Learning Canonical Representations for Scene Graph to Image Generation

Roei Herzig[1⋆†] Amir Bar[5⋆] Huijuan Xu[2] Gal Chechik[3,4] Trevor Darrell[2] Amir Globerson[1]
[1]Tel Aviv University, [2]UC Berkeley, [3]Bar Ilan University, [4]NVIDIA research, [5]Zebra Medical Vision

Abstract

Generating realistic images of complex visual scenes becomes very challenging when one wishes to control the structure of the generated images. Previous approaches showed that scenes with few entities can be controlled using scene graphs, but this approach struggles as the complexity of the graph (number of objects and edges) increases. Moreover, current approaches fail to generalize conditioned on the number of objects or when given different input graphs which are semantic equivalent. In this work, we propose a novel approach to mitigate these issues. We present a novel model which can inherently learn canonical graph representations, thus ensuring that semantically similar scene graphs will result in similar predictions. In addition, the proposed model can better capture object representation independently of the number of objects in the graph. We show improved performance of the model on three different benchmarks: Visual Genome, COCO and CLEVR.

1. Introduction

Generating realistic images is a key task in current computer vision research. Recently, a series of methods were presented for creating realistic-looking, high-resolution images of objects and faces (e.g. [18, 33] and many others).

In these generative models, a key challenge remains: how can one control the content of the generated image at multiple levels, to generate images that have specific desired composition and attributes. Controlling the spatial layout can be particularly challenging when generating rich visual scenes that include several people interacting with objects.

One natural way of describing image composition is via the structure of a *Scene Graph* (SG), which contains a set of objects as nodes and their attributes and relations as edges. Indeed, several studies addressed generating images from SGs [15, 1]. Unfortunately, the quality of images generated from SGs still lags far behind that of generating single objects or faces.

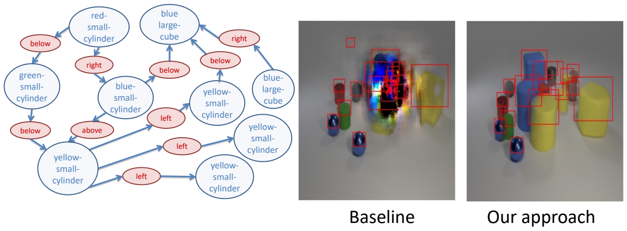


Figure 1: **Generation of packed scenes.** Our method achieves better performance on packed scenes than previous methods. **Left:** A partial input scene graph. **Middle:** Generation using [15]. **Right:** Generation using our proposed method.

Here, we show that current models fail to capture certain logical invariances in SGs, and that this failure hinders learning effective generation models. We further show that taking into account these SG invariances improves the quality of generated layouts and downstream generated images.

A key step in SG to image generation is the so-called SG-to-layout step. In this step, the SG (which does not contain bounding box coordinates) is used to generate a scene layout that contains the bounding box coordinates for all the objects in the SG. The transformation relies on geometric properties specified in the SG such as "$(A, \text{right of}, B)$" and "(C, inside, D)".

Since SGs are typically generated by humans, they usually do not contain *all* correct relations in the data. For example, in an SG with relation $(A, \text{right of}, B)$ it is always true that $(B, \text{left of}, A)$. However, typically only one of these relations will appear. On the other hand, we would like an SG where both relations appear to result in the same layout and image. As we show here, this desired property is in fact often violated by existing models, resulting in low-quality generated images (see Fig. 1). In particular, this means that such models break down for larger SGs.

Here we present an approach to overcome the above difficulty. We first formalize the problem as being invariant to certain logical equivalences (i.e., all equivalent SGs should generate the same image). Next, we propose to replace any SG with a canonical form such that all logically equivalent graphs would be represented by the same canonical SG, and this canonical SG would be the one used in the layout generation step. This approach would, by definition, result in

[⋆] Equal Contribution.
[†] Work done while at the University of Berkeley California.

the same output for all graphs in the same equivalence class. Here we offer a practical approach to learning such a canonicalization process, that does not use any prior knowledge about the relations (e.g., it does not know that "right of" is a transitive relation). We show how to integrate the resulting canonical SGs within a Scene-graph-to-image generation model, and how to learn it from data.

Our contributions are as follows:

- We show that current SG-to-layout models are not invariant to certain logical equivalences corresponding to semantically identical graphs. This in turn results in degraded quality of generated images.
- We propose a model that employs a canonical representation of SGs, and thus has stronger invariance properties than existing methods. Furthermore, the model learns which relations satisfy which properties, rather than receive this information explicitly.
- We show that the number of layers used by our model is roughly independent of the number of objects in the image, whereas previous approaches empirically require depth that grows with the number of objects.
- Empirically, our approach improves layout generation on COCO and VG when compared to state of the art baselines. Furthermore, the resulting generated images also have improved generation scores.

2. Related Work

Early image generation approaches used autoregressive networks [32, 47] to model the pixel conditional distribution. Recently, generative adversarial networks (GAN) [9] and variational autoencoders (VAE) [21] became the models of choice for this task. Specifically for GANs, a series of works [4, 36, 40, 60, 33, 18] were proposed for generating sharper and more realistic images.

Conditional image generation controls the content in a generated by conditioning on certain inputs to the generator content. As a few examples, conditioning inputs may include class labels [6, 31, 28], source images [14, 45, 13, 25, 62, 63], model interventions [2], and text [12, 37, 57, 42, 54, 34]. Generating images from text has been a particularly active field. Early approaches [38, 37, 51] directly encoded a full sentence into a vector that was provided as a conditioning input to the generator. Recent models [12, 61] incorporate an intermediate structured representation, like a layout or a skeleton, to control the coarse structure of the generated image. A two-stage pipeline that generates such layout as an intermediate step has been shown effective.

Several studies focused on generating images directly from structured representations, such as semantic segmentation masks [5], layout [58], and SGs [15]. Layout and SGs are more compact structured representations as compared to whole scene segmentation masks. While layout [58] provides the spatial information of objects in the scene, SGs [15] provide richer and sometimes more abstract information about object attributes and relations. Other advantages of SGs is that they are closely related to the semantics of the image as perceived by humans, and therefore editing a SG correspond to clear changes in semantics, allowing potentially powerful generation interfaces. SGs have also been used in image retrieval [17, 41], relationship modeling [35, 22], image captioning [50] and even action recognition [10]. Several works have addressed the problem of generating SGs from text [41, 46], standalone objects [49] and images [11].

As we note above, current SG to image models [15, 7, 29, 1] show degraded performance on SGs with many objects. To mitigate this, the authors in [1] have utilized stronger supervision in the form of a coarse grid, where attributes of location and size are specified for each object. The focus of this paper is to alleviate this difficulty by directly modeling some of the invariances in the SG representation.

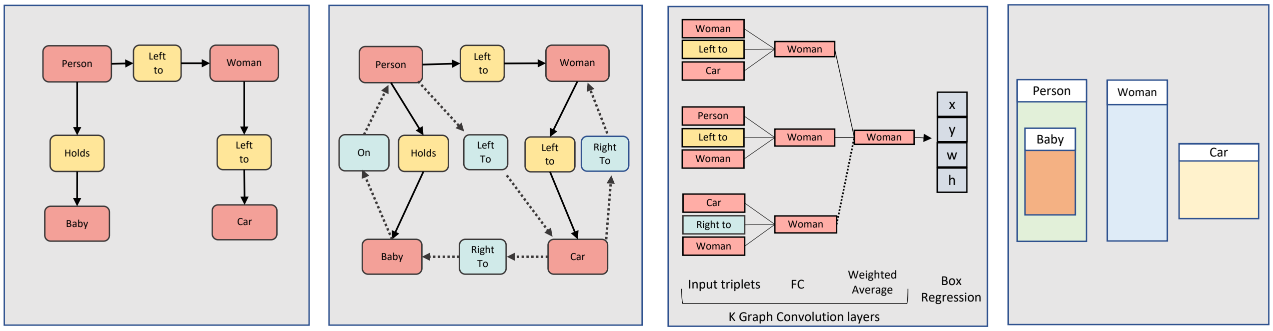
Finally, the topic of invariance in deep architectures has also attracted considerable interest, but mostly in the context of certain permutation invariances [11, 55, 27]. Our approach focuses on a more complex notions of invariance, and addresses them via canonicalization.

3. Scene Graph Canonicalization

A scene graph is a formalism for describing a set of objects along with their relations and attributes. As mentioned above, the same image can be represented by multiple logically-equivalent SGs. Below, we define graph-equivalence more formally and propose an approach to canonicalize graphs to enforce invariance to these equivalences.

Formally, let $\mathcal{C}$ to be the set of objects categories and $\mathcal{R}$ to be the set of possible relations. Objects in SGs also contain attributes but we drop these for notational simplicity. A SG over n objects is a tuple (O, E) where $O \in \mathcal{C}^n$ describing the object categories and E is a set of labeled directed edges (triplets) of the form (i, r, j) where $i, j \in \{1, \ldots, n\}$ and $r \in \mathcal{R}$. Thus an edge (i, r, j) implies that the i^{th} object (that has category o_i) should have relation r with the j^{th} object. Alternatively the set E can be viewed as a set of $|\mathcal{R}|$ directed graphs where for each r the graph E_r contains only the edges for relation r.

Our key observation is that relations in a scene graph are often dependent, because they reflect the semantics of the underlying physical world. This dependence means that for a given relation r, the presence of certain edges in E_r implies that other edges have to hold as well. For example, assume r is a **transitive relation** like "left of" or "below". Then if $i, j \in E_r$ and $j, k \in E_r$ then i and k should also



(a) Input Scene Graph

(b) Soft Relations Closure Module

(c) Soft relations GCN

(d) Generated Scene Layout

Figure 2: **Proposed Scene Graph to Layout architecture.** Given an input scene graph (a), the graph is first canonicalized using the SRC module (b). Dashed edges correspond to completed relations that are assigned weights w_e. Next, a GCN is applied to the resulting weighted graph (c), and its output representations are used to generated the predicted layout (d).

satisfy the relation r. Similarly, if r is a **symmetric relation** then $i, j \in E_r$ implies that the relation j, i also holds. Finally, there are also dependencies between different relations. For example, if r, r' are converse relations (e.g., r is “left of” and r' “right of”) then $i, j \in E_r$ implies $j, i \in E_{r'}$.

Formally, all the above dependencies correspond to first order logic formulas. For example, the fact that a relation r is symmetric corresponds to the formula $\forall i, j : r(i, j) \implies r(j, i)$. Let $\mathcal{F}$ denote a set of such formulas.

The fact that certain relations are implied by a graph does not mean that they are contained in its set of relations. For example, E may contain $(1, \text{left of}, 2)$ but not $(2, \text{right of}, 1)$. This is because empirical graphs E are created by human annotators, and they typically skip redundant edges that can be inferred from other edges. Importantly, we wish that a SG that contains only the relation $(2, \text{left of}, 1)$ or both these relations to result in the same image as a graph that contains either of the relation and also both of them. In other words, we would like all logically equivalent graphs to result in the same image. We next provide some more notation for formalizing this.

Given a scene graph E (we leave out O in what follows) denote by $Q(E)$ the set of graphs that are logically equivalent to E.[1] As mentioned above, we would like all these graphs to result in the same generated image. Currently, architectures addressing SG-to-layout generation, do not have this invariance property because they operate on E and thus sensitive to whether it has certain edges or not. A natural approach to solve this is to replace E with a *canonical form* $C(E)$ such that $\forall E' \in Q(E)$ we have $C(E') = C(E)$.

There are several ways of defining $C(E)$. Perhaps the most natural one is the “relation-closure” which is the graph containing all relations implied by those in E.

Definition 3.1. *Given a set of logical formulas $\mathcal{F}$ describing a set of relations, the relation-closure of $C(E)$ is the set of all relations that are true in any SG that contains relations E and satisfies $\mathcal{F}$.*

We note that the above definition coincides with the standard definition for closure of relations. Our definition merely emphasizes the fact that $C(E)$ are relations that are necessarily true given those in E. Additionally we allow for multiple relations, whereas closure is typically defined with respect to a single property (e.g., transitivity or symmetry).

Next we describe how to calculate $C(E)$ when $\mathcal{F}$ is known, and then explain how to learn $\mathcal{F}$ from data.

3.1. Calculating Relation Closures

For a general set of formulas, calculating the closure is hard as it is an instance of inference in first order logic. However, here we restrict ourselves to a certain subset of formulas for which this calculation is efficient. In what follows we restrict our attention to the following two formulas:

- Transitivity: We assume a set of relations $\mathcal{R}_{trans} \subset \mathcal{R}$ where all $r \in \mathcal{R}_{trans}$ have a corresponding formula $\forall x, y, z : r(x, y) \wedge r(y, z) \implies r(x, z)$.
- Converse Relations: We assume a set of relations pairs $\mathcal{R}_{conv} \subset \mathcal{R} \times \mathcal{R}$ where all $(r, r') \in \mathcal{R}_{conv}$ have a corresponding formula $\forall x, y : r(x, y) \implies r'(y, x)$.

We note that we could have added an option for symmetric relations, but since these are not exhibited in the dataset we use, we do not include those.

Under the above set of formulas, the relation closure $C(E)$ can be computed via the following procedure:
Initialization: Set $C(E) = E$.
Converse Relations: For all $(r, r') \in \mathcal{R}_{conv}$, if $(i, r, j) \in E$, add (j, r', i) to $C(E)$. Call this the *Converse Closure*.

[1] Equivalence of course depends on what relations are considered, but we do not specify this directly to avoid notational clutter.

Transitivity: For each $r \in \mathcal{R}_{trans}$ calculate the transitive closure of $C_r(E)$ (namely the r relations in $C(E)$) and add it to $C(E)$. The transitive closure can be calculated using the Floyd-Warshall algorithm [8].

Lemma 3.2. *The procedure outputs the closure* $C(E)$.

Proof. See Section C in the Supplementary file. ■

3.2. Soft Relation Closures for SG-to-Layout

Thus far we assumed that the sets R_{trans}, R_{conv} were given. Generally, we don't expect this to be the case. We next explain how to construct a model that doesn't have access to these and how to use it for the SG-to-layout task. A high level description of the architecture which employs the completion process is shown in Figure 2.

The Soft Relations Closure Module. Since we do not know which relations are transitive or converses, we use learned parameters towards this end. Specifically, for each $r \in \mathcal{R}$ we learn the probability that they are transitive $\boldsymbol{w}^{trans} \in [0,1]^{|\mathcal{R}|}$ and for each pair of relations $r, r' \in \mathcal{R} \times \mathcal{R}$ we learn the probability they are the converse of each other $\boldsymbol{w}^{conv} \in [0,1]^{|\mathcal{R}| \times |\mathcal{R}|}$.[2] Thus, we also allow for relations to be "softly transitive". In practice this means that if relation r has $w_r^{trans} = 0.7$ then all the completed edges for this relation will have a weight of 0.7 whereas the edges in the original graph will have a weight of 1.

Given a scene graph E, we perform Soft Relation Closure (SRC), as described in Algorithm 1. The input to SRC is a scene graph E, and the output is a weighted scene graph E, where each edge (i, r, j) now also has an assigned weight $w \in [0, 1]$. The function $AssignWeight(E', w)$ simply assigns the weight w to all edges in the graph E'. The closure functions are defined in 3.1.

Algorithm 1 Soft Relations Closure

```
procedure SRC($E$)
    $E' \leftarrow \phi$                                          ▷ Initialization
    $E_{subsets} \leftarrow \{E_r | r \in \mathcal{R}\}$
    for $r \in \mathcal{R}$ do
        $AssignWeight(E_r, 1)$
        $E' \leftarrow E' \cup E_r$
    $\mathcal{R}_{pairs} \leftarrow \{r_i, r_j | r_i, r_j \in \mathcal{R}x\mathcal{R} \wedge r_i < r_j\}$
    for $r_i, r_j \in \mathcal{R}_{pairs}$ do                    ▷ Converse Completion
        $E_{CC} \leftarrow ConverseClosure(E_{r_i}, E_{r_j})$
        $E_{new} \leftarrow E_{CC} - E_{r_i} - E_{r_j}$
        $AssignWeight(E_{new}, w_{i,j}^{conv})$
        $E' \leftarrow E' \cup E_{new}$
    for $r_i \in \mathcal{R}$ do                                 ▷ Transitive Completion
        $E_{TC} \leftarrow TransitiveClosure(E_{r_i})$
        $E_{new} \leftarrow E_{TC} - E_{r_i}$
        $AssignWeight(E_{new}, w_i^{trans})$
        $E' \leftarrow E' \cup E_{new}$
    return $E'$
```

The SRC procedure is essentially the same as the procedure in 3.1, only with weights assigned to the added relations, rather than adding them with a weight of 1.[3]

A Graph Convolution Network for Soft Relations. The layout of each object is influenced by the other objects in the image and their configuration. Thus, it is natural to use graph convolutional networks (GCN) to propagate information in the scene graph [15, 11, 1, 52, 53]. Since in our case edges are weighted, we propose a novel GCN architecture for this case, as described below.

Each category $c \in \mathcal{C}$ is assigned a learned embedding $\boldsymbol{\phi}_c \in \mathbb{R}^D$ and each relation $r \in \mathcal{R}$ is assigned a learned embedding $\boldsymbol{\psi}_r \in \mathbb{R}^D$. Given a scene graph with n objects, the GCN iteratively calculates a representation for each object and each relation in the graph. Let $\boldsymbol{v}_i^k \in \mathbb{R}^d$ be the representation of the i^{th} object in the k^{th} layer of the GCN. Similarly, for each edge $e = (i, r, j)$ in the graph let $\boldsymbol{u}_e^k \in \mathbb{R}^d$ be the representation of the relation in this edge. These representations are calculated as follows. Initially we set: $\boldsymbol{v}_i^0 = \boldsymbol{\phi}_{o(i)}, \boldsymbol{u}_e^0 = \boldsymbol{\psi}_{r(e)}$, where $r(e)$ is the relation for edge e. Next, we use three functions F_s, F_r, F_o, each from $\mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R}^D$ to $\mathbb{R}^D$. These can be thought of as processing three vectors on an edge (the subject, relation and object representations) and returning three new representations. Given these functions, the updated object representation is:

$$\boldsymbol{v}_i^{t+1} = \sum_{e=(i,r,j)} \hat{w}_e F_s(\boldsymbol{v}_i^t, \boldsymbol{u}_e^t, \boldsymbol{v}_j^t) + \sum_{e=(j,r,i)} \hat{w}_e F_o(\boldsymbol{v}_j^t, \boldsymbol{u}_e^t, \boldsymbol{v}_i^t) \quad (1)$$

where $\hat{w}$ is a normalized version of w. Namely:

$$\hat{w}_e = \frac{w_e}{\sum_{e=(i,r,j)} \hat{w}_e + \sum_{e=(j,r,i)} \hat{w}_e} \quad (2)$$

And for the edge we set: $\boldsymbol{u}_e^{t+1} = F_r(\boldsymbol{v}_i^{t+1}, \boldsymbol{u}_e^t, \boldsymbol{v}_j^{t+1})$.

From GCN-to-layout. The last step transforms the GCN representations above to a SG as follows. let L denote the number of updates in the GCN. The bounding box are the four outputs of an MLP applied to $\boldsymbol{v}_i^L$.

3.3. From Layout to Image

In 3.2 we described our model for generating a layout from a SG using the notion of soft relation closure. If our goal is to generate an image from the layout, an additional

[2] To keep the weights in $[0, 1]$ we apply a sigmoid function to an unconstrained variable.

[3] Indeed, note that when the $\boldsymbol{w}$ weights are binary, Algorithm 1 is equivalent to the procedure in 3.1

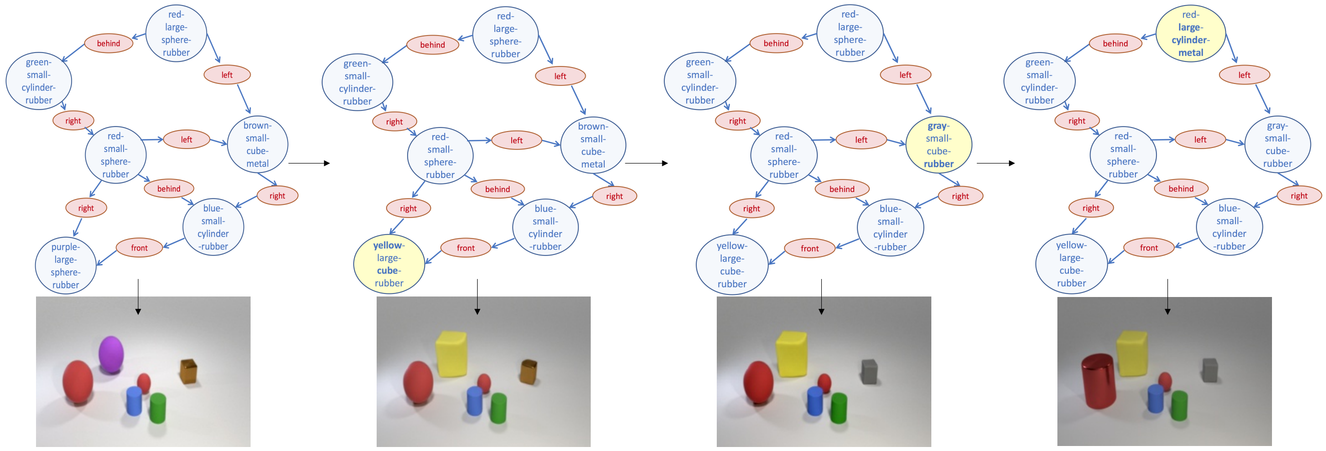


Figure 3: Demonstration of the AttSPADE generator for scene graphs with varying attributes.

layout-to-image model is required. Indeed, several works have proposed models for this step [43, 59]. However, none of these approaches support generating with attributes such as color, shape, material, etc., which exist in the CLEVR dataset for example. Thus we also propose a novel generator, AttSPADE, that supports attributes. The idea is that mask-based models essentially work with a single attribute (the mask) and this can be extended to a vector of attributes rather naturally. Due to space limits we provide more details in Section A in the Supplementary. Figure 3 shows an example of the model trained on CLEVR and applied to several SGs.

4. Experiments

To evaluate our proposed SRC model, we measure performance on two tasks. First, we evaluate on SG-to-layout task, the task that the SRC model is designed for (see 3.2). Then, we further use these layouts to generate images (layout-to-image) and demonstrate that improved layouts also yield improved generated images.

4.1. Datasets

We evaluate SRC on three benchmark datasets: COCO-stuff dataset [3], Visual Genome (VG) [23] and CLEVR [16]. We also created a synthetic dataset to quantify the performance of SRC in a controlled settings where we can control the number of objects. Details about the datasets are provided below.

Synthetic dataset. To test the contribution of learned transitivity to scene-graph-to-layout model, we generate a synthetic dataset. In this data, every object is a square with one of two possible sizes, $small$ or $large$. The set of relations includes

- $Above$ - The center of the subject is above the object. This relation is transitive.
- $OppositeHorizontally$ - The subject and the object are on opposite sides of the image with respect to the middle vertical line. This relation is not transitive.
- $XNear$ - The subject and object are within distance equal to 10% of the image with respect to the x coordinate of each center. This relation is not transitive.

To generate training and evaluation data, we uniformly sample coordinates of object centers and object sizes and automatically compute relations among object pairs based on their spatial locations. See Supplementary file for further visual examples.

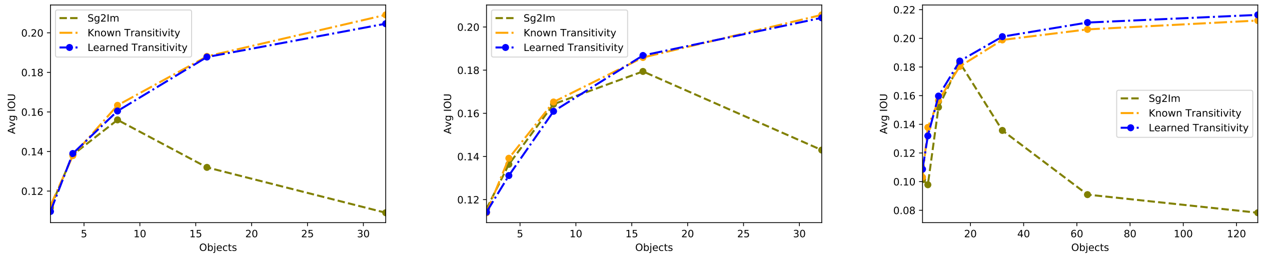
COCO-Stuff 2017 [3]. This dataset contains pixel-level stuff annotations with 40K train and 5K validation images with bounding boxes and segmentation masks for 80 thing categories, and 91 stuff categories. We use two subsets:

- The standard split as in previous work, which contains $\sim$25K training, 1024 validation, and 2048 test images.
- Packed COCO, a subset of COCO containing all images with least 16 objects. Contains $4,341$ train images, 238 validation, and 238 test images.

Visual Genome (VG) [23]. Contains $108,077$ images annotated with SGs. We use two different subsets:

- Standard split, as in previous work: 62K training images, 5506 validation and 5088 test images.
- Packed VG, a subset of VG containing all images with at least 16 objects. This results in 6341 train, 809 validation, and 809 test images.

CLEVR [16]. A dataset synthetically generated based on scene-graphs with four spatial relations: $left$, $right$, $front$ and $behind$. It has 70k training images and 15k for validation and test. Here, every object also has as a set of attributes: $shape$, $size$, $material$ and $color$.



(a) 2 GCN Layers

(b) 4 GCN Layers

(c) Trained on 16 objects

Figure 4: Synthetic dataset results. (a-b) The effect of the number of GCN layers on accuracy. Curves denote IOU performance as a function of the number of objects. Each point is a model trained and tested on a fixed number of objects given by the x axis. (c) Out of sample number of objects. The model is trained on 16 objects and evaluated on up to 128 objects.

4.2. Implementation Details

In all SG to layout experiments, we use Adam [19] optimizer, setting $lr = 1e^{-4}$. For our SRC model, we add an additional Adam optimizer specifically applied to $\boldsymbol{w}^{conv}$ and $\boldsymbol{w}^{trans}$, setting $lr = 0.1$ to encourage faster convergence of these weights. See Section B in the Supplementary for more details.

4.3. Scene-Graph-to-layout Generation

We evaluate the scene-graph-to-layout module based on $AvgIOU$, $R@0.3$ and $R@0.5$ where $AvgIOU$ is the average soft IOU value and $R@0.3$, $R@0.5$ are the average recall over predictions with IOU larger than 0.3 and 0.5.

Testing the Effect of Number of Objects. We begin by exploring scene-graph-to-layout models and the effect of our model. In this setting, we train models with $\{2, 4\}$ GCN layers on the synthetic dataset on each up to 32 objects. Additionally, to evaluate generalization to a different number of objects at test time, we train models with fixed 8 GCN layers on $\{16\}$ objects and test on up to 128 objects. For all experiments we evaluate the following models a) A "Learned Transitivity" model that uses SRC to learn the weights of each relation. b) A "Known Transitivity" model that assumes knowledge of the transitive relations in the data, and performs hard completion for those (as in 3.1). Comparison between "Learned Transitivity" and "Known Transitivity" is meant to evaluate how well SRC can learn which relations are transitive. c) A baseline model that does not use any relation completion, but otherwise has the same architecture (i.e., the architecture of [15]).

Results are shown in Figure 4a-b. First, it can be seen that the baseline (no relation closure) performs significantly worse than the transitivity based models. Second, "Learned Transitivity" closely matches the "Known Transitivity" indicating that the model successfully learned which relations are transitive without supervision (we also confirmed this by looking at the $\boldsymbol{w}$ weights). Third, the baseline model requires more layers to correctly capture scenes with more objects, whereas our models already perform well with two layers. This suggests that SRC indeed improves generalization ability by capturing invariances.

Figure 4c shows that our model also generalizes well when evaluated on a much larger set of objects than those it has seen at training time, whereas the accuracy of the baseline severely degrades for these out of sample scenes.

Layout Accuracy on Packed Scenes. Layout generation is particularly challenging in dense scenes. To quantify this we evaluate on the Packed COCO and Packed VG datasets. Since Sg2Im [15] and Grid2Im [1] contain the same graph model we compare SRC to Sg2Im [15] model, and modify it such that it includes 8 and 16 Graph Convolution layers. Lastly, we reproduce similar experiments on the standard COCO/VG splits which contain relatively few objects.

Evaluation on the Packed versions of COCO and VG is reported in Table 2. It can be seen that the SRC improves layout on all metrics. We also evaluate on the standard splits of COCO and VG, that have less objects and we therefore expect SRC to not improve there. Results are shown in Table 1, showing comparable performance to baselines.

Generalization on Semantically Equivalent Graphs. A key advantage of SRC is that it produces similar layouts for semantically equivalent graphs. This is not true for methods that do not apply canonicalization. To test the effectiveness of this property, we modify the test set such that every input SG is replaced with a different semantically equivalent variation. For example if the original test SG was $(A, \text{right of}, B)$ we may change it to $(B, \text{left of}, A)$. To achieve this, we randomly generate a semantically equivalent SG by randomly choosing to include or exclude edges which do not change the semantics of the input SG. We evaluate on the Packed COCO dataset. Further details are provided in Section E in the Supplementary material.

Results are shown in Table 3. It can be seen that SRC significantly outperforms baselines, demonstrating the advantage of using a canonical representation. The results also

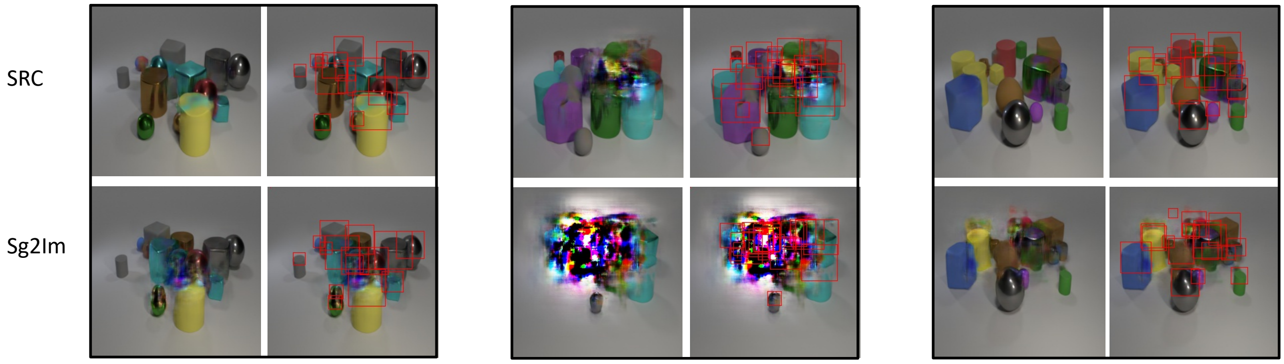


Figure 5: Examples of image generation from the CLEVR dataset with 16+ objects. Shown are three examples, each with four panels: Top row: our generation (with boxes and without). Bottom row: baseline generation (with boxes and without).

highlight the fact that baseline models are very sensitive to semantic-equivalence changes.

4.4. Scene-graph-to-image Generation

To test the contribution of our proposed Scene-Graph-to-layout approach to the overall task of Scene-graph-to-image generation, we further test it in an end-to-end pipeline for generating images. For Packed COCO and Packed VG, we compare our proposed approach with Sg2Im [15] using a fixed pretrained LostGAN [43] as layout-to-image generator. For CLEVR, we use our own AttSPADE generator (3.3).

We evaluate performance using Inception score [40] and Human studies where Amazon Mechanical Turk raters are asked to rank the quality of two images: one generated using our layouts, and the other using SG2Im layouts.

Table 4 describes the quality of the generated images, using a fixed layout-to-image generator. Results on COCO and VG indicate that SRC improves the overall quality of generated images. On CLEVR, Table 5, we use our AttSPADE generator from Section 3.3 which can take into account attributes during image generation. SRC improves the IOU over Sg2Im thanks to the improved layout. Furthermore, when raters are asked to rank which image is better, they prefer images generated using SRC. **In 93% of the cases, our generated images were ranked higher than SG2Im**. Finally, Figure 5 and Figure 6 provide qualitative examples and comparisons of images generated based on CLEVR and COCO. More generation results on COCO and Visual Genome datasets can be seen in Section A.3 in Supplementary.

	Avg IOU		R@0.3		R@0.5	
	COCO	VG	COCO	VG	COCO	VG
	3-8	3-10	3-8	3-10	3-8	3-10
Sg2Im [15][4]	-	-	52.4	21.9	32.2	**10.6**
Sg2Im [15][5]	41.7	**16.9**	62.6	**24.7**	37.5	9.7
SRC (ours)	**42.0**	16.6	**63.0**	24.0	**38.5**	9.0

Table 1: Accuracy of predicted bounding boxes. Models trained on images with 3 to 10 objects for VG and 3 to 8 for COCO.

	Avg IOU		R@0.3		R@0.5	
	COCO	VG	COCO	VG	COCO	VG
	16+	16+	16+	16+	16+	16+
Sg2Im [15][4]	35.8	25.4	56.0	36.2	25.3	15.8
Sg2Im [15] 8 GCN[5]	37.2	25.8	58.6	36.9	26.4	15.9
Sg2Im [15] 16 GCN[5]	37.7	27.1	60.3	39.0	26.6	17.0
SRC (ours)	**39.2**	**28.8**	**62.9**	**43.0**	**29.0**	**18.8**

Table 2: Accuracy of predicted bounding boxes on packed scenes. Models were trained on images containing at least 16 objects.

	Avg IOU	R@0.3	R@0.5
Sg2Im [15] 5 GCN[5]	21.8	29.5	10.7
Sg2Im [15] 8 GCN[5]	23.6	33.2	11.4
Sg2Im [15] 16 GCN[5]	21.6	29.0	10.1
SRC (ours)	**27.6**	**39.4**	**18.5**

Table 3: Evaluation of Canonical representations on the COCO dataset. Every input SG sample is randomly mapped into a semantically equivalent SG.

[4] Results copied from manuscript.

[5] Our implementation of [15]. This is the same as our model without the relation-closure module.

[6] w/o location features

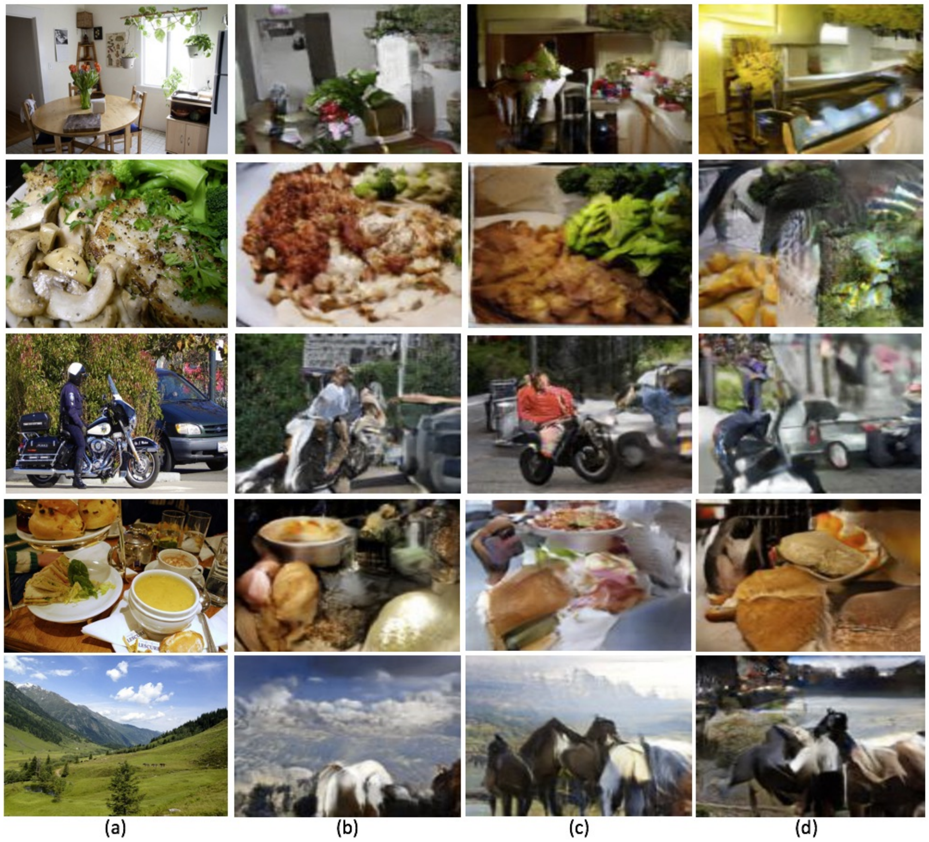


Figure 6: Selected Scene-graph-to-image generation results on the Packed-COCO dataset. Here, we fix the layout-to-image model to LostGAN [44], while changing different scene graph-to-layout models. (a) GT image. (b) Generation from GT layout. (c) our SRC model with LostGAN [44]. (d) Sg2Im [15] model with LostGAN [44].

SG2Layout	Layout2Im	Dataset	Inception
Grid2Im [1][6]	Grid2Im [1]	COCO	7.26 ± 1.1
Sg2Im [15]	LostGAN [44]	COCO	8.0 ± 0.6
SRC (ours)	LostGAN [44]	COCO	$\mathbf{8.6 \pm 0.9}$
Sg2Im [15]	LostGAN [44]	VG	10.0 ± 0.7
SRC (ours)	LostGAN [44]	VG	$\mathbf{10.5 \pm 1.1}$

Table 4: Results for Scene-graph-to-image on 128 × 128. We fix a layout-to-image architecture and test the effect of different SG-to-layout models, for the **packed** datasets versions (16+ objects).

SG2Layout	Layout2Im	Dataset	IOU	Human
Sg2Im [15]	AttSPADE	CLEVR	0.05	7%
SRC (ours)	AttSPADE	CLEVR	**0.15**	**93%**

Table 5: Results for Scene-graph-to-image on CLEVR for 256 × 256, evaluated on test scenes with 16-32 objects per image.

5. Conclusion

We presented a method for mapping SG to images, that is invariant to a set of logical equivalences. Empirical results show that the method results in improved layouts and as a result also improved image quality. We observe empirically that introducing canonical representations allows one to handle packed scenes with fewer layers than the non canonical approach. Our results also suggest improved IOU accuracy even against stronger baselines (e.g., [15] with 16 layers). Intuitively, this is because the transitive closure calculation effectively propagates information across the graph, thus saves the need for propagation using neural architectures. The advantage is that this step is hard-coded and not learned, thus reducing the size of the model. Finally, our approach also results in models that generalize better to semantically equivalent inputs, due to canonization process.

Our results show the advantage of preprocessing a SG before layout generation. Here we studied this in the context of two types of relation properties. However, it can be extended to more complex ones. In this case, finding the closure will be computationally hard, and would amount to performing inference in a Markov Logic Network [39]. On the other hand, it is likely that modeling such complex invariances will result in further robustness of the learned models, and is thus an interesting direction for future work.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC HOLI 819080). Prof. Darrell's group was supported in part by DoD, NSF, BAIR, and BDD. This work was completed in partial fulfillment for the Ph.D degree of the first author.

Supplementary Material

Here we provide additional details regarding our submission, including implementation details and additional results.

A. Layout-to-image with AttSPADE

In Section 3.3, we discussed on how the AttSPADE model extends the paradigm of [33] by modeling multiple semantic attributes per pixel or box rather than a single class descriptor. A high level description of the architecture is included in Figure 7. Let $b \in \mathbb{L}^{A\times H\times W}$ be a bounding box where A is a set of integers defined by different attributes per box, and H and W are the boxes height and width. Each box contains a different set of attributes. For example, for CLEVR [16] we have a set of attributes such as size, shape, material and color, while in Visual Genome [23] we use the natural set of attributes defined by the original proposed split set of [15]. Unlike [33, 1], our model can use masks or boxes as given input with an embedding layer of size 128 to represent each attribute per pixel or box. Then, we concatenate these attribute embeddings and apply a FC layer of 128 to get a unique representation per box or pixel to obtain the entire attributes information.

Lastly, our model uses two discriminators: one for the image (to achieve better quality of the entire image), and one for the boxes (in order to better capture each box).

A.1. The Loss Functions

The generator is trained with the same multi-scale discriminator and loss function used in pix2pixHD [48], except we replace the least squared loss term [26] with the hinge loss term [24, 30, 56]. Since our Layout-to-image model generates the image from a given layout of bounding boxes, we add a box term loss to guarantee that the generated objects in these boxes look real.

A.2. Baseline Models

We evaluate AttSPADE using two types of experiments. First, using the ground truth (GT) layout, meaning layout-to-image task, and second, inferred the layout in an end-to-end manner by the scene-graph-to-image task.

Table 6 compares between the following state-of-the-art models on 128 × 128 and 256 × 256 resolutions:

Grid2Im [1]. The model proposed by Ashual et al. contains the same graph model as [15], but it also uses the supervision of a grid in the generation process. The grid contains the approximate location of the objects in a coarse 5×5 grid, and thus practically very close to using a GT layout. Since our goal here is to test the end-to-end method from scene-graph to image, we trained [1] (code was provided from the authors of [1]) without the "grid attributes" and refer to it as "Grid2Im No-Att". Moreover, Grid2Im [1] did not have any results on Visual Genome dataset to compare with.

LostGAN [44]. The model proposed by Sun et al. is the most recent state-of-the-art model on the task of generation from layout. Therefore, it is natural to compare to it on both layout-to-image and scene-graph-to-image with our generated layout using the SRC graph model.

All models were tested for a fair comparison with the same external code evaluation metrics and will be provided upon acceptance including our models and code implementation.

A.3. Results

The results in Table 6 suggest that the AttSPADE model improves over previous approaches [1, 44] when inferring an image from a GT layout. Also, our end-to-end model, which includes SRC + AttSPADE models, is better on a generation from scene-graph task on both COCO and Visual Genome datasets. The qualitative results on COCO for generation from a GT layout can be seen in Figure 9, while Figure 10 shows a direct comparison between different baselines: AttSPADE with GT layout and SRC + AttSPADE model without GT layout. Additional qualitative results on Visual Genome can be seen in Figure 11 and Figure 12.

B. Implementation Details

B.1. SG to Layout

We apply a GCN with 5 layers and an embedding layer of 128 units for each object and relation. For every Graph Convolution hidden layer, we use 512 units.

B.2. AttSPADE

We apply Spectral Norm [30] to all the layers in both generator and discriminator. We use the ADAM solver [20] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and a learning rate of 0.0001 for both the generator and the discriminator. All the experiments are conducted on NVIDIA V100 GPUs. We use pytorch synchronized BatchNorm with the following batch sizes: 32 for 128 × 128 and 64 × 64 resolutions and 16 for 256 × 256 (statistics are collected from all the GPUs).

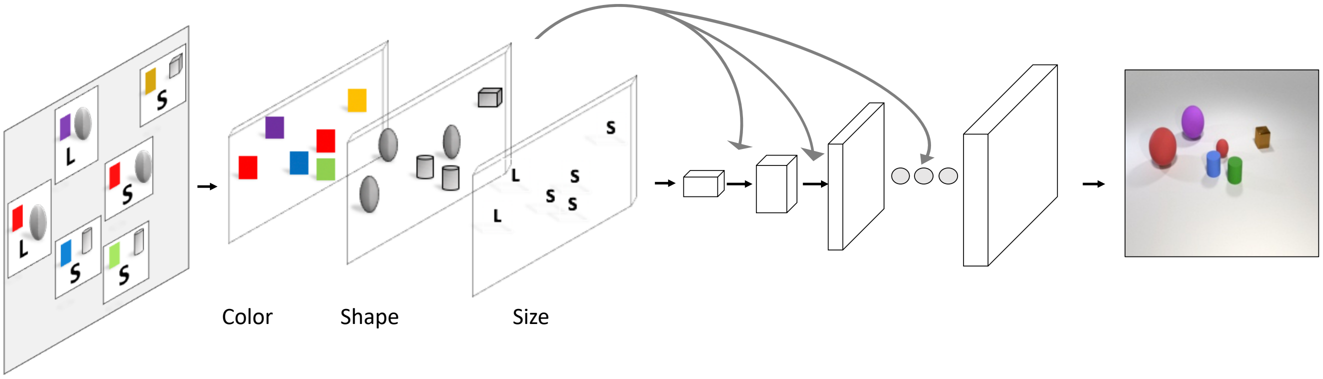


Figure 7: Generating images with AttSPADE. Given a layout of boxes or masks, our model generates an image using attributes layout into a series of residual blocks with upsampling layers.

Resolution	Methods	Inception Score		FID		Diversity Score	
		COCO	VG	COCO	VG	COCO	VG
128x128	Real Images	20.4 ± 3.5	20.5 ± 1.5	-	-	-	-
	Grid2Im [1] GT Layout	12.5 ± 0.3	-	59.5	-	-	-
	LostGAN [44] GT Layout	12.3 ± 0.3	8.6 ± 0.4	64.0	66.7	**0.57** ± **0.06**	**0.59** ± **0.06**
	AttSPADE (ours) GT Layout	**15.6** ± **0.5**	**11.0** ± **0.9**	**54.7**	**36.4**	0.44 ± 0.09	0.51 ± 0.08
	SRC + LostGAN [44]	11.1 ± 0.6	8.1 ± 0.3	**65.9**	73.4	0.57 ± 0.06	0.58 ± 0.06
	Grid2Im [1]	10.4 ± 0.4	-	75.4	-	-	-
	SRC + AttSPADE (ours)	**11.2** ± **0.5**	**10.0** ± **0.7**	77.9	**43.7**	**0.58** ± **0.06**	**0.58** ± **0.06**
256x256	Real Images	30.7 ± 1.2	22.7 ± 7.1	-	-	-	-
	Grid2Im [1] GT Layout	16.4 ± 0.7	-	65.2	-	0.48 ± 0.09	-
	AttSPADE (ours) GT Layout	**19.5** ± **0.9**	**17.1** ± **0.7**	**64.65**	**42.9**	**0.55** ± **0.11**	**0.62** ± **0.08**
	Grid2Im [1] No-Att	6.6 ± 0.3	-	127.0	-	0.65 ± 0.05	-
	SRC + AttSPADE (ours)	**11.1** ± 0.5	**16.5** ± **0.6**	**119.1**	**48.2**	**0.72** ± **0.08**	**0.69** ± **0.07**

Table 6: Quantitative comparisons for SG-to-image methods using Inception Score (higher is better), FID (lower is better) and Diversity Score (higher is better). Evaluation is done on the COCO-Stuff and VG datasets.

C. Proof of Lemma 3.2. from the Paper

Lemma C.1. *The procedure described in Section 3.1 of the main paper outputs the closure* $C(E)$.

Proof. Let $G = (O, E)$. Denote $\hat{C}$ be the canonicalization procedure proposed. To show $\hat{C}(E) = C(E)$, it suffices to prove that (1) $C(E) \subseteq \hat{C}(E)$ and (2) $\hat{C}(E) \subseteq C(E)$.

Proof that $\hat{C}(E) \subseteq C(E)$:. Let there be $e \in \hat{C}(E)$ s.t $e = (i, r, j)$. We split into cases by e construction:

- **Original graph edge**. if $e \in E$ then by C definition $e \in C(E)$.

- **Converse constructed edge**. Therefore there exists $r' \in \mathcal{R}$ such that $(r, r') \in \mathcal{R}_{conv}$ and $(j, r', i) \in E$. Then $(j, r', i) \in C(E)$ and therefore $(i, r, j) = e \in C(E)$ by definition.

- **Transitive constructed edge**. Since e was constructed in the $Transitivity$ step, it must hold that $r \in \mathcal{R}_{trans}$ and e was contained in the transitive closure of r. Therefore, after the $ConverseRelations$ step, there existed a directed path $p = (o_{v_1}, ..., o_{v_k})$ with respect to r where $v_1 = i$ and $v_k = j$. To prove $e \in C(E)$, it is enough to show that for every edge in p it is also in $C(E)$. From here, since C respects transitivity, this will follow. Namely, let there be $e' = (i', r, j') \in \{(o_{v_m}, o_{v_{m+1}}) | m \in \{1, .., k\}\}$. If $e' \in E$, then $e' \in C(E)$ and we are done. Otherwise, by the $ConverseRelations$ construction step, there exists r' such that $(r, r') \in \mathcal{R}_{conv}$ and $(j', r', i') \in E$. Therefore, it follows that $(j', r', i') \in C(E)$ and $e' \in C(E)$ and we are done.

Proof that $C(E) \subseteq \hat{C}(E)$: For every $e = (i, r, j) \in C(E)$ we need to show that $e \in \hat{C}(E)$. Since $e \in C(E)$, e is a relation implied by E. If $e \in E$, since $\hat{C}$ does not drop edges, it holds that $e \in \hat{C}(E)$ and we're done. Otherwise, we assume by contradiction that $e \notin \hat{C}(E)$. let $p = (o_{v_1}, ..., o_{v_k})$ be a directed path from o_i to o_j in $C(E)$. Then, there exists $e' = (i', r, j') \in \{(o_{v_i}, o_{v_{i+1}}) | i \leq k\}$ where $e' \notin \hat{C}(E)$. Otherwise, if there is no such e', we get that there is a directed path between o_i to o_j and by $Transitivity$ step construction $e \in \hat{C}(E)$. Therefore, there must be $e_{conv} \in E$,

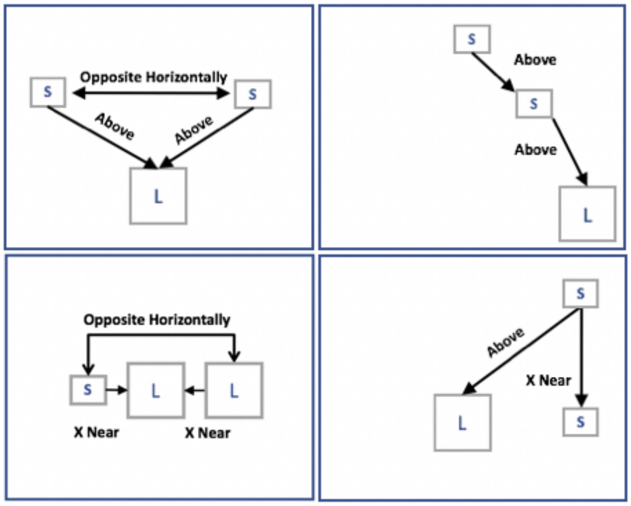


Figure 8: Example of synthetic dataset samples. In these samples, the scene graph relations are overlaid on top of the layout. Every edge is described with a corresponding relation type and every square object is annotated with an object type: ”S” for small and ”L” for large.

such that $e_{conv} = (j, r', i)$ and $(r, r') \in \mathcal{R}_{conv}$. However, from the $ConverseRelations$ step construction, if there exists such edge we get that $e \in \hat{C}(E)$, in contrary to the assumption that $e \notin \hat{C}(E)$. ■

D. Synthetic Dataset

Section 4.1 of the main paper describes a synthetic dataset that is used to explore properties of our algorithm. Figure 8 shows examples of samples in the synthetic dataset. The synthetic dataset was used only in the scene graph to layout model validation.

E. Generalization on Semantically Equivalent Graphs

Results in Table 3 of the main paper demonstrate that the learned SRC model is more robust to changes in the scene graph input. In this experiment, we randomly transform each test sample scene graph into a semantically equivalent one and test models on the resulting sample. To generate such samples from a given scene graph, we start by calculating all the possible location-based relations for any pair of objects. Then, for each pair of objects we use prior knowledge to identify pairs of converse relations, and drop one of the edges in such pair with probability $p = 0.5$. After this step, we compute the transitive closure with respect to each relation and randomly drop ($p = 0.5$) each edge that does not change the semantics of the scene graph.

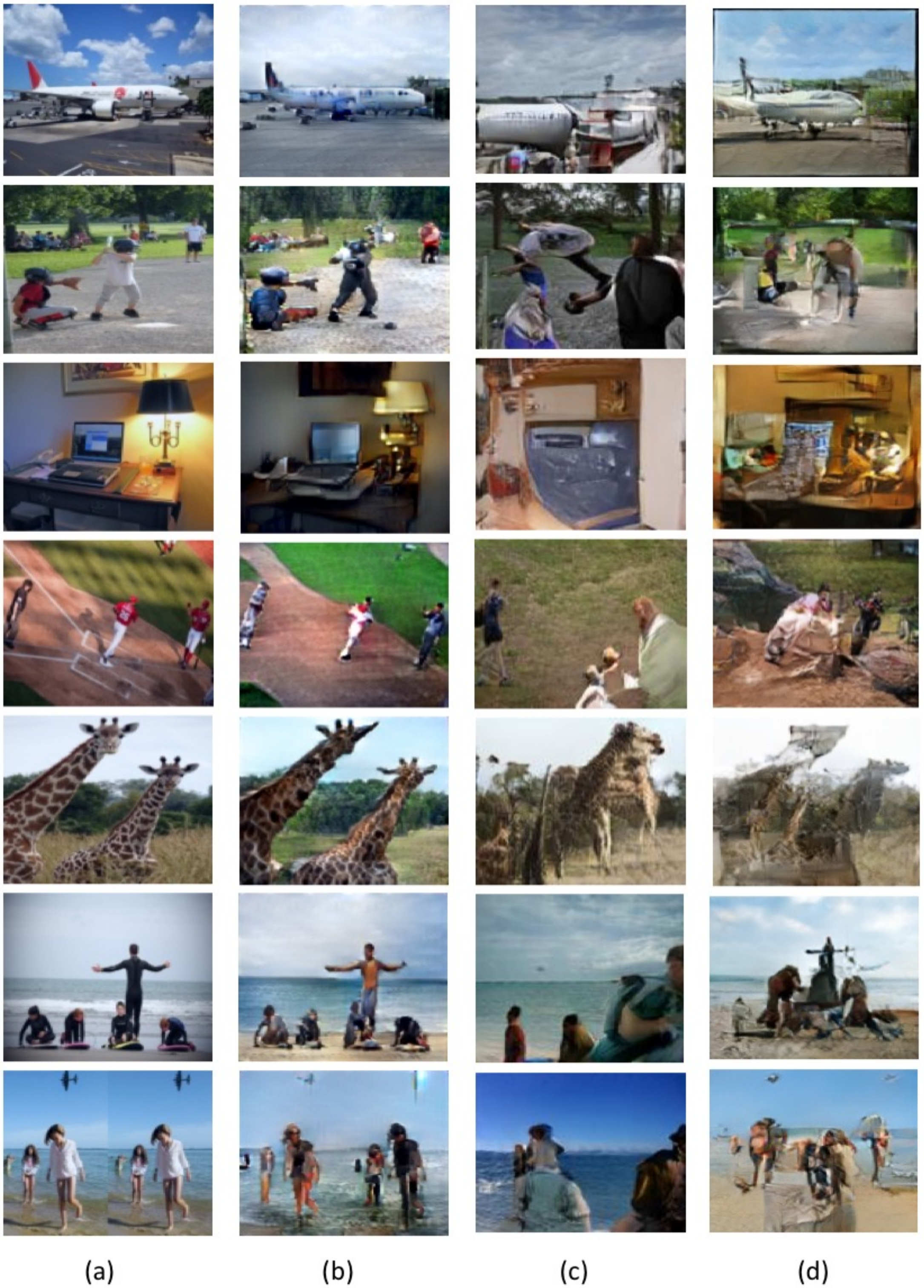


Figure 9: Selected GT layout-to-image generation results on COCO-Stuff dataset on 128×128 resultion. Here, the GT layout also includes masks. (a) GT image. (b) Generation with AttSPADE model. (c) Generation with LostGAN [44] model. (d) Generation with Grid2Im [1].

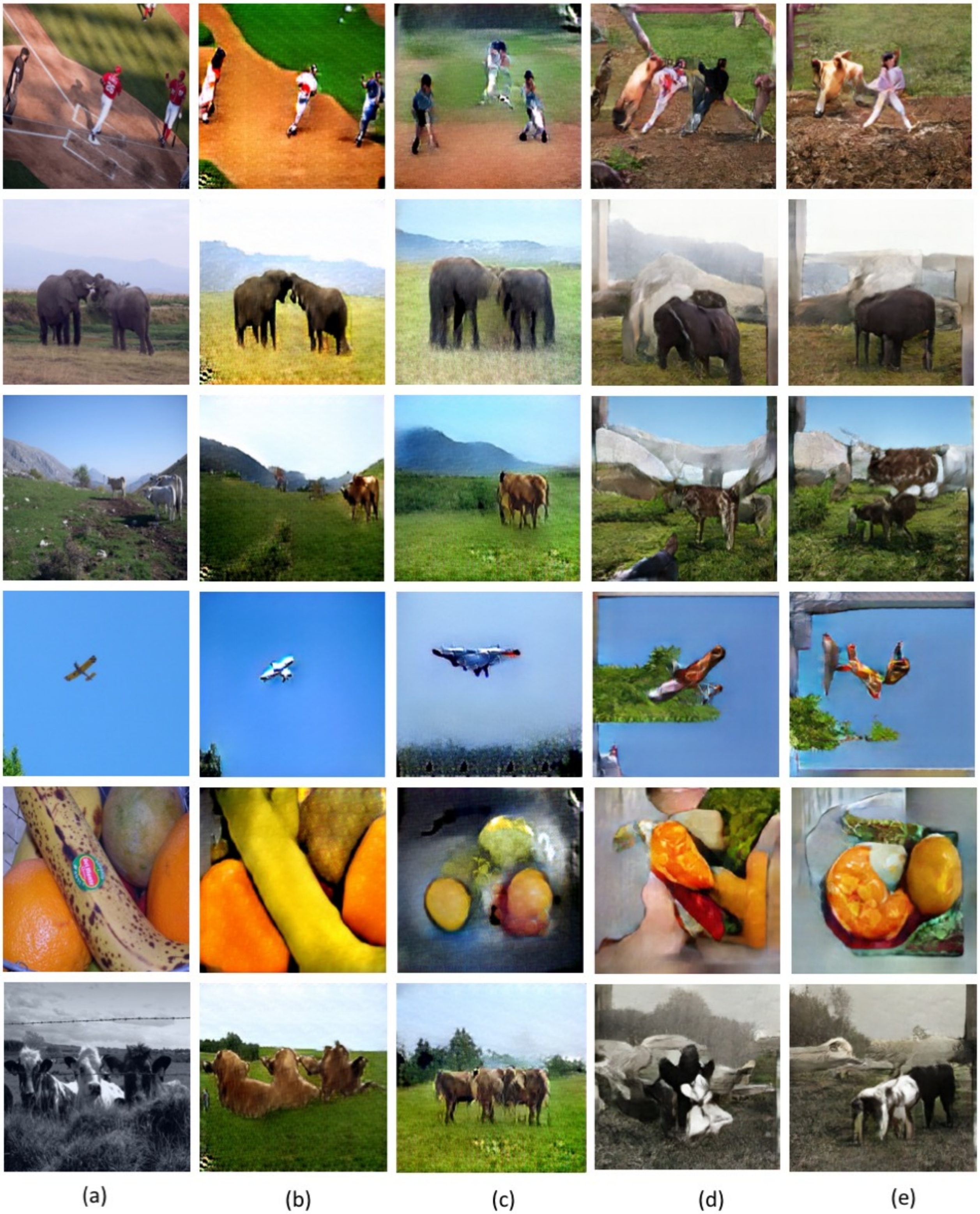


Figure 10: Selected generation results on the COCO-Stuff dataset at 256×256 resolution. Here the GT layout also includes masks. (a) GT image. (b) Generation with AttSPADE model using the GT layout. (c) Generation with SRC + AttSPADE model from the scene graph (GT layout not used). (d) Generation with Grid2Im [1] No-att using the GT layout. (e) Generation with Grid2Im No-att [1] from the scene graph (GT layout not used).

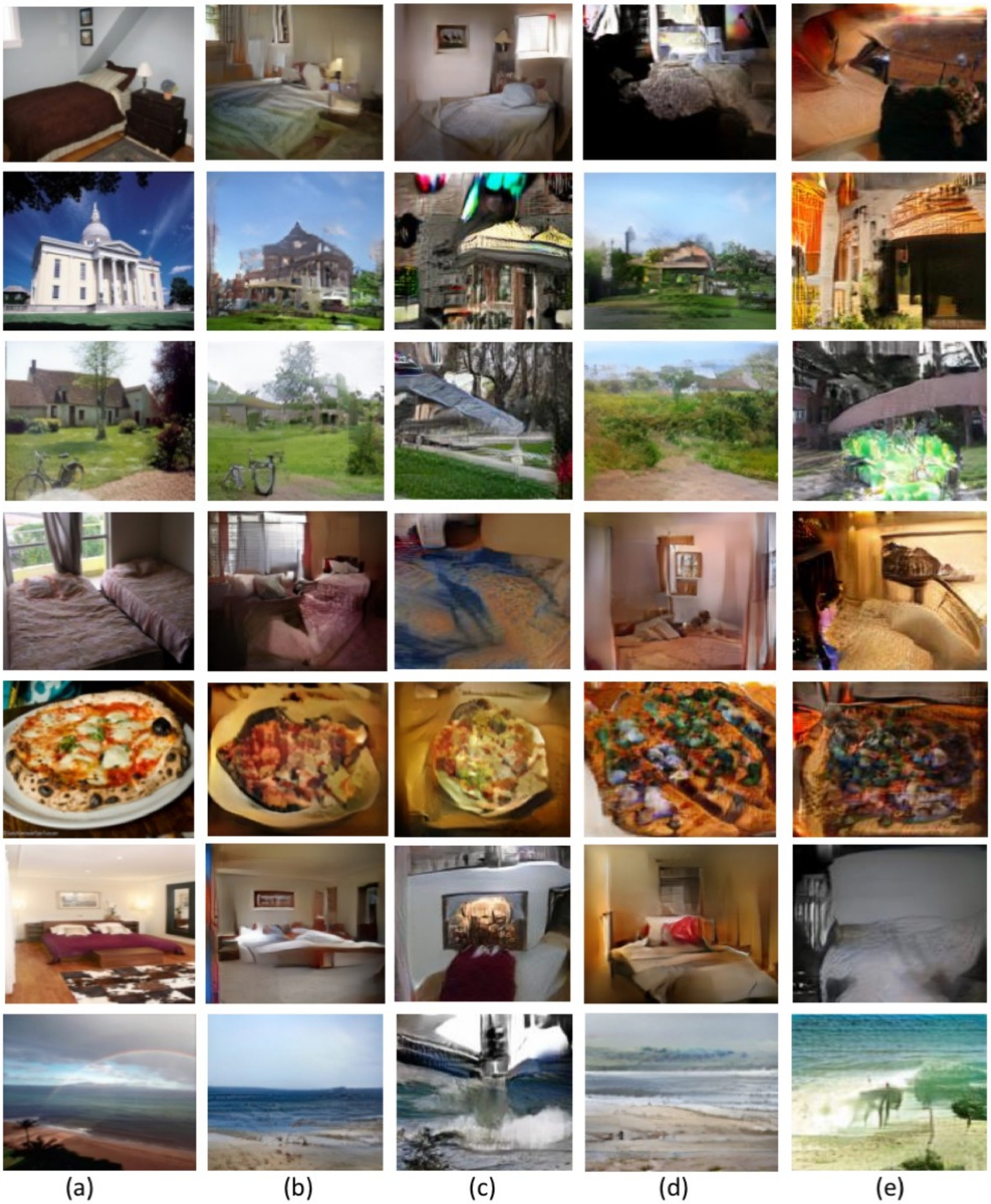


Figure 11: Selected scene-graph-to-image results on Visual Genome dataset on 128×128 resolution. Here, the GT layout includes only boxes. (a) GT image. (b) Generation with the AttSPADE model using the GT Layout. (c) Generation using LostGAN [44] using the GT layout. (d) Generation with the SRC + AttSPADE model using the scene graph (GT layout not used). (e) Generation with the SRC + LostGAN [44] using the scene graph (GT layout not used).

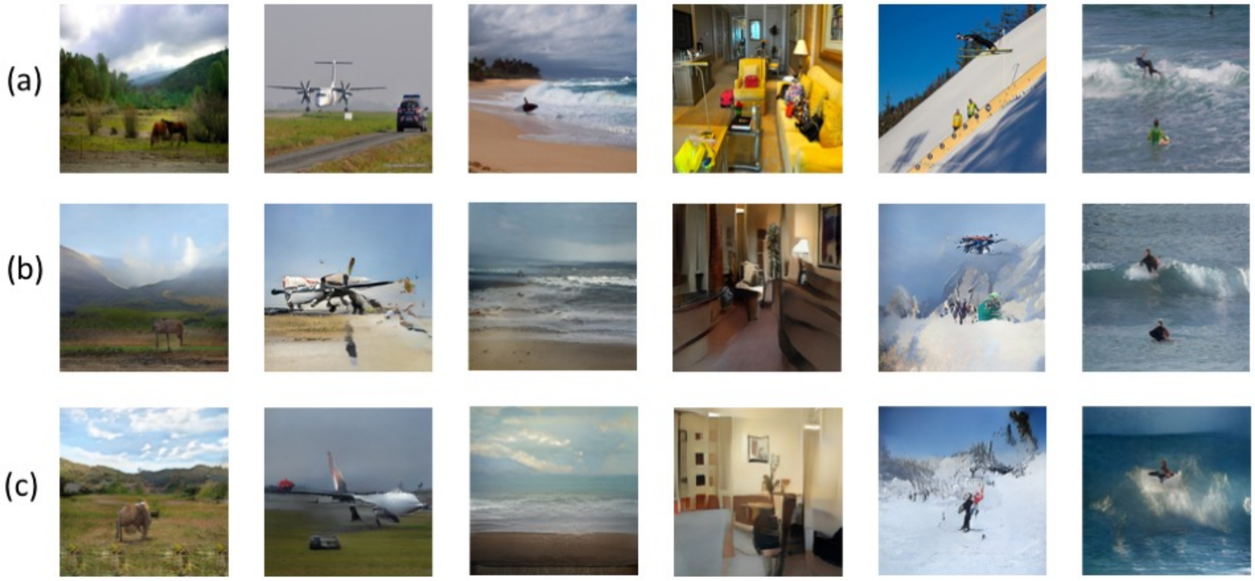


Figure 12: Selected scene-graph-to-image results on the Visual Genome dataset at 256×256 resolution. Here, the GT layout includes only boxes. (a) GT image. (b) Generation with the AttSPADE model using GT Layout. (c) Generation with the SRC + AttSPADE model using the scene graph (GT layout not used).

References

[1] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4561–4569, 2019. 1, 2, 4, 6, 8, 9, 10, 12, 13

[2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2

[3] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[4] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016. 2

[5] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1520, 2017. 2

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. 2

[7] Zhiwei Deng, Jiacheng Chen, Yifang Fu, and Greg Mori. Probabilistic neural programmed networks for scene generation. In *Advances in Neural Information Processing Systems*, pages 4028–4038, 2018. 2

[8] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962. 4

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[10] Roei Herzig, Elad Levi, Huijuan Xu, Hang Gao, Eli Brosh, Xiaolong Wang, Amir Globerson, and Trevor Darrell. Spatio-temporal action graph networks. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 2

[11] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2, 4

[12] 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

[13] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. 2

[14] 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. 2

[15] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 4, 6, 7, 8, 9

[16] 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 *CVPR*, 2017. 5, 9

[17] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015. 2

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. on Learning Representations*, 2015. 9

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[22] Ranjay Krishna, Ines Chami, Michael S. Bernstein, and Li Fei-Fei. Referring relationships. *ECCV*, 2018. 2

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *ArXiv e-prints*, 2016. 5, 9

[24] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 9

[25] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 2

[26] Xudong Mao, Qing Li, Haoran Xie, YK Raymond Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proc. Int. Conf. Comput. Vision*, 2017. 9

[27] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. In *International Conference on Machine Learning*, pages 4363–4371, 2019. 2

[28] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[29] Gaurav Mittal, Shubham Agrawal, Anuva Agarwal, Sushant Mehta, and Tanya Marwah. Interactive image generation using scene graphs. *arXiv preprint arXiv:1905.03743*, 2019. 2

[30] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Int. Conf. on Learning Representations*, 2018. 9

[31] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017. 2

[32] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 2

[33] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 1, 2, 9

[34] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019. 2

[35] Moshiko Raboh, Roei Herzig, Gal Chechik, Jonathan Berant, and Amir Globerson. Differentiable scene graphs. In *Winter Conf. on App. of Comput. Vision*, 2020. 2

[36] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[37] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 2

[38] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. In *Advances in Neural Information Processing Systems*, pages 217–225, 2016. 2

[39] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006. 8

[40] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 2, 7

[41] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. 2

[42] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. Chatpainter: Improving text to image generation using dialogue. *arXiv preprint arXiv:1802.08216*, 2018. 2

[43] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10531–10540, 2019. 5, 7

[44] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 8, 9, 10, 12, 14

[45] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 2

[46] Fuwen Tan, Song Feng, and Vicente Ordonez. Text2scene: Generating compositional scenes from textual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6710–6719, 2019. 2

[47] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016. 2

[48] 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 *Proc. Conf. Comput. Vision Pattern Recognition*, 2018. 9

[49] QIAO Xiaotian, Quanlong ZHENG, CAO Ying, and WH Rynson. Tell me where i am: Object-level scene context prediction. In *The 32nd meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*. IEEE, 2019. 2

[50] Ning Xu, An-An Liu, Jing Liu, Weizhi Nie, and Yuting Su. Scene graph captioner: Image captioning based on structural visual representation. *Journal of Visual Communication and Image Representation*, 58:477–485, 2019. 2

[51] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018. 2

[52] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018. 4

[53] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 684–699, 2018. 4

[54] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. Semantics disentangling for text-to-image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2327–2336, 2019. 2

[55] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems 30*, pages 3394–3404. Curran Associates, Inc., 2017. 2

[56] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Int. Conf. Mach. Learning*, 2019. 9

[57] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE*

International Conference on Computer Vision, pages 5907–5915, 2017. 2

[58] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8584–8593, 2019. 2

[59] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019. 5

[60] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 2

[61] Xingran Zhou, Siyu Huang, Bin Li, Yingming Li, Jiachen Li, and Zhongfei Zhang. Text guided person image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3663–3672, 2019. 2

[62] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2

[63] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017. 2