Learning to Compose Visual Relations

Nan Liu *
University of Michigan
liunan@umich.edu

Shuang Li *
MIT CSAIL
lishuang@mit.edu

Yilun Du *
MIT CSAIL
yilundu@mit.edu

Joshua B. Tenenbaum
MIT CSAIL, BCS, CBMM
jbt@mit.edu

Antonio Torralba
MIT CSAIL
torralba@mit.edu

Abstract

The visual world around us can be described as a structured set of objects and their associated relations. An image of a room may be conjured given only the description of the underlying objects and their associated relations. While there has been significant work on designing deep neural networks which may compose individual objects together, less work has been done on composing the individual relations between objects. A principal difficulty is that while the placement of objects is mutually independent, their relations are entangled and dependent on each other. To circumvent this issue, existing works primarily compose relations by utilizing a holistic encoder, in the form of text or graphs. In this work, we instead propose to represent each relation as an unnormalized density (an energy-based model), enabling us to compose separate relations in a factorized manner. We show that such a factorized decomposition allows the model to both generate and edit scenes that have multiple sets of relations more faithfully. We further show that decomposition enables our model to effectively understand the underlying relational scene structure. Project page at: https://composevisualrelations.github.io/

1 Introduction

The ability to reason about the component objects and their relations in a scene is key for a wide variety of robotics and AI tasks, such as multistep manipulation planning [11], concept learning [24], navigation [42], and dynamics prediction [3]. While a large body of work has explored inferring and understanding the underlying objects in a scene, robustly understanding the component relations in a scene remains a challenging task. In this work, we explore how to robustly understand relational scene description (Figure 1).

Naively, one approach towards understanding relational scene descriptions is to utilize existing multi-modal language and vision models. Such an approach has recently achieved great success in DALL-E [35] and CLIP [34], both of which show compelling results on encoding object properties with language. However, when these approaches are instead utilized to encode relations between objects, their performance rapidly deteriorates, as shown in [35] and which we further illustrate in Figure 7. We posit that the lack of *compositionality* in the language encoder prevents it from capturing all the underlying relations in an image.

To remedy this issue, we propose instead to *factorize* the scene description with respect to each individual relation. Separate models are utilized to encode each individual relation, which are then subsequently *composed* together to represent a relational scene description. The most straightforward

* indicates equal contribution

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

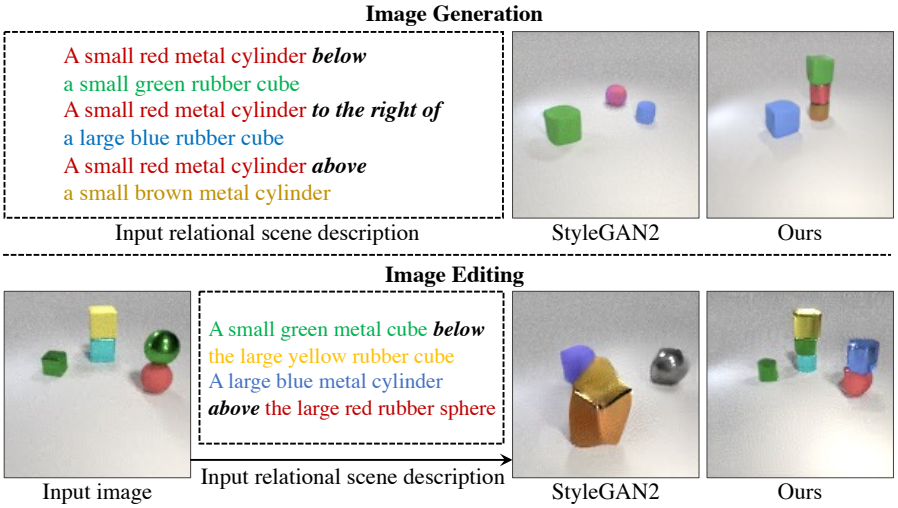

Figure 1: Our model can generate and edit images with multiple composed relations. **Top**: Image generation results based on relational scene descriptions. **Bottom**: Image editing results based on relational scene descriptions.

approach is to specify distinct regions of an image in which each relation can be located, as well as a composed relation description corresponding to the combination of all these regions.

Such an approach has significant drawbacks. In practice, the location of one pair of objects in a relation description may be heavily influenced by the location of objects specified by another relation description. Specifying a priori the exact location of a relation will thus severely hamper the number of possible scenes that can be realized with a given set of relations. Is it possible to factorize relational descriptions of a scene and generate images that incorporate each given relation description simultaneously?

In this work, we propose to represent and factorize individual relations as unnormalized densities using Energy-Based Models. A relational scene description is represented as the product of the individual probability distributions across relations, with each individual relation specifying a separate probability distribution over images. Such a composition enables interactions between multiple relations to be modeled.

We show that this resultant framework enables us to reliably capture and generate images with multiple composed relational descriptions. It further enables us to edit images to have a desired set of relations. Finally, by measuring the relative densities assigned to different relational descriptions, we are able to infer the objects and their relations in a scene for downstream tasks, such as image-to-text retrieval and classification.

There are three main contributions of our work. First, we present a framework to factorize and compose separate object relations. We show that the proposed framework is able to generate and edit images with multiple composed relations and significantly outperforms baseline approaches. Secondly, we show that our approach is able to infer the underlying relational scene descriptions and is robust enough in understanding semantically equivalent relational scene descriptions. Finally, we show that our approach can generalize to a previously unseen relation description, even if the underlying objects and descriptions are from a separate dataset not seen during training. We believe that such generalization is crucial for a general artificial intelligence system to adapt to the infinite number of variations of the world around it.

2 Related Work

Language Guided Scene Generation. A large body of work has explored scene generation utilizing text descriptions [13, 17, 20, 29, 33, 35, 37–39, 45–47]. In contrast to our work, prior work [17, 37, 46, 47] have focused on generating images given only a limited number of relation descriptions. Recently, [35] shows compelling results on utilizing text to generate images, but also explicitly state that relational generation was a weakness of the model. In this work, we seek to tackle how we may generate images given an underlying relational description.

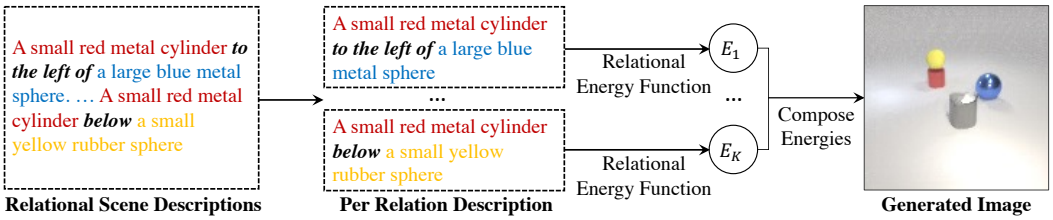

Figure 2: Overview of our pipeline for understanding a relational scene description. A relational scene description is split into a set of underlying relation descriptions. Individual relation descriptions are represented as EBMs which are subsequently composed together to generate an image.

Visual Relation Understanding. To understand visual relations in a scene, many works applied neural networks to graphs [1–3, 5, 14, 15, 18, 20, 27, 28, 31, 36]. Raposo et al. [36] proposed Relation Network (RN) to explicitly compute relations from static input scenes while we implicitly encode relational information to generate and edit images based on given relations and objects. Johnson et al. [20] proposed to condition image generation on relations by utilizing scene graphs which were further explored in [1, 14, 15, 18, 28, 30, 43]. However, these approaches require excessive supervisions, such as bounding boxes, segmentation masks, etc., to infer and generate the underlying relations in an image. Such a setting greatly restricts the possible combinations of individual relations in a scene. In our work, we present a method that enables us to generate images given only a relational scene description.

Energy Based Models. Our work is related to existing work on energy-based models [6, 8, 10, 12, 22, 32, 41, 44]. Most similar to our work is that of [7], which proposes a framework utilizing EBMs to compose several object descriptions together. In contrast, in this work, we study the problem of how we may compose relational descriptions together, an important and challenging task for existing text description understanding systems.

3 Method

Given a training dataset $\mathcal{C} = \{\boldsymbol{x}_i, R_i\}_{i=1}^N$ with N distinct images $\boldsymbol{x}_i \in \mathbb{R}^D$ and associated relational descriptions $R_i := \{w_i^1, \dots w_i^{n_i}\}$, we aim at learning the underlying probability distribution $p_\theta(\boldsymbol{x}|R)$ — the probability distributions of an image given a corresponding relational description. To represent $p_\theta(\boldsymbol{x}|R)$, we split a relational description R into K separate relations $\{r_1 \cdots, r_K\}$ and model each component relation separately using a probability distribution $p_\theta(\boldsymbol{x}|r_k)$ which is represented as an Energy-Based Model. Our overall scene probability distribution is then modeled by a composition of individual probability distributions for each relation description $p_\theta(\boldsymbol{x}|R) \propto \prod_k p_\theta(\boldsymbol{x}|r_k)$.

In this section, we give an overview of our approach towards factorizing and representing a relational scene description. We first provide a background overview of Energy-Based Models (EBMs) in Section 3.1. We then describe how we may parameterize individual relational probability distributions with EBMs in Section 3.2. We further describe how we compose relational probability distributions to model a relational scene description in Section 3.3. Finally, we illustrate downstream applications of our relational scene understanding model in Section 3.4.

3.1 Energy-Based Models

We model each relational probability distribution utilizing an Energy-Based Model (EBM) [6, 25]. EBMs are a class of unnormalized probability models, which parameterize a probability distribution $p_\theta(\boldsymbol{x})$ utilizing a learned energy function $E_\theta(\mathbf{x})$:

$$p_\theta(\mathbf{x}) \propto e^{-E_\theta(\mathbf{x})}. \tag{1}$$

EBMs are typically trained utilizing contrastive divergence [16], where energies of training data-points are decreased and energies of sampled data points from $p_\theta(\boldsymbol{x})$ are increased. We adopt the training code and models from [8] to train our EBMs. To generate samples from an EBM $p_\theta(\boldsymbol{x})$, we utilize MCMC sampling on the underlying distribution, and Langevin dynamics, which refines a data sample iteratively from a random noise:

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k-1}) + \omega^k,\ \omega^k \sim \mathcal{N}(0, \lambda) \tag{2}$$

where k refers to the iteration and λ is the step size, utilizing the gradient of the energy function with respect to the input sample $\nabla_{\mathbf{x}} E_\theta$, and ω^k is sampled from a Gaussian noise. EBMs enable us to naturally compose separate probability distributions together [7]. In particular, given a set of independent marginal distributions $\{p_\theta^i(\boldsymbol{x})\}$, the joint probability distribution is represented as:

$$\prod_i p_\theta^i(\boldsymbol{x}) \propto e^{-\sum_i E_\theta^i(\boldsymbol{x})}, \tag{3}$$

where we utilize Langevin dynamics to sample from the resultant joint probability distribution.

3.2 Learning Relational Energy Functions

Given a scene relation r_i, described using a set of words $\{w_i^1, \dots w_i^n\}$, we seek to learn a conditional EBM which models the underlying probability distribution $p(\boldsymbol{x}|r_i)$:

$$p_\theta(\boldsymbol{x}; r_i) \propto e^{-E_\theta^i(\boldsymbol{x}|\text{Enc}(r_i))}, \tag{4}$$

where $p(\boldsymbol{x}|r_i)$ represents the probability distribution over images given relation r_i and Enc(r_i) denotes a text encoder for relation r_i.

The most straightforward manner to encode relational scene descriptions is to encode the entire sentence using an existing text encoder, such as CLIP [34]. However, we find that such an approach cannot capture scene relations as shown in Appendix Section A. An issue with such an approach is that the sentence encoder loses or masks the information captured by the relation token in r_i.

To enforce that the underlying relation token in r_i is effectively encoded, we instead propose to decompose the relation r_i into a relation triplet (r_i', o_i^1, o_i^2), where r_i' is the relation token, *e.g.* "below", "to the right of", o_i^1 is the description of the first object, and o_i^2 is the description of the second object appeared in r_i. Each separate entry in the relation triplet is then separately embedded.

Such an encoding scheme encourages the models to encode underlying objects and relations in a scene, enabling us to effectively model the relational distribution. We explored two separate approaches to embed the underlying object descriptions into our relational energy function.

CLIP Embedding. One approach we consider is to directly utilize CLIP to obtain the embedding of an object description. Such an approach may potentially enable us to generalize relations in a zero-shot manner to new objects by utilizing CLIP's underlying embedding of the object, but may also hurt learning if the underlying object embedding does not distinctly separate two object descriptions.

Random Initialization. Alternatively, we may encode an object description using a learned embedding layer that is trained from scratch. In this approach, we extract a scene embedding by concatenating the learned embeddings of color, shape, material, and size of any two objects, o_i^1, o_i^2, and their relation r_i'.

3.3 Representing Relational Scene Descriptions

Given an underlying scene description R, we represent the underlying probability distribution $p(\boldsymbol{x}|R)$ by factorizing it as a product of probabilities over the the underlying relations r_k inside R.

Given the separate relational energy functions learned in Section 3.2, this probability $p(\boldsymbol{x}|R)$ is proportional to

$$p(\boldsymbol{x}|R) = e^{-E_\theta^k(\boldsymbol{x}|R)} \propto \prod_k p(\boldsymbol{x}|r_k) = e^{-\sum_k E_\theta^k(\boldsymbol{x}|\text{Enc}(r_k))}, \tag{5}$$

which is a new EBM with underlying energy function $E_\theta(\boldsymbol{x}|R) = \sum_k E_\theta(\boldsymbol{x}|\text{Enc}(r_k))$. The overview of the proposed method is shown in Figure 2, where $E_1, \cdots, E_K$ correspond to the K individual relational energy functions $E_\theta^k(\boldsymbol{x}|\text{Enc}(r_k))$.

3.4 Downstream Applications

By learning the probability distribution $p(\boldsymbol{x}|R)$ with corresponding EBM $E_\theta(\boldsymbol{x}|R)$, our model can be applied to solve many downstream applications, such as image generation, editing, and classification, which we detail below and validate in the experiment section.

Table 1: Evaluation of the accuracy of object relations in the generated images or edited images on the CLEVR and iGibson datasets. We compare our method with baselines on three test sets, *i.e. 1R*, *2R*, and *3R* (see text).

Dataset	Model	Image Generation (%)			Image Editing (%)		
		1R Acc	2R Acc	3R Acc	1R Acc	2R Acc	3R Acc
CLEVR	StyleGAN2	10.68	2.46	0.54	10.04	2.10	0.46
	StyleGAN2 (CLIP)	65.98	9.56	1.78	-	-	-
	Ours (CLIP)	94.79	48.42	18.00	95.56	52.78	16.32
	Ours (Learned Embed)	**97.79**	**69.55**	**37.60**	**97.52**	**65.88**	**32.38**
iGibson	StyleGAN2	12.46	2.24	0.60	11.04	2.18	0.84
	StyleGAN2 (CLIP)	49.20	17.06	5.10	-	-	-
	Ours (CLIP)	74.02	43.03	**19.59**	78.12	32.84	12.66
	Ours (Learned Embed)	**78.27**	**45.03**	19.39	**84.16**	**44.10**	**20.76**

Image Generation. We generate images from a relational scene description R by sampling from the probability distribution $p(\boldsymbol{x}|R)$ using Langevin sampling on the energy function $E_\theta(\boldsymbol{x}|R)$ from random noise.

Image Editing. To edit an image $\boldsymbol{x}'$, we utilize the same probability distribution $p(\boldsymbol{x}|R)$ and Langevin sampling on the energy function $E_\theta(\boldsymbol{x}|R)$ but initialize sampling from the image we wish to edit $\boldsymbol{x}'$ instead of random noise.

Relational Scene Understanding. We may further utilize the energy function $E_\theta(\boldsymbol{x}|R)$ as a tool for relational scene understanding by noting that $p(\boldsymbol{x}|R) \propto e^{-E_\theta(\boldsymbol{x}|R)}$. The output values of the energy function can be used as a matching score of the generated/edited image and the given scene relational scene description R.

4 Experiment

We conduct empirical studies to answer the following questions: (1) Can we learn relational models that can generate and edit complex multi-object scenes when given relational scene descriptions with multiple composed scene relations? (2) Can we use our model to generalize to scenes that are never seen in training? (3) Can we understand the set of relations in a scene and infer semantically equivalent descriptions?

To answer these questions, we evaluate the proposed method and baselines on image generation, image editing, and image classification on two main datasets, *i.e.* CLEVR [19] and iGibson [40]. We also test the image generation performance of the proposed model and baselines on a real-world dataset *i.e.* Block [26].

4.1 Datasets

CLEVR. We use $50,000$ pairs of images and relational scene descriptions for training. Each image contains $1 \sim 5$ objects and each object consists of five different attributes, including color, shape, material, size, and its spatial relation to another object in the same image. There are 9 types of colors, 4 types of shapes, 3 types of materials, 3 types of sizes, and 6 types of relations.

iGibson. On the iGibson dataset, we use 30,000 pairs of images and relational scene descriptions for training. Each image contains $1 \sim 3$ objects and each object consists of the same five different types of attributes as the CLEVR dataset. There are 6 types of colors, 5 types of shapes, 4 types of materials, 2 types of sizes, and 4 types of relations. The objects are randomly placed in the scenes.

Blocks. On the real-world Block dataset, a number of 3,000 pairs of images and relational scene descriptions are used for training. Each image contains $1 \sim 4$ objects and each object differs in color. We only consider the "above" and "below" relations as objects are placed vertically in the form of towers.

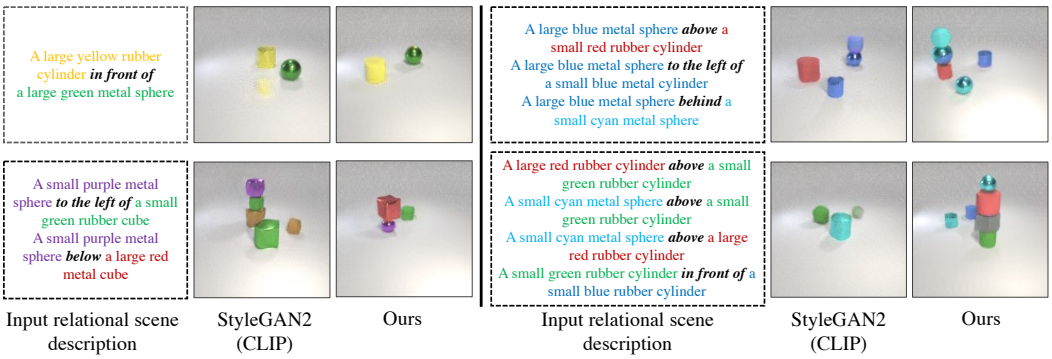

Figure 3: Image generation results on the CLEVR dataset. Image are generated based on $1 \sim 4$ relational descriptions. Note that the models are trained on a single relational description and the composed scene relations (2, 3, and 4 relational descriptions) are outside the training distribution.

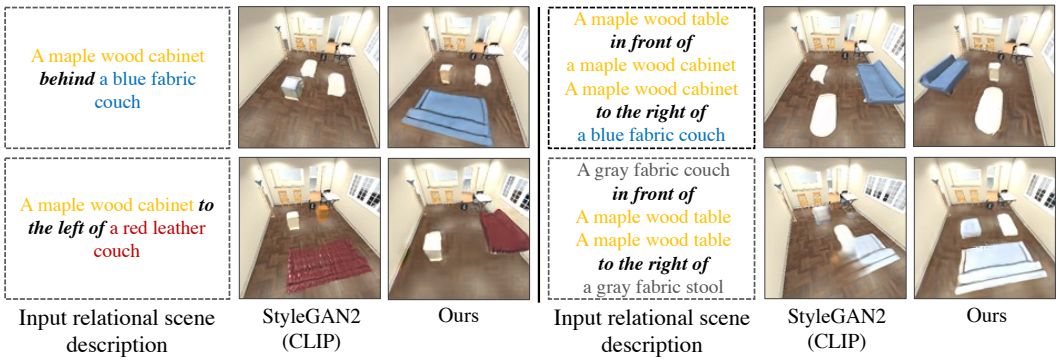

Figure 4: Image generation results on the iGibson dataset. Images are generated based on $1 \sim 2$ relational descriptions. Note that the two composed scene relations are outside the training distribution.

In the training set, each image's relational scene description only contains one scene relation and objects are randomly placed in the scene. We generated three test subsets that contain relational scene descriptions with a different number of scene relations to test the generation ability of the proposed methods and baselines. The *1R* test subset is similar to the training set where each relational scene description contains one scene relation. The *2R* and *3R* test subsets have two and three scene relations in each relational scene description, respectively. Each test set has $5,000$ images with corresponding relational scene descriptions.

4.2 Baselines

We compare our method with two baseline approaches. The first baseline is StyleGAN2 [21], one of the state-of-the-art methods for unconditional image generation. To enable StyleGAN2 to generate images and edit images based on relational scene descriptions, we train an energy-based model to classifier whether given scene contains given objects and relations. Recently, CLIP [34] has achieved a substantial improvement on the text-image retrieval task by learning good text-image feature embeddings on large-scale datasets. Thus we design another baseline, StyleGAN2+CLIP, that combines the capabilities of both approaches. To do this, we encode relational scene descriptions into text embeddings using CLIP and condition StyleGAN2 on the embeddings to generate images. Please see Appendix A.5 for more details of baselines.

4.3 Image Generation Results

Given a relational scene description, *e.g.* "a blue cube on top of a red sphere", we aim to generate images that contain corresponding objects and their relations as described in the given descriptions.

Quantitative comparisons. To evaluate the quality of generated images, we train a binary classifier to predict whether the generated image contains objects and their relations described in the given relational scene description.

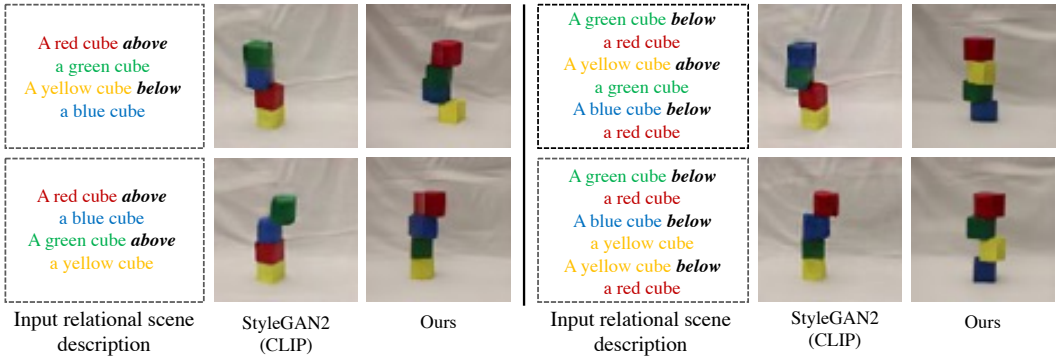

Figure 5: Image generation results on the Block dataset. Image are generated based the relational scene description. Note that the models are trained on a single relational scene description and the composed scene relations are outside the training distribution.

Given a pair of an image and a relational scene description, we first feed the image to several convolutional layers to generate an image feature and then send the relational scene description to an embedding layer followed by several fully connected layers to generate a relational scene feature. The image feature and relational scene feature are combined and then passed through several fully connected and finally a sigmoid function to predict whether the given image matches the relational scene description. The binary classifier is trained on real images from the training dataset. We train a classifier on each dataset and observe classification accuracy on real images to be close to 100%, indicating that the classifier is effective. During testing, we generate an image based on a relational scene description and send the generated image and the relational scene description to the classifier for prediction. For a fair comparison, we use the same classifier to evaluate images generated by all the approaches on each dataset.

The "Image Generation" column in Table 1 shows the classification results of different approaches on the CLEVR and iGibson datasets. On each dataset, we test each method on three test subsets, *i.e.* *1R*, *2R*, *3R*, and report their binary classification accuracies. Both variants of our proposed approach outperform StyleGAN2 and StyleGAN2 (CLIP), indicating that our method can generate images that contain the objects and their relations described in the relational scene descriptions. We find that our approach using the learned embedding, *i.e.* Ours (Learned Embed), achieves better performances on the CLEVR and iGibson datasets than the other variant using the CLIP embedding, *i.e.* Ours (CLIP).

StyleGAN2 and StyleGAN2 (CLIP) can perform well on the *1R* test subset. This is an easier test subset because the models are trained on images with a single scene relation and the models generate images based on a single relational scene description during testing as well. The *2R* and *3R* are more challenging test subsets because the models need to generate images conditioned on relational scene descriptions of multiple scene relations. Our models outperform the baselines by a large margin, indicating the proposed approach has a better generalization ability and can compose multiple relations that are never seen during training.

Human evaluation results. To further evaluate the performance of the proposed method on image generation, we conduct a user study to ask humans to evaluate whether the generated images match the given input scene description. We compare the correctness of the object relations in the generated images and the input language of our proposed model, *i.e.* "Ours (Learned Embed)", and "StyleGAN2 (CLIP)". Given a language description, we generate an image using "Ours (Learned Embed)" and "StyleGAN2 (CLIP)". We shuffle these two generated images and ask the workers to tell which image has better quality and the object relations match the input language description. We tested 300 examples in total, including 100 examples with 1 sentence relational description (1R), 100 examples with 2 sentence relational descriptions (2R), and 100 examples with 3 sentence relational descriptions (3R). There are 32 workers involved in this human experiment.

The workers think that there are 87%, 86%, and 91% of generated examples that "Ours (Learned Embed)" is better than "StyleGAN2 (CLIP)" for *1R*, *2R*, and *3R* respectively. The human experiment shows that our proposed method is better than "StyleGAN2 (CLIP)". The conclusion is coherent with our binary classification evaluation results.

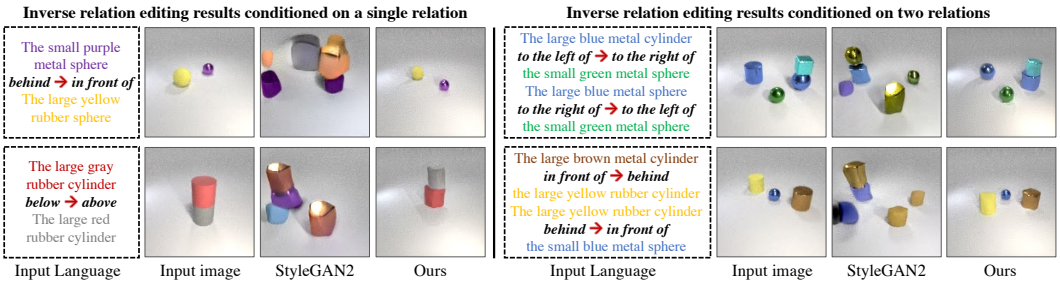

Figure 6: Image editing results on the CLEVR dataset. **Left**: image editing results based on a single relational scene description. **Right**: image editing results based on two composed relational scene descriptions. Note that the composed scene relations in the right part are outside the training distribution and our approach can still edit the images accurately.

Qualitative comparisons. The image generation results on CLEVR, iGibson and Block scenes are shown in Figure 3, 4, and 5, respectively. We show examples of generated images conditioned on relational scene descriptions of different number of scene relations. Our method generates images that are consistent with the relational scene descriptions. Note that both the proposed method and the baselines are trained on images that only contain a relational scene description of a single scene relation describing the visual relationship between two objects in each image. We find that our approach can still generalize well when composing more visual relations. Taking the upper right figure in Figure 3 as an example, a relational scene description of multiple scene relations, *i.e.* "A large blue metal sphere above a small red rubber cylinder. A large blue metal sphere to the left of a small blue metal cylinder $\cdots$", is never seen during training. "StyleGAN2 (CLIP)" generates wrong objects and scene relations that are different from the scene descriptions. In contrast, our method has the ability to generalize to novel relational scenes.

4.4 Image Editing Results

Given an input image, we aim to edit this image based on relational scene descriptions, such as "put a red cube in front of the blue cylinder".

Quantitative comparisons. Similar to the image generation, we use a classifier to predict whether the image after editing contains the objects and their relations described in the relational scene description. For the evaluation on each dataset, we use the same classifier for both image generation and image editing.

The "Image Editing" column in Table 1 shows the classification results of different approaches on the CLEVR and iGibson datasets. Both variants of our proposed approach, *i.e.* "Ours (CLIP)" and "Ours (Learned Embed)" outperform the baselines, *i.e.* "StyleGAN2" and "StyleGAN2 (CLIP)", substantially. The good performance of our approach on the *2R* and *3R* test subsets shows that the proposed method has a good generalization ability to relational scene descriptions that are outside the training distribution. The images after editing based on relational scene descriptions can incorporate the described objects and their relations accurately.

Qualitative comparisons. We show image editing examples in Figure 6. The left part is image editing results conditioned on a single scene relation while the right part is conditioned on two scene relations. We show examples that edit images by inverting individual spatial relations between given two objects. Taking the first image in Figure 6 as an example, "the small purple metal sphere" is behind "the large yellow rubber sphere", after editing, our model can successfully put "the small purple metal sphere" in front of "the large yellow rubber sphere". Even for relational scene descriptions of two scene relations that are never seen during training, our model can edit images so that the selected objects are placed correctly.

4.5 Relational Understanding

We hypothesize the good generation performance of our proposed approach is because our approach understands the underlying set of relations and can distinguish different relational scene descriptions. In this section, we evaluate the relational understanding ability of our proposed method and baselines by comparing their image-to-text retrieval and semantic equivalence results.

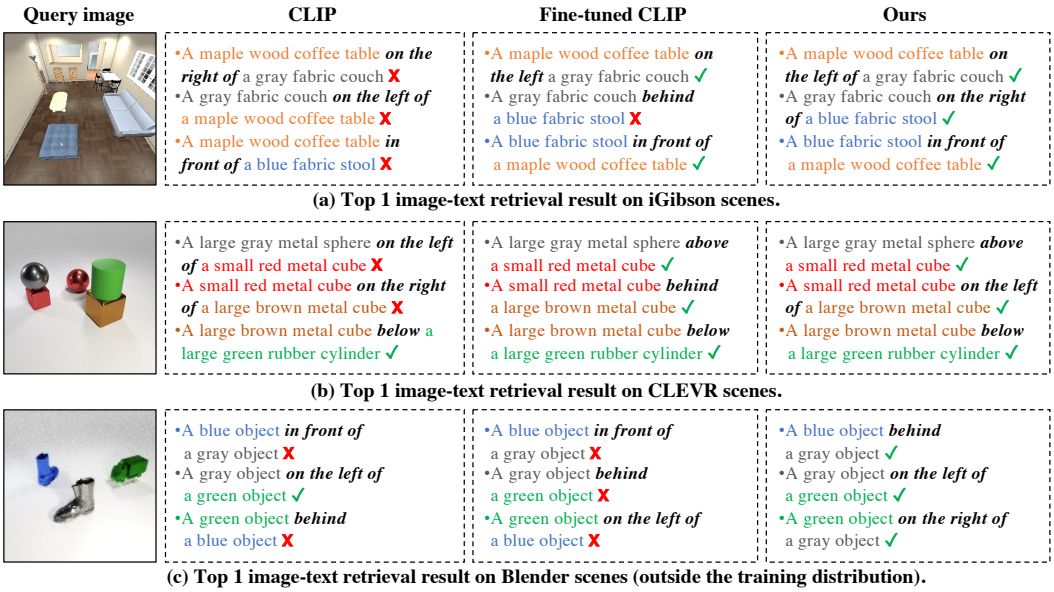

Figure 7: Image-to-text retrieval results. We compare the proposed approach with the pretrained CLIP and fine-tuned CLIP and show their top-1 retrieved relation description based on the given image query.

Image-to-text retrieval. In Figure 7, we evaluate whether our proposed model can understand different relational scene descriptions by image-to-text retrieval. We create a test set that contains 240 pairs of images and relational scene descriptions. Given a query image, we compute the similarity of this image and each relational scene description in the gallery set. The top 1 retrieved relational scene description is shown in Figure 7. We compare our method with two baselines. We use the pre-trained CLIP model and test it on our dataset directly. "Fine-tuned CLIP" means the CLIP model is fine-tuned on our dataset. Even though CLIP has shown good performance on the general image-text retrieval task, we find that it cannot understand spatial relations well, while EBMs can retrieve all the ground truth descriptions.

We also find that our approach generalizes across datasets. In the bottom row of Figure 7, we conduct an additional image-to-text retrieval experiment on the Blender [4] scenes that are never seen during training. Our approach can still find the correct relational scene description for the query image.

Table 2: Quantitative evaluation of **semantic equivalence** on the CLEVR dataset.

Model	**Semantic Equivalence (%)**		
	1R Acc	2R Acc	3R Acc
Classifier	52.82	27.76	14.92
CLIP	37.02	14.40	5.52
CLIP (Fine-tuned)	60.02	35.38	20.9
Ours (CLIP)	70.68	50.48	38.06
Ours (Learned Emb)	**74.76**	**57.76**	**44.86**

Can we understand semantically equivalent relational scene descriptions? Given two relational scene descriptions describing the same image but in different ways, can our approach understand that the descriptions are semantically similar or equivalent? To evaluate this, we create a test subset that contains $5,000$ images and each image has 3 different relational scene descriptions. There are two relational scene descriptions that match the image but describe the image in different ways, such as "a cabinet in front of a couch" and "a couch behind a cabinet". There is one further description that does not match the image. The relative score difference between the two ground truth relational scene descriptions should be smaller than the difference between one ground truth relational scene description and one wrong relational scene description.

We compare our approach with three baselines. For each model, given an image, if the difference between two semantically equivalent relational scene description is smaller than the difference between the semantically different ones, we will classify it as correct. We compute the percentage of correct predictions and show the results in Table 2. Our proposed method outperforms the baselines substantially, indicating that our EBMs can distinguish semantically equivalent relational scene descriptions and semantically nonequivalent relational scene descriptions. In Figure 8, We

Figure 8: Examples of **semantic equivalence** on CLEVR and iGibson scenes. Given an input image, our approach is able to recognize whether the relational scene descriptions are semantically equivalent or not.

further show two examples generated by our approach on the iGibson and CLEVR datasets. The energy difference between the semantically equivalent relational scene description is smaller than the mismatching pairs.

4.6 Zero-shot Generalization Across Datasets

We find that our method can generalize across datasets as shown in the third example in Figure 7. To quantitatively evaluate the generalization ability across datasets of the proposed method, we test the image-to-text retrieval accuracy on the Blender dataset. We render a new Blender dataset using objects including boots, toys, and trucks. Note that our model and baselines are trained on CLEVR and have never seen the Blender scenes during training.

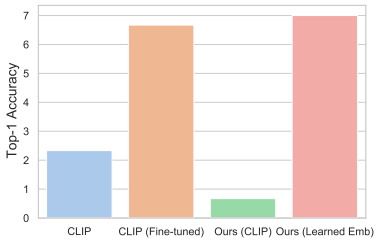

Figure 9: **Zero-shot generalization** on Blender scenes. Our approach with learned embedding outperforms other methods on image-to-text retrieval.

We generate a Blender test set that contains 300 pairs of images and relational scene descriptions. For each image, we do text retrieval on the 300 relational scene descriptions. The top 1 accuracy is shown in Figure 9. We compare our approaches with two baselines, *i.e.* CLIP and CLIP fine-tuned on the CLEVR dataset. We find the CLIP model and our approach using the CLIP embedding perform badly on the Blender dataset. This is because CLIP is not good at modeling relational scene description, as we have shown in Section 4.5. Our approach using the learned embedding outperforms other methods, indicating that our EBMs with a good embedding feature can generalize well even on unseen datasets, such as Blender.

5 Conclusion

In this paper, we demonstrate the potential usage of our model of compositional image generation, editing, and even generalization on unseen datasets based on relational scene descriptions. Our results provide evidence that EBMs are an interesting class of models to study relational understanding.

One limitation of the current approach is that the evaluated datasets are simpler compared to the complex relational descriptions used in the real world. A good direction for future work would be to study how these models scale to complex datasets found in the real world. One particular interest could be measuring the zero-shot generalization capabilities of the proposed model.

Our system, as with all systems based on neural networks, is susceptible to dataset biases. The learned relations will exhibit biases if the training datasets are biased. We must take balanced and fair datasets if we develop our model to solve real-world problems, as otherwise, it could inadvertently worsen existing societal prejudices and biases.

Acknowledgements. Shuang Li is supported by Raytheon BBN Technologies Corp. under the project Symbiant (reg. no. 030256-00001 90113) and Mitsubishi Electric Research Laboratory (MERL) under the project Generative Models For Annotated Video. Yilun Du is supported by NSF graduate research fellowship. This work is in part supported by ONR MURI N00014-18-1-2846 and IBM Thomas J. Watson Research Center CW3031624.

References

[1] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4561–4569, 2019. 3

[2] Amir Bar, Roei Herzig, Xiaolong Wang, Gal Chechik, Trevor Darrell, and Amir Globerson. Compositional video synthesis with action graphs. *arXiv preprint arXiv:2006.15327*, 2020.

[3] Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016. 1, 3

[4] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL http://www.blender.org. 9

[5] N De Cao and Kipf TMGAN. An implicit generative model for small molecular graphs. arxiv preprint 2018. *arXiv preprint arXiv:1805.11973*, 2018. 3

[6] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019. 3

[7] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation and inference with energy based models. *arXiv preprint arXiv:2004.06030*, 2020. 3, 4

[8] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020. 3, 19

[9] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Keep drawing it: Iterative language-based image generation and editing. *arXiv preprint arXiv:1811.09845*, 2, 2018. 14

[10] Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7518–7528, 2020. 3

[11] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021. 1

[12] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019. 3

[13] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471. PMLR, 2015. 2

[14] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2019. 3

[15] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *European Conference on Computer Vision*, pages 210–227. Springer, 2020. 3

[16] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018. URL https://doi.org/10.1162/089976602760128018. 3

[17] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7986–7994, 2018. 2

[18] Tianyu Hua, Hongdong Zheng, Yalong Bai, Wei Zhang, X. Zhang, and Tao Mei. Exploiting relationship for complex-scene image generation. In *AAAI*, 2021. 3

[19] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 5

[20] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018. 2, 3, 15, 17

[21] Tero Karras, M. Aittala, Janne Hellsten, S. Laine, J. Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *ArXiv*, abs/2006.06676, 2020. 6, 17

[22] Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *ArXiv*, abs/1606.03439, 2016. 3

[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. *International journal of computer vision*, 123(1):32–73, 2017. 14

[24] B. Lake, R. Salakhutdinov, and J. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332 – 1338, 2015. 1

[25] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu-Jie Huang. A tutorial on energy-based learning. In G. Bakir, T. Hofman, B. Schölkopf, A. Smola, and B. Taskar, editors, *Predicting Structured Data*. MIT Press, 2006. 3

[26] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. In *International conference on machine learning*, pages 430–438. PMLR, 2016. 5, 14

[27] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. *ArXiv*, abs/1901.06767, 2019. 3

[28] Yikang Li, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. Pastegan: A semi-parametric method to generate image from scene graph. In *NeurIPS*, 2019. 3

[29] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015. 2

[30] Gaurav Mittal, Shubham Agrawal, Anuva Agarwal, Sushant Mehta, and T. Marwah. Interactive image generation using scene graphs. *ArXiv*, abs/1905.03743, 2019. 3

[31] Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. Housegan: Relational generative adversarial networks for graph-constrained house layout generation. In *European Conference on Computer Vision*, pages 162–177. Springer, 2020. 3

[32] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Y. Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. *ArXiv*, abs/1903.12370, 2020. 3

[33] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016. 2

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 1, 4, 6, 17

[35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 1, 2

[36] David Raposo, Adam Santoro, David Barrett, Razvan Pascanu, Timothy Lillicrap, and Peter Battaglia. Discovering objects and their relations from entangled scene representations. *arXiv preprint arXiv:1702.05068*, 2017. 3

[37] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016. 2

[38] Scott Reed, Aäron Oord, Nal Kalchbrenner, Sergio Gómez Colmenarejo, Ziyu Wang, Yutian Chen, Dan Belov, and Nando Freitas. Parallel multiscale autoregressive density estimation. In *International Conference on Machine Learning*, pages 2912–2921. PMLR, 2017.

[39] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. *Advances in neural information processing systems*, 29: 217–225, 2016. 2

[40] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Shyamal Buch, Claudia D'Arpino, Sanjana Srivastava, Lyne P Tchapmi, et al. igibson, a simulation environment for interactive tasks in large realistic scenes. *arXiv preprint arXiv:2012.02924*, 2020. 5

[41] Yunfu Song and Zhijian Ou. Learning neural random fields with inclusive auxiliary generators. *arXiv preprint arXiv:1806.00271*, 2018. 3

[42] Mike Stilman and James J Kuffner. Navigation among movable obstacles: Real-time reasoning in complex environments. *International Journal of Humanoid Robotics*, 2(04):479–503, 2005. 1

[43] Subarna Tripathi, Anahita Bhiwandiwalla, Alexei Bastidas, and Hanlin Tang. Using scene graph context to improve image generation. *arXiv preprint arXiv:1901.03762*, 2019. 3

[44] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644. PMLR, 2016. 3

[45] Wei Yu, Wenxin Chen, Songhenh Yin, Steve Easterbrook, and Animesh Garg. Concept grounding with modular action-capsules in semantic video prediction. *arXiv preprint arXiv:2011.11201*, 2020. 2

[46] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 2

[47] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1947–1962, 2019. 2

Appendix

In this appendix, we first present qualitative and quantitative comparisons with additional baselines and ablations in Section A. We then show the experiment results on the more complex real world datasets in Section B. We then show additional dataset details in Section C. More details of our proposed approach and baselines are shown in Section D and Section E, respectively. The inference details of different approaches are introduced in Section F. We provide the model architecture details of various approaches and their training details in Section G and Section H. Finally, we show the algorithm in Section I.

A Additional Results

Quantitative results. We provide additional quantitative analyiss of our approach. Firstly, we investigate performance utilizing the graph-based relational similarity metric proposed in [9]. Relational similarity measures how many of the ground truth relations are present in the generated images. To compute the relational similarity scores, we use off-the-shelf object detector and localizer from [9] to determine the color, shape and positions of image objects that are used to estimate corresponding scene graphs. Finally, constructed scene graphs are used to compute the graph-based relational similarity score shown in Table 3.

We provide further quantitative comparisons against additional baselines in Table 4, where baselines are described in Section E. Table 4 shows that our approach achieves the highest relation classification score among all methods.

Qualitative results. We next demonstrate more qualitative results of image generation and associated comparisons in Figure 10, and Figure 11. We show that our approach can generate images with correct relations, and even generalize to out of distribution relational scene descriptions.

Table 3: Quantitative evaluation of **relational similarity** of objects on CLEVR dataset. It measures a graph similarity for constructed scene graphs of generated and real images.

Model	**Relational Similarity (%)**		
	1R Acc	2R Acc	3R Acc
StyleGAN2	22.37	19.75	17.13
StyleGAN2 (CLIP)	37.50	28.62	28.75
Ours (Learned Emb)	**50.77**	**36.87**	**42.50**

More details We describe additional baselines comparisons in Section D and E. In main paper Section 3.2, we mention that encoding the entire sentence such as "a large blue rubber cube on the left of a small red metal cube" does not perform well when using EBMs. We show that utilizing CLIP to encode the whole sentence of relation scene description for an EBM performs poorly in Table 4 and the associated qualitative figures under the label "Ours (CLIP-all)".

B Results on the Real World Datasets

In terms of image generation results on real scenes, we train and evaluate our model on two real-world datasets, one real-world block dataset and the visual genome dataset.

The real-world block dataset is from [26] and we train our model using the object relations, e.g. "above" and "below". We show the images generated conditioned on two relational descriptions, and three relational descriptions in Figure 12.

For the visual genome dataset [23], we train our models on a subset that consists of common objects and relations for computational efficiency. As shown in Figure 13, we find that the CLIP text encoder performs better, as it has seen large-scale image-text pairs that cover a wide range of relations, attributes and objects.

Our approach is able to generate images (objects and their relations) matching the given language descriptions on the real-world block dataset and the visual genome dataset. The quality of generated images on the real-world block dataset is great. However, the quality of the visual genome is a bit worse. We believe, given more time and more computation resources, we can further improve the generation quality.

Table 4: Evaluation of the accuracy of relational scene descriptions in the generated images on the CLEVR and iGibson datasets. We compare our method with baselines on three test setsets, *i.e. 1R*, *2R*, and *3R* (see text). In the main Table 1, we had two baselines – StyleGAN2 and StyleGAN2 (CLIP). Here we add the other 3 baselines, including Scene Graph GAN [20], Ours (CLIP-all) that indicates the relational scene descriptions are fully encoded by the CLIP's pretrained encoder, and StyleGAN2 (CLIP) (Multi-Relations).

Dataset	**Model**	**Image Generation (%)**		
		1R Acc	2R Acc	3R Acc
CLEVR	StyleGAN2	10.68	2.46	0.54
	StyleGAN2 (CLIP)	65.98	9.56	1.78
	StyleGAN2 (CLIP) (Multi-Relations)	66.62	9.60	1.68
	Scene Graph GAN	83.72	14.18	4.48
	Ours (CLIP-all)	4.75	0.24	0.00
	Ours (CLIP)	94.79	48.42	18.00
	Ours (Learned Embed)	**97.79**	**69.55**	**37.60**
iGibson	StyleGAN2	12.46	2.24	0.60
	StyleGAN2 (CLIP)	49.20	17.06	5.10
	StyleGAN2 (CLIP) (Multi-Relations)	36.94	13.42	6.86
	Scene Graph GAN	54.64	0.02	0.00
	Ours (CLIP-all)	34.25	8.05	3.47
	Ours (CLIP)	74.02	43.04	**19.59**
	Ours (Learned Embed)	**78.28**	**45.03**	19.39

C Datasets Details

CLEVR. Each image contains $1 \sim 5$ objects and each object consists of five different attributes, including color, shape, material, size, and its relation to another object in the same image. There are 9 types of colors, 4 types of shapes, 3 types of materials, 3 types of sizes, and 6 types of relations.

iGibson. Each image contains $1 \sim 3$ objects and each object consists of the same five different types of attributes as the CLEVR dataset. There are 6 types of colors, 5 types of shapes, 4 types of materials, 2 types of sizes, and 4 types of relations. The objects are randomly placed in the scenes.

Blocks. In the real-world block tower dataset, each image contains $1 \sim 4$ cubes and each cube only differs in color. Objects in the images are placed vertically in the form of towers.

Training datasets for CLEVR, iGibson and block include 50,000, 30,000 and 3,000 images respectively, while we use datasets of 5,000 images for testing for both CLEVR and iGibson. Each input data is a triplet $t = (r'_i, o^1_i, o^2_i)$, corresponding to the *relation*, *object 1* and *object 2*. For generalization testing dataset, we use three .blend files, such as *trucks*, *toys*, and *boots* to generate 5,000 images where each image has a total of 3 objects. All objects are placed in the scenes randomly.

Dataset Splits. To ensure that training datasets and testing datasets do not overlap, we use the following heuristic: we find all possibilities of input triplets considering the relative order of o^1_i and o^2_i. so $(r'_i, o^1_i, o^2_i) \neq (r'_i, o^2_i, o^1_i)$. Then we add each triplet t into the training dataset $\mathcal{S}$ while maintaining two sets $\mathcal{R}$ and $\mathcal{O}$, where $\mathcal{O} = \{o_1, \dots o_i\}$ and $\mathcal{R} = \{r_1, \dots r_i\}$, represented as previously seen relations and objects. When each triplet is added into $\mathcal{S}$, we update both $\mathcal{R}$ and $\mathcal{O}$. Once we reach a certain number of triplets in $\mathcal{S}$, and all possible objects and relations are all seen, then we use these sets to filter out seen triplets from the rest of data available. For example, we add (r'_i, o^2_i, o^1_i) into the test set if the training dataset has (r'_i, o^1_i, o^2_i) but not (r'_i, o^2_i, o^1_i), and (r'_i, o^1_i, o^2_i) has never been seen as a triplet but each piece of information has been added into the training dataset individually. This heuristic guarantees that each piece of information r_i, o^1_i or o^2_i is in the training dataset at least partially and we test on triplets $t \in \mathcal{T} \setminus \mathcal{S}$, where $\mathcal{T}$ is the universal set of triplets given the dataset. Therefore, our triplets for testing will not appear in the training set $\mathcal{S}$.

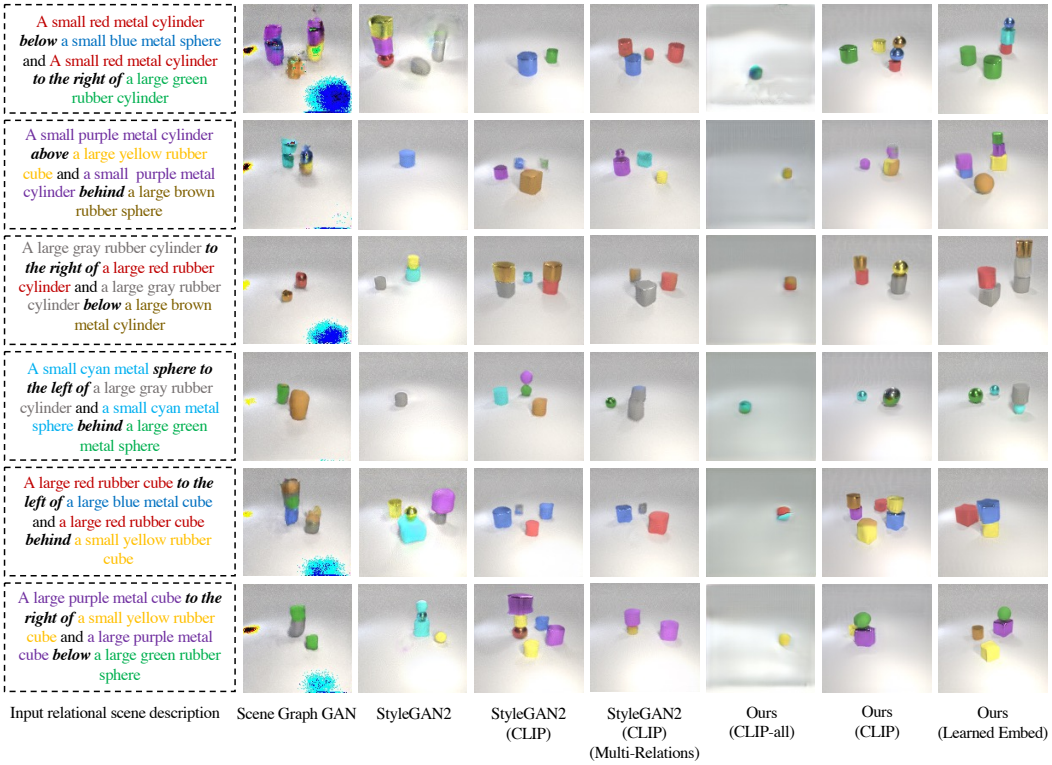

Figure 10: Examples of image generation on CLEVR scenes based on two scene relation. We show more generated images based on two relational scene descriptions. Note that the composed scene relational descriptions are outside the training distribution and our main approach "Ours (Learned Embed)" can still generate correct images, but "Ours (CLIP-all)" fails completely.

D Details of Our Approaches

Ours (CLIP). In our energy-based model setting, we only encode object descriptions using CLIP and initialize a relation embedding to encode relations separately. For example, we concatenate embeddings of "a large blue rubber cube" and "a small red metal cube" encoded by CLIP and "left" encoded by the relation embedding to represent the relational scene representation. We also concatenate the embedding with learned object position embedding so that "left" indicates the relation from the first object o_i^1 to the second object o_i^2 where we use 0, 1, and 2 to represent the first object, the second object and none object respectively where none object is used when the image only contains one single object.

Ours (CLIP-all). In this setting, we use CLIP's text encoder to encode every word in the relational scene descriptions, so relation embedding is not used in this model. We show that this model underperforms both qualitatively and quantitatively in Table 4 and Figures 10, 11 and 12.

Ours (Learned Embed). In the setting of learned embedding, we have a total of 6 different embedding layers for color, size, material, shape, relation and position. A total of 8 attribute embeddings (4 for each object), 2 position embeddings and 1 relation embedding is concatenated to represent the scene. The embedding dimensions for color, size, material, shape and position are 64, and the relation embedding dimension is 512. We have one fully connected layer to map embedding concatenation of (color, size, material, shape, position) to a dimension of 512. Finally, we concatenate the final embeddings of 2 objects and 1 relation, which has an output dimension of 1536.

E Details of Baselines

In this section, we explain the details of the baselines we compared with, such as training details (hyperparameters, learning rates, etc.) for image generation and image editing.

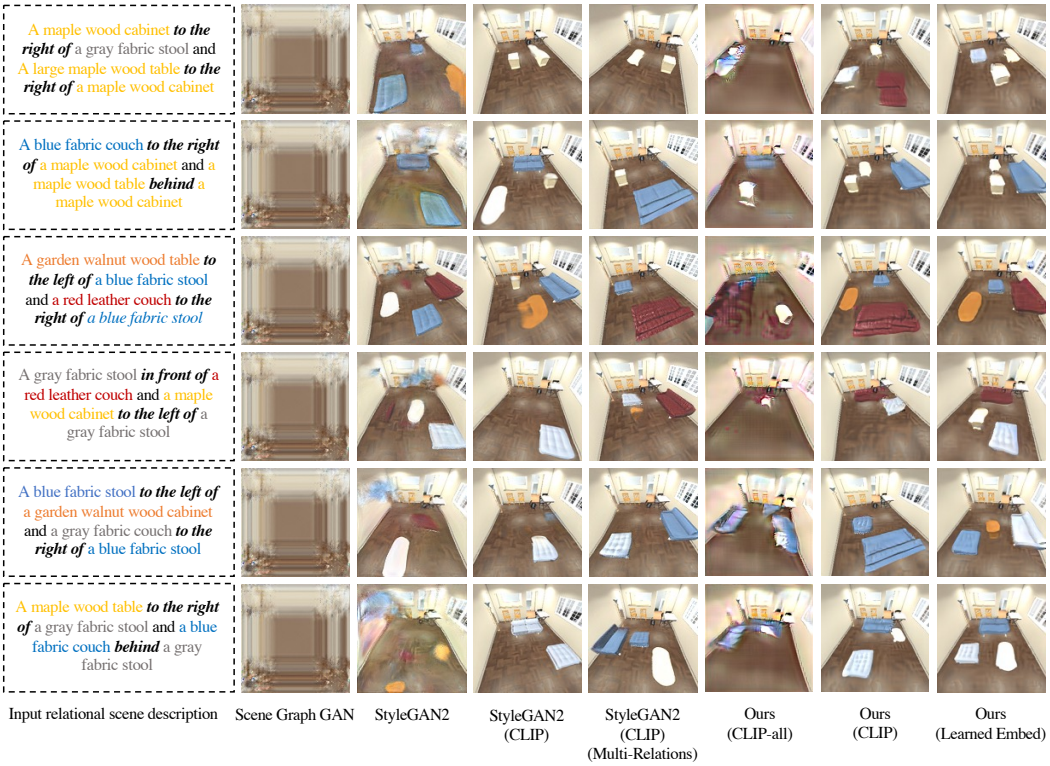

Figure 11: Examples of image generation on iGibson scenes based on two scene relations. Our main approach outperforms the rest of the methods by generating correct images. Note that given relational scene descriptions are out of the training distribution.

StyleGAN2. In Section 4.2 of the main paper, we mentioned unconditional StyleGAN2 [21] as a baseline. We train the unconditional StyleGAN2 using the default setting provided by the original paper [21]. The unconditional StyleGAN2 is guided by a classifier C that is trained to maximize the likelihood for the ground truth triplet data in both image generation and editing. More details are provided in Inference Section F.1.

StyleGAN2 (CLIP). We use an existing CLIP's text encoder [34] to encode relational scene descriptions with a single scene relation and use the text embedding as the label. For example, "A large blue cube on the left of a small red cube" is used in training but "A large blue cube on the left of a small red cube, and a large blue cube above a large yellow cylinder" is not used since it consists of two scene relations: "left" and "above".

StyleGAN2 (CLIP) (Multi-Relations). This model is a variant of StyleGAN2 (CLIP), but it has more scene relations given during training. In StyleGAN2 (CLIP), we only use relational scene descriptions with a single scene relation as the label, but we train this variant with at most three scene relations for each relational scene description because the CLIP's context length is 77, which can only fit relational scene descriptions with at most three scene relations.

Scene Graph GAN. We further apply models from [20] and utilize scene graphs as input to train a conditional StyleGAN2 for each dataset. However, the original setup requires bounding boxes, so we set the input bounding box to be the whole image frame, and our input scene graphs only consist of two objects and one relation to be consistent.

F Inference Details

F.1 Unconditional StyleGAN2

Note that we use each unconditional StyleGAN2 for both image generation and image editing.

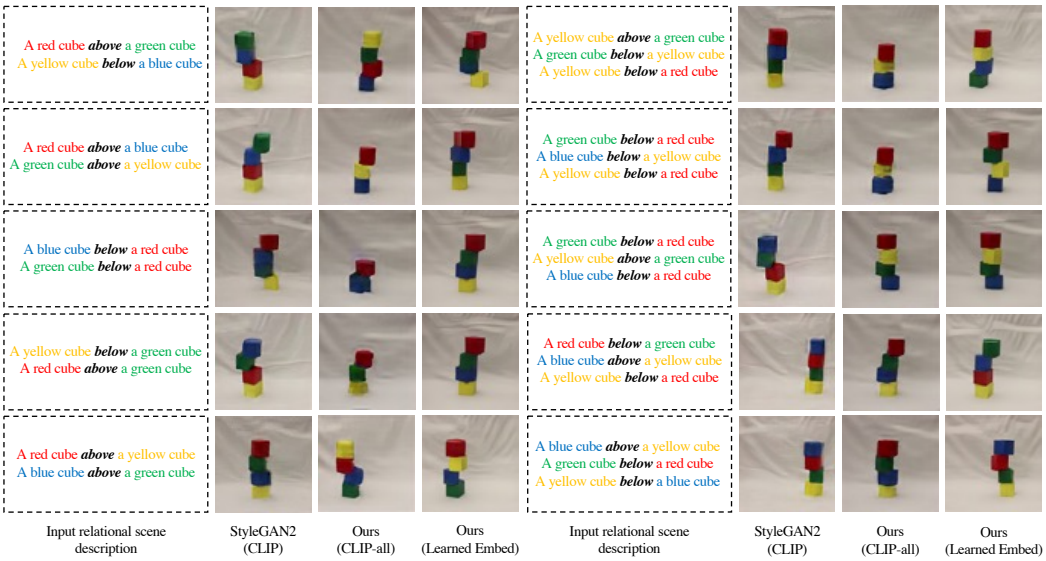

Figure 12: Examples of image generation on real-world block scenes based on two and three scene relations. Given relational scene descriptions are out of the training distribution.

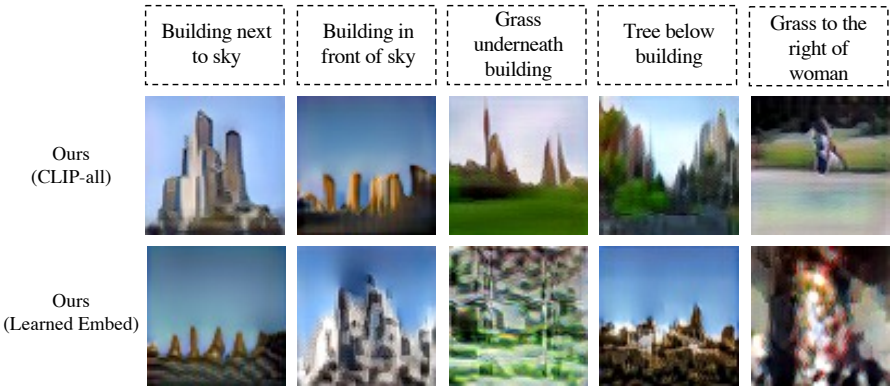

Figure 13: Examples of image generation on a subset of visual genome scenes based on a single scene relation. We compare our main approach Ours (Learned Emb) and Ours (CLIP-all) to visualize which type of embedding is more suitable for real-world scenes.

Image Generation. Note that for a StyleGAN2 model, we may optimize and reconstruct a particular image in the latent space by utilizing a latent optimizer to optimize the L1 loss between the input image and the image generated by the latent. Thus, given a relational scene description, we optimize on the final projected latent to generate images such that it maximizes the likelihood of a classifier predicting this relational scene description.

Image Editing. We utilize a similar strategy to that of image generation but first initialize our starting latent to one that reconstructs the input image. We then similarly optimize the latent to maximize the likelihood of a classifier predicting this relational descriptions.

F.2 Conditional StyleGAN2

We use conditional GANs (*i.e.* different variants of conditional StyleGAN2 and Scene Graph GAN) for image generation only in our experiments.

Image Generation. To enable image generation after training, we encode any given relational scene description to any required format and we condition on the embedding to generate the corresponding

Table 5: Multi-scale model architecture for EBMs proposed in [8].

3x3 Conv2d 128
CondResBlock 128
CondResBlock Down 128
CondResBlock 128
CondResBlock Down 256
Self-Attention 256
CondResBlock 256
CondResBlock Down 256
CondResBlock 512
CondResBlock Down 512
Global Mean Pooling
Dense $\rightarrow$ 1

3x3 Conv2d 128
CondResBlock 128
CondResBlock Down 128
CondResBlock 128
CondResBlock Down 128
Self-Attention 256
CondResBlock 256
CondResBlock Down 256
Global Mean Pooling
Dense $\rightarrow$ 1

3x3 Conv2d 128
CondResBlock 128
CondResBlock Down 128
Self-Attention 128
CondResBlock 128
CondResBlock Down 128
Global Mean Pooling
Dense $\rightarrow$ 1

images. For example, we use CLIP's text encoder to encode text in StyleGAN2 (CLIP), and StyleGAN2 (CLIP) (Multi-Relations) while scene graphs are used for Scene Graph GAN.

F.3 Ours

In this section, we introduce the details of image generation and image editing using our EBMs.

Image Generation. After parsing a relational scene description into corresponding input labels, we draw a random noise, and run 10 alternating series of data augmentation and Langevin sampling to get an initial input. Finally, we do a final run of Langevin sampling that consists of 80 steps to generate the final image. The step size used in image generation depends on the number of scene relations in a given relational scene description. Our step size equals the training step size divided by the number of relations.

Image Editing. After splitting a relational scene description into corresponding input labels, we simply run Langevin sampling on the input image with 80 steps to generate the final image. The step size used in image generation depends on the number of scene relations in a given relational scene description. Our step size equals the training step size divided by the number of scene relations.

G Model Architecture Details

We use the multi-scale model architecture proposed in [8] for all datasets, as shown in Table 5. The final energy is the sum of energies output by different sizes of models. The only difference between different variants is how we encode the label. The label dimension affects the label size used in CondResBlock. We use a size of 1536 for Ours (Learned Embed), 512 for Ours (CLIP) since the final dimension of learned embedding is 1536 and output dimension of CLIP's text encoder is 512.

H Training Details

StyleGAN2. We train an unconditional StyleGAN2 and a ResNet-18 classifier separately given the training dataset. Training the StyleGAN2 takes a total of 2 days on a single Tesla 32GB GPU for each dataset, while the classifier trains on training datasets with labels for 2 hours. The classifier achieves around 40% accuracy. Note that each image includes $1 \sim 5$ objects for CLEVR and $1 \sim 3$ objects for iGibson, so the classifier can not achieve a high accuracy due to the number of objects in each image.

StyleGAN2 (CLIP). We train StyleGAN2 (CLIP) on each training dataset for 2 days on a single Tesla 32GB GPU. We firstly precompute text features for relational scene descriptions and save them

into *dataset.json* that is, the label file for conditional StyleGAN2 training. The training configuration, *stylegan2*, is used to train our conditional StyleGAN2 (CLIP) models.

StyleGAN2 (CLIP) (Multi-Relations). We train the model on each training dataset for 2 days on a single Tesla 32GB GPU. We firstly precompute text features for relational scene descriptions and save them into *dataset.json* that is, the label file for conditional StyleGAN2 training. The training configuration, *stylegan2*, is used to train our conditional StyleGAN2 (CLIP) models.

Scene Graph GAN. We train the model on each training dataset for 2 days on a single Tesla 32GB GPU. We use the image frame as the bounding boxes since our input doesn't contain any bounding boxes for any objects in given images. The default setting is used for training on both datasets. We train the scene graph GAN for 300,000 iteration on CLEVR while 200,000 iterations on iGibson dataset. In our experiment, we use Scene Graph GAN for image generation.

Ours. We use Adam optimizer with learning rates of 0.0001 and 0.0002 for CLEVR and iGibson respectively for all variants of our approach. For MCMC sampling, a step size of 300 is used on CLEVR and a step size of 750 is used on iGibson. We train the model on each dataset for 3 days on a single Tesla 32GB GPU.

I Algorithms

We provide snippets for a total of four algorithms, including training, image generation, image editing and image-to-text retrieval in Algorithm 1, 2, 3 and 4 respectively.

Algorithm 1 Conditional EBM training algorithm

Input: data dist $p_D(\boldsymbol{x})$, relational scene descriptions $R_D(\boldsymbol{r})$, step size λ, number of steps K, data augmentation $D(\cdot)$, stop gradient operator $\Omega(\cdot)$, EBM $E_\theta(\cdot)$, Encoder $\text{Enc}(\cdot)$
$\mathcal{B} \leftarrow \varnothing$
while not converged **do**
  $\boldsymbol{x}_i^+ \sim p_D$
  $R_i \sim R_D$
  $\tilde{\boldsymbol{x}}_i^0 \sim \mathcal{B}$ with 99% probability and $\mathcal{U}$ otherwise
  $X \sim \mathcal{B}$ for nearest neighbor entropy calculation

  ▷ *Split a relational scene description into individual scene relations:*
  $\{\boldsymbol{r}_1, \dots \boldsymbol{r}_m\} \leftarrow R_i$

  ▷ *Apply data augmentation to sample:*
  $\tilde{\boldsymbol{x}}_i^0 = D(\tilde{\boldsymbol{x}}_i^0)$

  ▷ *Generate sample using Langevin dynamics:*
  for sample step $k = 1$ to K **do**
    $\tilde{\boldsymbol{x}}_i^{k-1} = \Omega(\tilde{\boldsymbol{x}}_i^{k-1})$
    $\tilde{\boldsymbol{x}}^k \leftarrow \tilde{\boldsymbol{x}}^{k-1} - \nabla_{\boldsymbol{x}} \sum_{j=1}^m E_\theta(\tilde{\boldsymbol{x}}^{k-1} \mid \text{Enc}(\boldsymbol{r}_j)) + \omega, \quad \omega \sim \mathcal{N}(0, \sigma)$
  end for

  ▷ *Generate two variants of $\boldsymbol{x}^-$ with and without gradient propagation:*
  $\boldsymbol{x}_i^- = \Omega(\tilde{\boldsymbol{x}}_i^k)$
  $\hat{\boldsymbol{x}}_i^- = \tilde{\boldsymbol{x}}_i^k$

  ▷ *Optimize objective $\mathcal{L}_{CD} + \mathcal{L}_{KL}$ wrt θ:*
  $\mathcal{L}_{\text{CD}} = \frac{1}{N} \sum_i \sum_{j=1}^m (E_\theta(\boldsymbol{x}_i^+ \mid \text{Enc}(\boldsymbol{r}_j)) - E_\theta(\boldsymbol{x}_i^- \mid \text{Enc}(\boldsymbol{r}_j))$
  $\mathcal{L}_{\text{KL}} = \sum_{j=1}^m E_{\Omega(\theta)}(\hat{\boldsymbol{x}}_i^- \mid \text{Enc}(\boldsymbol{r}_j)) - \log(NN(\hat{\boldsymbol{x}}_i^-, X)$

  ▷ *Optimize objective $\mathcal{L}_{CD} + \mathcal{L}_{KL}$ wrt θ:*
  $\Delta\theta \leftarrow \nabla_\theta(\mathcal{L}_{\text{CD}} + \mathcal{L}_{\text{KL}})$
  Update θ based on $\Delta\theta$ using Adam optimizer

  ▷ *Update replay buffer $\mathcal{B}$*
  $\mathcal{B} \leftarrow \mathcal{B} \cup \tilde{\boldsymbol{x}}_i^-$
end while

Algorithm 2 Image generation during testing

Input: Relational scene description R, number of data augmentation applications N, step size λ, number of steps K, data augmentation $D(\cdot)$, EBM $E_\theta(\cdot)$, Encoder Enc$(\cdot)$
$\tilde{\boldsymbol{x}}^0 \sim \mathcal{U}$

▷ *Split a relational scene description into individual scene relations:*
$\{\boldsymbol{r}_1, \dots \boldsymbol{r}_m\} \leftarrow R$

▷ *Generate samples through N iterative steps of data augmentation/Langevin dynamics:*
for sample step $n = 1$ to N **do**
  ▷ *Apply data augmentation to samples:*
  $\tilde{\boldsymbol{x}}^0 = D(\tilde{\boldsymbol{x}}_i^0)$

  ▷ *Run K steps of Langevin dynamics:*
  for sample step $k = 1$ to K **do**
    $\tilde{\boldsymbol{x}}^k \leftarrow \tilde{\boldsymbol{x}}^{k-1} - \sum_{i=1}^{n} \nabla_{\boldsymbol{x}} E_\theta(\tilde{\boldsymbol{x}}^{k-1} \mid \text{Enc}(\boldsymbol{r}_i)) + \omega, \quad \omega \sim \mathcal{N}(0, \sigma)$
  end for

  ▷ *Iteratively refine samples:*
  $\tilde{\boldsymbol{x}}^0 = \tilde{\boldsymbol{x}}^k$
end for

▷ *Final output:*
$\boldsymbol{x} = \tilde{\boldsymbol{x}}^0$

Algorithm 3 Image editing during testing

Input: input image $\tilde{\boldsymbol{x}}^0$, relational scene description R, number of data augmentation applications N, step size λ, number of steps K, data augmentation $D(\cdot)$, EBM $E_\theta(\cdot)$ Encoder Enc$(\cdot)$
▷ *Split a relational scene description into individual scene relations:*
$\{\boldsymbol{r}_1, \dots \boldsymbol{r}_m\} \leftarrow R$

▷ *Generate samples through N iterative steps of data augmentation/Langevin dynamics:*
for sample step $n = 1$ to N **do**
  ▷ *Apply data augmentation to samples:*
  $\tilde{\boldsymbol{x}}^0 = D(\tilde{\boldsymbol{x}}_i^0)$

  ▷ *Run K steps of Langevin dynamics:*
  for sample step $k = 1$ to K **do**
    $\tilde{\boldsymbol{x}}^k \leftarrow \tilde{\boldsymbol{x}}^{k-1} - \sum_{i=1}^{n} \nabla_{\boldsymbol{x}} E_\theta(\tilde{\boldsymbol{x}}^{k-1} \mid \text{Enc}(\boldsymbol{r}_i)) + \omega, \quad \omega \sim \mathcal{N}(0, \sigma)$
  end for

  ▷ *Iteratively refine samples:*
  $\tilde{\boldsymbol{x}}^0 = \tilde{\boldsymbol{x}}^k$
end for

▷ *Final output:*
$\boldsymbol{x} = \tilde{\boldsymbol{x}}^0$

Algorithm 4 Image-to-text retrieval during testing

Input: input image $\boldsymbol{x}$, relational scene descriptions $\{R_1, \dots, R_n\}$, EBM $E_\theta(\cdot)$, Encoder Enc$(\cdot)$, output energy list $\mathcal{O}$, caption prediction $\mathcal{C}$
$\mathcal{O} \leftarrow []$

▷ *Generate image-caption matching energies iteratively*
for number of scene relations descriptions $i = 1$ to n **do**
  ▷ *Split a relational scene description into individual scene relations:*
  $\{\boldsymbol{r}_1, \dots \boldsymbol{r}_m\} \leftarrow R_i$
  $\boldsymbol{e}_i = \sum_{j=1}^{m} E_\theta(\boldsymbol{x} \mid \text{Enc}(\boldsymbol{r}_j))$

  ▷ *output energy list $\mathcal{O}$*
  $\mathcal{O}.append(\boldsymbol{e}_i)$
end for

▷ *Final output:*
$\mathcal{C} = \arg\min \mathcal{O}$