On the other hand, for object replacement, the object features are dropped while the bounding box features are preserved.

## 4.2 Results

**Quantitative Results** Quantitative evaluation of image manipulation methods on a real-world dataset is a difficult task due to the lack of paired source and target data. Therefore, following previous work [5], we evaluate our method based on the reconstruction quality. The input images are partially masked, and the goal of the model is to reconstruct the masked parts from the information in the scene graph. The results of our experiments are presented in Tab. 1 and Tab. 3. In the generative mode, the whole image is masked, and the model is generated purely from the scene graph to evaluate its image generation performance. The models are given either ground truth graphs as input or predicted ones by a scene graph to image model [26]. The results show that, the DisPositionNet model outperforms the state-of-the-art in almost all metrics and scenarios. We also provide the results of our user study



on the comparison between SIMSG and DisPositionNet for different manipulation modes in Tab. 4. The user study details are provided in the supplementary material.

**Qualitative Results** Some qualitative results of our method on VG dataset are shown in Fig. 3. As it can be seen, our proposed method is able to learn better feature representations and therefore generate more meaningful results. We also provide some qualitative examples on diversity in the supplementary material, and show that in contrast to SIMSG, our model is able to generate diverse images in terms of color and texture.

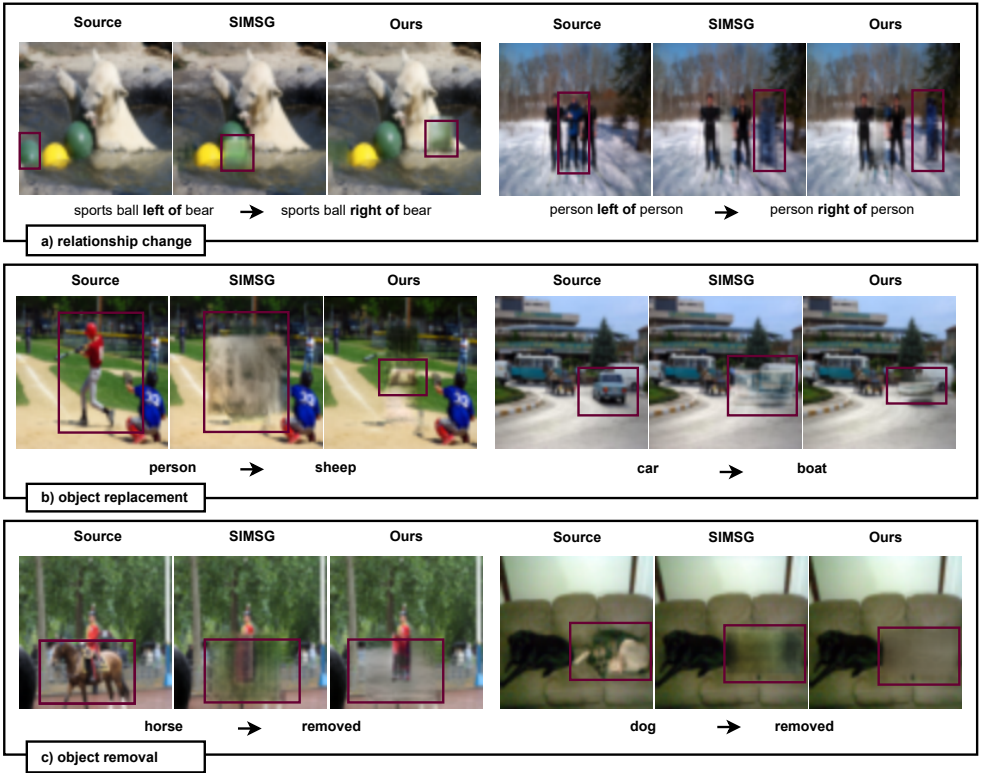


Figure 4: **Qualitative results for image manipulation on COCO.** Our method shows more accurate positioning and better visual appearance in three main image manipulation tasks of (a) relationship change, (b) object replacement and (c) object removal.

**Ablation Study** The results of our ablation study are reported in Tab. 2. First, we present the model performance without the disentanglement. Then, we evaluate the effect of disentangling the latent embeddings. Finally, we show the model performance with disentanglement in both latent embedding and the graph features. Notably, the disentanglement of both components leads to an improvement in most metrics.

Table 2: Ablation Study on VG

Disentanglement		All pixels			RoI only	
Embeddings	Graph	MAE ↓	SSIM ↑	LPIPS ↓	MAE ↓	SSIM ↑
Generative						
—	—	41.88	34.89	0.27	N/A	N/A
✓	—	41.80	35.18	0.26	N/A	N/A
✓	✓	<b>41.62</b>	<b>35.30</b>	<b>0.26</b>	N/A	N/A
GT Graphs						
—	—	8.61	87.55	0.050	<b>21.62</b>	<b>58.51</b>
✓	—	8.47	87.53	0.048	21.77	58.30
✓	✓	<b>8.41</b>	<b>87.56</b>	<b>0.048</b>	21.76	58.18
Predicted Graphs						
—	—	13.82	83.98	0.077	28.82	49.34
✓	—	9.65	86.68	0.054	25.62	51.19
✓	✓	<b>9.39</b>	<b>86.91</b>	<b>0.052</b>	<b>25.40</b>	<b>51.85</b>

Table 3: Image reconstruction on COCO

Method	All pixels			RoI only	
	MAE ↓	SSIM ↑	LPIPS ↓	MAE ↓	SSIM ↑
Generative					
SIMSG [9]	54.03	24.12	0.490	N/A	N/A
DisPositioNet (Ours)	<b>51.07</b>	<b>26.53</b>	<b>0.418</b>	N/A	N/A
Non Generative					
SIMSG [9]	9.36	87.00	0.086	27.68	49.93
DisPositioNet (Ours)	<b>9.24</b>	<b>88.26</b>	<b>0.057</b>	<b>27.52</b>	<b>50.35</b>

Table 4: User study on VG

Method	Removal	Replacement	Relationship Change	Mean
SIMSG [9]	14.06	27.68	26.95	23.51
DisPositioNet (Ours)	<b>85.94</b>	<b>72.32</b>	<b>73.04</b>	<b>76.49</b>

## 4.3 Discussion

We showed that our proposed model outperforms the previous works in all scenarios by generating images with higher quality and more meaningful results. We also showed that the generated images by our method have higher diversity and less artifacts. In our user study, the users were given the option to choose which model performs better image manipulation in terms of image quality and how well the change in the image corresponds to the modification in the graph. The DisPositioNet model was chosen as the best model compared to SIMSG [9] in 76.49% of the cases. Nevertheless, our method has some limitations similar to the related work, which we discuss here.

**Limitations** The dominant limitations of our approach are manipulating high-resolution images and the reconstruction of faces and complex scenes. We believe that these limitations originate from the difficulty in generating high-quality images from scene graphs [20] that could be due to the wild nature of images in the VG dataset and occasional errors in the scene graph annotations. We assume that, it would be possible to overcome this issue by using a higher quality dataset with scene graphs and semantic segmentation annotations. Regarding the face reconstruction problem, although this is an easy task given datasets of pure face images, the model fails to generalize well to the faces when combined with images in the wild. We present some failure case examples in the supplementary material.

## 5 Conclusions

We presented a novel disentangling framework for image manipulation using scene graphs. The results of our experiments showed that using disentangled representation in the latent embedding for semantic image manipulation is an effective way to improve image generation and manipulation quality. The variational representation for object features enables generating diverse images compared to previous work. Further, we showed that using a disentangled graph neural network for extracting the scene graph features provides more meaningful and useful features for the disentangled latent embedding, resulting in higher reconstruction performance. As a future direction, we consider improving the decoder network by taking advantage of diffusion models.

## References

- [1] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *ICCV*, pages 4561–4569, 2019.
- [2] Junwen Bai, Weiran Wang, and Carla Gomes. Contrastively disentangled sequential variational autoencoder. *arXiv preprint arXiv:2110.12091*, 2021.
- [3] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, pages 1511–1520, 2017.
- [4] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016.
- [5] Helisa Dharmo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D Hager, Federico Tombari, and Christian Rupprecht. Semantic image manipulation using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5213–5222, 2020.
- [6] Helisa Dharmo, Fabian Manhardt, Nassir Navab, and Federico Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [7] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018.
- [8] Azade Farshad, Sabrina Musatian, Helisa Dharmo, and Nassir Navab. Migs: Meta image generation from scene graphs. In *BMVC*, 2021.
- [9] Jean Feng and Noah Simon. Sparse-input neural networks for high-dimensional non-parametric regression and classification. *arXiv preprint arXiv:1711.07592*, 2017.
- [10] Sarthak Garg, Helisa Dharmo, Azade Farshad, Sabrina Musatian, Nassir Navab, and Federico Tombari. Unconditional scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16362–16371, 2021.
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [12] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *NeurIPS*, 2018.
- [13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

- [14] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=Sy2fzU9gl>.
- [15] Seunghoon Hong, Xinchun Yan, Thomas E Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. In *Advances in Neural Information Processing Systems*, pages 2713–2723, 2018.
- [16] Shi-Min Hu, Fang-Lue Zhang, Miao Wang, Ralph R Martin, and Jue Wang. Patchnet: A patch-based image representation for interactive library-driven image editing. *ACM Transactions on Graphics (TOG)*, 32(6):1–12, 2013.
- [17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [18] Maor Ivgi, Yaniv Benny, Avichai Ben-David, Jonathan Berant, and Lior Wolf. Scene graph to image generation with contextualized object layout refinement. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2428–2432. IEEE, 2021.
- [19] J. Johnson, R. Krishna, M. Stark, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015.
- [20] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018.
- [21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.
- [22] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423*, 2021.
- [23] 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(1):32–73, 2017.
- [24] Lei Li, Kai Fan, and Chun Yuan. Cross-modal representation learning and relation reasoning for bidirectional adaptive manipulation. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3222–3228. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/447. URL <https://doi.org/10.24963/ijcai.2022/447>. Main Track.
- [25] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, pages 1261–1270, 2017.

- [26] 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, 2018.
- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [28] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *arXiv preprint arXiv:2111.03186*, 2021.
- [29] Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B Tenenbaum. End-to-end optimization of scene layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3754–3763, 2020.
- [30] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4212–4221. PMLR, 09–15 Jun 2019.
- [31] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [32] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edge-connect: Structure guided image inpainting using edge prediction. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.
- [33] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *NeurIPS*, pages 2171–2180, 2017.
- [34] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.
- [35] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. SESAME: Semantic Editing of Scenes by Adding, Manipulating or Erasing Objects. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 394–411, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58542-6.
- [36] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017.
- [37] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.
- [38] Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contradistinctive generative autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 823–832, 2021.

- [39] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.
- [40] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. *arXiv preprint arXiv:1811.10696*, 2018.
- [41] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in Neural Information Processing Systems*, 30, 2017.
- [42] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016.
- [43] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proceedings of the European conference on computer vision (ECCV)*, pages 650–665, 2018.
- [44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [45] Nicki Skafte and Søren Hauberg. Explicit disentanglement of appearance and perspective in generative models. *Advances in Neural Information Processing Systems*, 32: 1018–1028, 2019.
- [46] Sitong Su, Lianli Gao, Junchen Zhu, Jie Shao, and Jingkuan Song. Fully functional image manipulation using scene graphs in a bounding-box free way. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1784–1792, 2021.
- [47] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13936–13945, 2021.
- [48] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *ICCV*, October 2019.
- [49] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiabin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725, 2020.
- [50] N Joseph Tatro, Stefan C Schonsheck, and Rongjie Lai. Unsupervised geometric disentanglement for surfaces via cfan-vae. *arXiv preprint arXiv:2005.11622*, 2020.
- [51] Johanna Wald, Helisa Dharmo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [52] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

- [53] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [54] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7515–7525, 2021.
- [55] Xianglei Xing, Tian Han, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Unsupervised disentangling of appearance and geometry by deformable generator network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10354–10363, 2019.
- [56] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017.
- [57] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, pages 670–685, 2018.
- [58] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. *Advances in Neural Information Processing Systems*, 33, 2020.
- [59] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, pages 5485–5493, 2017.
- [60] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [61] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018.
- [62] Bassel Zeno, Ilya Kalinovskiy, and Yuri Matveev. Ip-gan: learning identity and pose disentanglement in generative adversarial networks. In *International Conference on Artificial Neural Networks*, pages 535–547. Springer, 2019.
- [63] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.
- [64] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019.
- [65] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.