Text2Scene: Generating Compositional Scenes from Textual Descriptions

Fuwen Tan[1] Song Feng[2] Vicente Ordonez[1]
[1]University of Virginia, [2]IBM Thomas J. Watson Research Center.
fuwen.tan@virginia.edu, sfeng@us.ibm.com, vicente@virginia.edu

Abstract

We propose Text2Scene, a model that interprets input natural language descriptions in order to generate various forms of compositional scene representations; from abstract cartoon-like scenes to synthetic images. Unlike recent works, our method does not use generative adversarial networks, but a combination of an encoder-decoder model with a semi-parametric retrieval-based approach. Text2Scene learns to sequentially produce objects and their attributes (location, size, appearance, etc) at every time step by attending to different parts of the input text, and the current status of the generated scene. We show that under minor modifications, the proposed framework can handle the generation of different forms of scene representations, including cartoon-like scenes, object layouts corresponding to real images, and synthetic image composites. Our method is not only competitive when compared with state-of-the-art GAN-based methods using automatic metrics and superior based on human judgments but it is also more general and interpretable.

1. Introduction

Generating images from textual descriptions has recently become an active research topic [15, 29, 39, 38, 36, 13]. This interest has been partially fueled by the adoption of Generative Adversarial Networks [9] which have demonstrated impressive results on a number of image synthesis tasks. Synthesizing images from text additionally requires a level of language and visual comprehension which could lead to applications in image retrieval through natural language queries, representation learning for text, and automated computer graphics and image editing applications.

In this work, we introduce Text2Scene, a model to interpret important semantics in visually descriptive language in order to generate compositional scene representations. We specifically focus on generating a scene representation consisting of a list of objects, along with their attributes (locations, sizes, aspect ratios, pose, appearance). We adapt and train models to generate three types of scene representations as shown in Figure 1, (1) Cartoon-like scenes from the Abstract Scenes dataset [41] where the objects include locations, sizes, aspect ratios, orientations, and poses (2) Object layouts for scenes in the COCO dataset [22] where the objects include locations, sizes, and aspect ratios, and (3) Synthetic image composites for scenes in the COCO dataset [22] where the objects include locations, sizes, aspect ratios, and pixel-appearance. We propose a unified framework to handle these three seemingly different tasks with unique challenges. Our method, unlike recent approaches, does not rely on Generative Adversarial Networks (GANs). Instead, we produce an interpretable model that iteratively generates the scene by predicting and adding a new object in each step. Our method is superior to the best result reported in Abstract Scenes [41], and provides near state-of-the-art performance on COCO [22] under automatic evaluation metrics, and superior performance than the state-of-the-art when evaluated by humans.

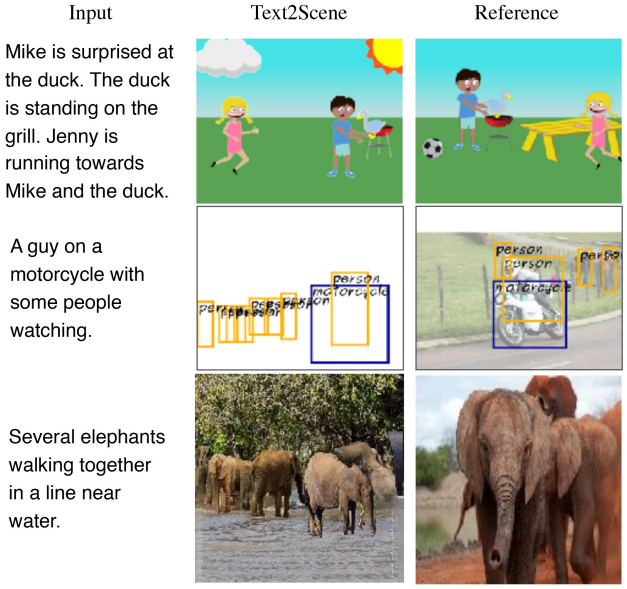

Figure 1. Sample inputs (left) and outputs of our Text2Scene model (middle), along with *ground truth* reference scenes (right) for generation of abstract scenes (top), object layouts (middle), and synthetic image composites (bottom).

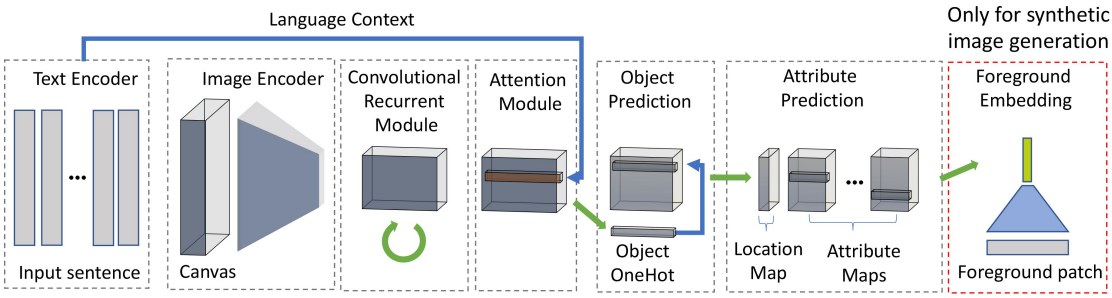

Figure 2. Overview of Text2Scene. Our general framework consists of (1) a Text Encoder that produces a sequential representation of the input, (2) an Image Encoder that encodes the current state of the generated scene, (3) a Convolutional Recurrent Module that tracks, for each spatial location, the history of what has been generated so far, (4) two attention modules that sequentially focus on different parts of the input text, first for deciding what object to place, and then to decide which attributes to assign to the object, and (5) an optional foreground embedding step that learns an appearance vector for patch retrieval in the synthetic image generation task.

Generating rich textual representations for scene generation is a challenging task. For instance, input textual descriptions might only indirectly hint at the presence of attributes (e.g. in the first example in Fig. 1 the input text "Mike is surprised" should change facial attributes on the generated object "Mike"). Textual descriptions also frequently contain complex information about relative spatial configurations (e.g. in the first example in Fig. 1 the input text "Jenny is running towards Mike and the duck" makes the orientation of "Jenny" dependent on the positions of both "Mike", and "duck"). In the last example in Fig. 1 the text "elephants walking together in a line" also implies certain overall spatial configuration of the objects in the scene.

We model this text-to-scene task using a sequence-to-sequence approach where objects are placed sequentially on an initially empty canvas (see an overview in Fig 2). Generally, Text2Scene, consists of a text encoder that maps input sentences to a set of embedding representations, an object decoder that predicts the next object conditioned on the current scene state, and an attribute decoder that determines the attributes of the predicted object.

Our Text2Scene model delivers state-of-the-art results on Abstract Scenes [41] generation and near state-of-the-art results for synthetic image generation on COCO [22]. To the best of our knowledge, Text2Scene is the first model demonstrating its capacity on both abstract and real images, thus opening the possibility for future work on transfer learning across domains.

Our main contributions can be summarized as follows:

- We propose Text2Scene, a framework to generate compositional scene representations from input language descriptions.
- We show that Text2Scene can be used, under minor modifications, to generate different forms of scene representations, including cartoon-like scenes, semantic layouts corresponding to real images, and synthetic image composites.
- We conduct extensive experiments on the tasks of abstract image generation for the Abstract Scenes [41] dataset and synthetic image generation for the COCO [22] dataset.

2. Related Work

Most research on visually descriptive language has focused on the task of image captioning or mapping images to text [6, 24, 19, 16, 34, 35, 25, 2]. Recently, there is work in the opposite direction of using text to synthesize images [29, 38, 15, 39, 36, 13]. Most of the recent approaches have leveraged conditional Generative Adversarial Networks (cGANs). While these works have managed to generate results of increasing quality, there are major challenges when attempting to synthesize images for complex scenes with multiple interacting objects. Inspired by *the principle of compositionality* in language and vision [40], we do not use GANs but use a compositional approach to image generation by sequentially generating objects (e.g. clip-art, bounding box, or segmented object patches) containing the semantic elements that compose the scene, which also makes our model more interpretable.

Our work is also related to prior research on using abstract scenes to mirror and analyze complex situations in the real world [41, 42, 8, 33]. The most related is [42] where a graphical model was introduced to generate an abstract scene from input textual descriptions. Unlike this previous work, our method does not use a semantic parser to obtain a set of tuples but is trained directly from input sentences in an end-to-end fashion. Moreover, we show that our method compares favorably to this previous work. Our work is also related to recent works on generating images from pixel-wise semantic labels [14, 5, 28], especially [28] which proposed a retrieval-based semi-parametric method for image synthesis given an input provided by a human. Our composite image generation model optionally uses the cascaded refinement module in [28] as a post-processing step. Our

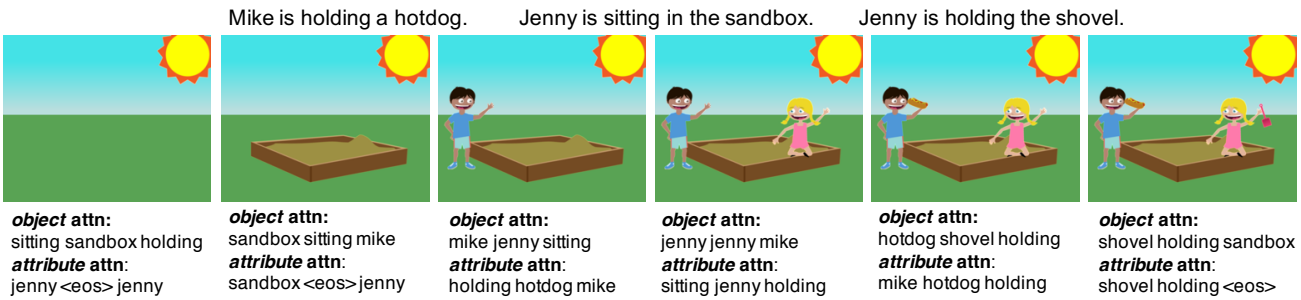

Figure 3. Step-by-step generation of an abstract scene, showing the top-3 attended words for the object prediction and attribute prediction at each time step. Notice how except for predicting the *sun* at the first time step, the attended words in the object decoder are almost a one-to-one mapping with the predicted object. The attended words by the attribute decoder also correspond semantically to useful information for predicting either pose or location, e.g. to predict the location of the *hotdog* in the fifth time step, the model attends to *mike* and *holding*.

work differs from these methods as they use ground-truth semantic layouts as input while our model learns to predict the layout of the objects in the scene indirectly from text.

Most closely related to our approach are [15], [10], [13], and [17], as these works also attempt to predict explicit 2D layout representations. [15] proposed a graph-convolution model to generate images from structured scene graphs. The presented objects and their relationship are provided as inputs in the scene graphs, unlike our work, where the presence of objects is inferred from text. [13] targets image synthesis using conditional GANs but unlike prior works, it generates layouts as intermediate representations in separably trained modules. Our work also attempts to generate photographic images from textual descriptions but unlike [13], we generate pixel-level outputs using a semi-parametric retrieval module without adversarial training. Our model also learns end-to-end to predict the semantic layouts and object patches simultaneously. [17] performs pictorial generation from chat logs, while our work uses text which is considerably more underspecified. The system presented in [10] proposed to generate cartoon-like pictures using a semi-parametric method. However, the presented objects are also provided as inputs to the model, and the prediction of layouts, foreground and background patches is performed by separably trained modules. Our method is trained end-to-end and goes beyond cartoon-like scenes. To the best of our knowledge, our model is the only one targeting various types of scenes (e.g. abstract scenes, semantic layouts and composite images) under a unified framework.

3. Model

Text2Scene adopts a sequence-to-sequence approach [31] and introduces key designs for spatial and sequential reasoning. Specifically, at each time step, the model modifies a background canvas in three steps: (1) the model attends to the input text to decide what is the next object to add, or decides whether the generation should end; (2) if the decision is to add a new object, the model *zooms in* the language context of the object to *decide* its attributes (e.g. pose, size) and relations with its surroundings (e.g. location, interactions with other objects); (3) the model refers back to the canvas and *grounds* (places) the extracted textual attributes into their corresponding visual representations.

To model this procedure, Text2Scene consists of a text encoder, which takes as input a sequence of M words w_i (section 3.1), an object decoder, which predicts sequentially T objects o_t, and an attribute decoder that predicts for each o_t their locations l_t and a set of k attributes $\{R_t^k\}$ (section 3.2). The scene generation starts from an initially empty canvas B_0 that is updated at each time step. In the image composition task, we also jointly train a foreground patch embedding network (section 3.3) and treat the embedded vector as a target attribute. Fig. 2 illustrates the overall pipeline of our model, and Fig. 3 shows a step-by-step generation of an abstract scene.

3.1. Text Encoder

Our text encoder consists of a bidirectional recurrent network with Gated Recurrent Units (GRUs). For a given input text, we compute for each word i:

$$h_i^E = \text{BiGRU}(x_i, h_{i-1}^E, h_{i+1}^E), \quad (1)$$

Here BiGRU is a bidirectional GRU cell, x_i is a word embedding corresponding to the i-th word, and h_i^E is a vector encoding the current word and its context. We use $\{(h_i^E, x_i)\}$ as the encoded text feature.

3.2. Object and Attribute Decoders

At each step t, our model predicts the next object o_t from an object vocabulary $\mathcal{V}$ and its k attributes $\{R_t^k\}$, using the text feature $\{(h_i^E, x_i)\}$ and the current canvas B_t as input. For this part, we use a convolutional network (CNN) Ω to encode B_t into a $\mathcal{C} \times H \times W$ feature map, representing the

current scene state. We model the history of the scene states $\{h_t^D\}$ by a convolutional GRU (ConvGRU):

$$h_t^D = \text{ConvGRU}(\Omega(B_t), h_{t-1}^D), \tag{2}$$

The initial hidden state is created by spatially replicating the last hidden state of the text encoder. Here h_t^D provides an informative representation of the temporal dynamics of each spatial (grid) location in the scene. Since this representation might fail to capture small objects, a one-hot vector of the object predicted at the previous step o_{t-1} is also provided as input to the downstream decoders. The initial object is set as a special start-of-scene token.

Attention-based Object Decoder: Our object decoder is an attention-based model that outputs the likelihood scores of all possible objects in an object vocabulary $\mathcal{V}$. It takes as input the recurrent scene state h_t^D, the text features $\{(h_i^E, x_i)\}$ and the previously predicted object o_{t-1}:

$$u_t^o = \text{AvgPooling}(\Psi^o(h_t^D)), \tag{3}$$

$$c_t^o = \Phi^o([u_t^o; o_{t-1}], \{(h_i^E, x_i)\}), \tag{4}$$

$$p(o_t) \propto \Theta^o([u_t^o; o_{t-1}; c_t^o]), \tag{5}$$

here Ψ^o is a convolutional network with spatial attention on h_t^D, similar as in [35]. The goal of Ψ^o is to collect the spatial contexts necessary for the object prediction, e.g. what objects have already been added. The attended features are then fused into a vector u_t^o by average pooling. Φ^o is the text-based attention module, similar as in [23], which uses $[u_t^o; o_{t-1}]$ to attend to the language context $\{(h_i^E, x_i)\}$ and collect the context vector c_t^o. Ideally, c_t^o encodes the knowledge of all the described objects that have not been added to the scene thus far. Θ^o is a two-layer perceptron predicting the likelihood of the next object $p(o_t)$ from $[u_t^o; o_{t-1}; c_t^o]$, using a softmax function.

Attention-based Attribute Decoder The attribute set $\{R_t^k\}$ corresponding to the object o_t can be predicted similarly. We use another attention module Φ^a to "zoom in" the language context of o_t, extracting a new context vector c_t^a. Compared with c_t^o which may contain information of all the objects that have not been added yet, c_t^a focuses more specifically on contents related to the current object o_t. For each spatial location in h_t^D, the model predicts both a location likelihood $\{l_t^i\}_{i=1...N}$, and attribute likelihoods $\{R_t^k\}$. Here, possible locations are discretized into the same resolution of h_t^D. In summary, we have:

$$c_t^a = \Phi^a(o_t, \{(h_i^E, x_i)\}) \tag{6}$$

$$u_t^a = \Psi^a([h_t^D; c_t^a]) \tag{7}$$

$$p(l_t, \{R_t^k\}) = \Theta^a([u_t^a; o_t; c_t^a]), \tag{8}$$

Φ^a is the attention module using o_t to attend the input text. Ψ^a is a CNN spatially attending h_t^D. The goal of Ψ^a is to find an affordable location to add o_t. Here c_t^a is spatially replicated before concatenating with h_t^D. The final likelihood map $p(l_t, \{R_t^k\})$ is predicted by a convolutional network Θ^a with softmax classifiers over each value of l_t and the discrete R_t^k. For continuous attributes R_t^k such as Q_t (next section), we normalize the output using an ℓ_2-norm.

3.3. Foreground Patch Embedding

We predict a particular attribute Q_t only for the model trained to generate synthetic image composites (i.e. images composed of patches retrieved from other images). As with other attributes, Q_t is predicted for every location in the output feature map but is used at test time to retrieve similar patches from a pre-computed collection of object segments from other images. We train a patch embedding network using a CNN which reduces the foreground patch in the target image into a 1D vector F_t. The goal is to minimize the ℓ_2-distance between Q_t and F_t using a triplet embedding loss [7] that encourages a small distance $||Q_t, F_t||_2$ while encouraging a larger distance with other patches $||Q_t, F_k||_2$. Here F_k is the feature of a "negative" patch, which is randomly selected from the same category of F_t:

$$L_{triplet}(Q_t, F_t) = max\{||Q_t, F_t||_2 - ||Q_t, F_k||_2 + \alpha, 0\} \tag{9}$$

α is a margin hyper-parameter.

3.4. Objective

The loss function for a given example in our model with reference values $(o_t, l_t, \{R_t^k\}, F_t)$ is:

$$\begin{aligned} L = &- w_o \sum_t \log p(o_t) - w_l \sum_t \log p(l_t) \\ &- \sum_k w_k \sum_t \log p(R_t^k) + w_e \sum_t L_{triplet}(Q_t, F_t) \\ &+ w_a^O L_{attn}^O + w_a^A L_{attn}^A, \end{aligned}$$

where the first three terms are negative log-likelihood losses corresponding to the object, location, and discrete attribute softmax classifiers. $L_{triplet}$ is a triplet embedding loss optionally used for the synthetic image generation task. L_{attn}^* are regularization terms inspired by the doubly stochastic attention module proposed in section 4.2.1 of [35]. Here $L_{attn}^* = \sum_i [1 - \sum_t \alpha_{ti}^*]^2$ where $\{\alpha_{ti}^o\}$ and $\{\alpha_{ti}^a\}$ are the attention weights from Φ^o and Φ^a respectively. These regularization terms encourage the model to distribute the attention across all the words in the input sentence so that it will not miss any described objects. Finally, w_o, w_l, $\{w_k\}$, w_e, w_a^O, and w_a^A are hyperparameters controlling the relative contribution of each loss.

Details for different scene generation tasks In the Abstract Scenes generation task, B_t is represented directly as an RGB image. In the layout generation task, B_t is a 3D

Methods	U-obj		B-obj		Pose	Expr	U-obj	B-obj
	Prec	Recall	Prec	Recall			Coord	Coord
Zitnick et al. 2013	0.722	0.655	0.280	0.265	0.407	0.370	**0.449**	0.416
Text2Scene (w/o attention)	0.665	0.605	0.228	0.186	0.305	0.323	0.395	0.338
Text2Scene (w object attention)	0.731	0.671	0.312	0.261	0.365	0.368	0.406	0.427
Text2Scene (w both attentions)	0.749	0.685	0.327	0.272	0.408	0.374	0.402	0.467
Text2Scene (full)	**0.760**	**0.698**	**0.348**	**0.301**	**0.418**	**0.375**	0.409	**0.483**

Table 1. Quantitative evaluation on the Abstract Scenes dataset

Methods	Scores	≥ 1	≥ 2	Obj-Single `sub-pred`	Obj-Pair `sub-pred-obj`	Location `pred:loc`	Expression `pred:expr`
Reference	0.919	1.0	0.97	0.905	0.88	0.933	0.875
Zitnick et al. 2013	0.555	0.92	0.49	0.53	0.44	**0.667**	0.625
Text2Scene (w/o attention)	0.455	0.75	0.42	0.431	0.36	0.6	0.583
Text2Scene (full)	**0.644**	**0.94**	**0.62**	**0.628**	**0.48**	**0.667**	**0.708**

Table 2. Human evaluation on the Abstract Scenes dataset.

tensor with a shape of $(\mathcal{V}, H, W)$, where each spatial location contains a one-hot vector indicating the semantic label of the object at that location. Similarly, in the synthetic image generation task, B_t is a 3D tensor with a shape of $(3\mathcal{V}, H, W)$, where every three channels of this tensor encode the color patch of a specific category from the background canvas image. For the foreground embedding module, we adopt the canvas representation in [28] to encode the foreground patch for simplicity. As the composite images may exhibit gaps between patches, we also leverage the stitching network in [28] as post-processing. Note that the missing region may also be filled by any other inpainting approaches. Full details about the implementation of our model can be found in the supplementary material, and the code will be made publicly available.

4. Experiments

We conduct experiments on three text-to-scene tasks: (I) constructing abstract scenes of clip arts in the Abstract Scenes [41] dataset; (II) predicting semantic object layouts of real images in the COCO [22] dataset; and (III) generating synthetic image composites in the COCO [22] dataset.

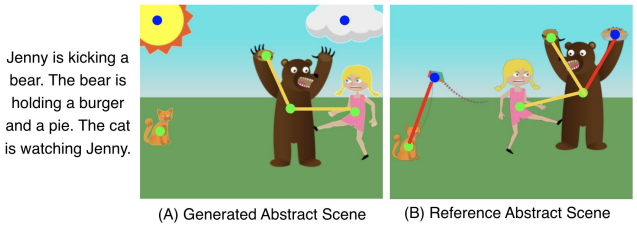

Figure 4. Evaluation metrics for the abstract scene generation task (best viewed in color): the green dots show the common `U-obj` between the references (B) and the generated abstract scene (A), the blue dots show the missing and mispredicted objects. Similarly, the yellow lines show the common `B-obj` and the red lines show the missing and mispredicted `B-obj`. The `U-obj` precision/recall for this example is 0.667/0.667, the `B-obj` precision/recall is 1.0/0.5.

Task (I): Clip-art Generation on Abstract Scenes We use the dataset introduced by [41], which contains over 1,000 sets of 10 semantically similar scenes of children playing outside. The scenes are composed with 58 clip art objects. The attributes we consider for each clip art object are their locations, sizes ($|R^{size}| = 3$), and the directions the object is facing ($|R^{direction}| = 2$). For the person objects, we also explicitly model the pose ($|R^{pose}| = 7$) and expression ($|R^{expression}| = 5$). There are three sentences describing different aspects of a scene. After filtering empty scenes, we obtain 9997 samples. Following [41], we reserve 1000 samples as the test set and 497 samples for validation.

Task (II): Semantic Layout Generation on COCO The semantic layouts contain bounding boxes of the objects from 80 object categories defined in the COCO [22] dataset. We use the val2017 split as our test set and use 5000 samples from the train2017 split for validation. We normalize the bounding boxes and order the objects from bottom to top as the y-coordinates typically indicate the distances between the objects and the camera. We further order the objects with the same y-coordinate based on their x-coordinates (from left to right) and categorical indices. The attributes that we consider are locations, sizes($|R^{size}| = 17$), and aspect ratios ($|R^{aratio}| = 17$). For the size attribute, we estimate the size range of the bounding boxes in the training split, and discretize it evenly into 17 scales. We also use 17 aspect ratio scales, which are $\{\frac{1}{i+1}\}_{i=1}^{8}$ and $\{i+1\}_{i=0}^{8}$.

Task (III): Synthetic Image Generation on COCO We demonstrate our approach by generating synthetic image composites given input captions in COCO [22]. For fair comparisons with alternative approaches, we use the

Methods	B1	B2	B3	B4	METEOR	ROUGE_L	CIDEr	SPICE
Captioning from True Layout [37]	0.678	0.492	0.348	0.248	0.227	0.495	0.838	0.160
Text2Scene (w/o attention)	0.591	0.391	0.254	0.169	0.179	0.430	0.531	0.110
Text2Scene (w object attention)	0.591	0.391	0.256	0.171	0.179	0.430	0.524	0.110
Text2Scene (w both attentions)	0.600	0.401	0.263	0.175	0.182	0.436	0.555	0.114
Text2Scene (full)	**0.615**	**0.415**	**0.275**	**0.185**	**0.189**	**0.446**	**0.601**	**0.123**

Table 3. Quantitative evaluation on the layout generation task.

Methods	IS	B1	B2	B3	B4	METEOR	ROUGE_L	CIDEr	SPICE
Real image	36.00±0.7	0.730	0.563	0.428	0.327	0.262	0.545	1.012	0.188
GAN-INT-CLS [29]	7.88±0.07	0.470	0.253	0.136	0.077	0.122	–	0.160	–
SG2IM* [15]	6.7±0.1	0.504	0.294	0.178	0.116	0.141	0.373	0.289	0.070
StackGAN [38]	10.62±0.19	0.486	0.278	0.166	0.106	0.130	0.360	0.216	0.057
HDGAN [39]	11.86±0.18	0.489	0.284	0.173	0.112	0.132	0.363	0.225	0.060
Hong et al [13]	11.46±0.09	0.541	0.332	0.199	0.122	0.154	–	0.367	–
AttnGan [36]	**25.89±0.47**	**0.640**	**0.455**	**0.324**	**0.235**	**0.213**	**0.474**	**0.693**	**0.141**
Text2Scene (w/o inpainting)	22.33±1.58	0.602	0.412	0.288	0.207	0.196	0.448	0.624	0.126
Text2Scene (w inpainting)	24.77±1.59	0.614	0.426	0.300	0.218	0.201	0.457	0.656	0.130

Table 4. Quantitative evaluation on the synthetic image generation task. *The result of SG2IM is evaluated on the validation set defined in [15], which is a subset of the COCO val2014 split.

	Ratio
Text2Scene > SG2IM [15]	0.7672
Text2Scene > HDGAN [39]	0.8692
Text2Scene > AttnGAN [36]	0.7588

Table 5. Human Evaluation on the synthetic image generation task.

val2014 split as our test set and use 5000 samples from the train2014 split for validation. We collect segmented object and stuff patches from the training split. The stuff segments are extracted from the training images by taking connected components in corresponding semantic label maps from the COCO-Stuff annotations [12]. For object segments, we use all 80 categories defined in COCO. For stuff segments, we use the 15 super-categories defined in [12] as the class labels, which results in 95 categories in total. We order the patches as in the layout generation experiment but when composing the patches, we always render the object patches in front of the stuff patches. The embedding vector for patch retrieval in our experiment has a dimension of 128.

4.1. Evaluation

Automatic Metrics Our tasks pose new challenges on evaluating the models as (1) the three types of scene representations are quite different from each other; and (2) there is no absolute one-to-one correspondence between text and scenes. For the abstract scene generation task, we draw inspiration from the evaluation metrics applied in machine translation [20] but we aim at aligning multimodal visual-linguistic data instead. To this end, we propose to compute the following metrics: precision/recall on single objects (`U-obj`), "bigram" object pairs (`B-obj`); classification accuracies for poses, expressions; Euclidean distances (defined as a Gaussian function with a kernel size of 0.2) for normalized coordinates of `U-obj` and `B-obj`. A "bigram" object pair is defined as a pair of objects with overlapping bounding boxes as illustrated in Figure 4.

In the layout generation experiment, it is harder to define evaluation metrics given the complexity of real world object layouts. Inspired by [13], we employ caption generation as an extrinsic evaluation. We generate captions from the semantic layouts using [37] and compare them back to the original captions used to generate the scene. We use commonly used metrics for captioning such as BLEU [26], METEOR [3], ROUGE_L [21], CIDEr [32] and SPICE [1].

For synthetic image generation, we adopt the Inception Score (IS) [30] metric which is commonly used in recent text to image generation methods. However, as the IS metric does not evaluate the semantic matches between images and captions, we also employ the caption generation metric as an extrinsic evaluation, as in [13]. Following [13], we use the Show-and-Tell caption generator [34] to generate sentences from the synthesized images.

Baselines For abstract scene generation, we compare our approach with [42]. Unlike synthetic image generation, there are less alternative approaches for the abstract scene and layout generation tasks. Therefore, we also consider variants of our full model: (1) Text2Scene (w/o attention): a model without any attention module. In particular, we replace Eq. 3 with a pure average pooling operation, discard c_t^o in Eq. 5, discard c_t^a in Eq. 8 and replace u_t^a with h_t^D. (2) Text2Scene (w object attention): a model with attention

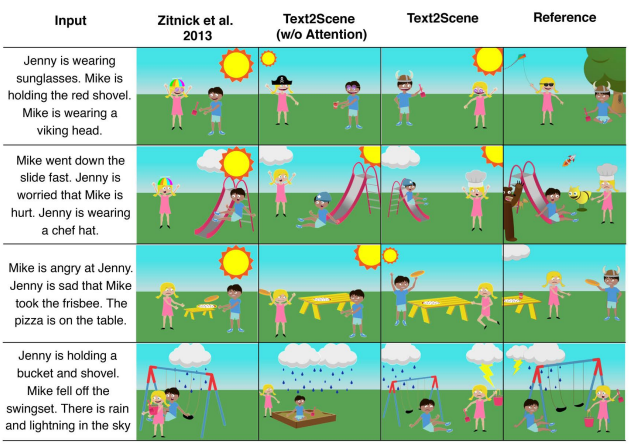

Figure 5. Examples of generated abstract scenes from textual descriptions. Please zoom in for details. We include more examples in the supplemental material.

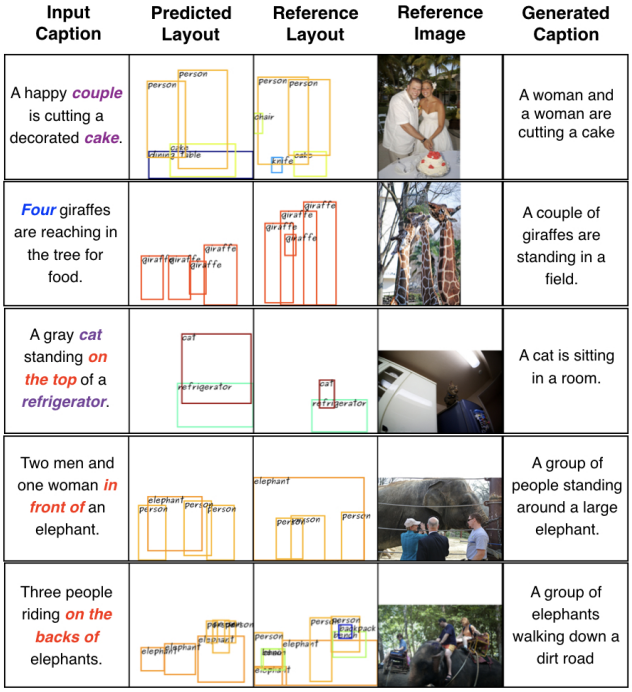

Figure 6. Examples of the generated layouts of COCO images from captions and generated captions from layouts (best viewed in color). Our model manages to learn the presence (first and second rows, purple text) and count (third and fourth rows, blue text) of the objects, and their spatial relations (fifth and sixth rows, red text). We include more examples in the supplemental material.

modules for object prediction but no dedicated attention for attribute prediction. Specifically, we replace (u_t^a, c_t^a) with (h_t^D, c_t^o) in Eq. 8. (3) Text2Scene (w both attentions): a model with dedicated attention modules for both object and attribute predictions but no regularization.

Human Evaluations Given the challenges of automatically measuring the relevance between generated scenes and input captions, we also conduct human evaluations via Amazon Mechanical Turk (AMT). For the Abstract Scene dataset, we collect human evaluations on 100 groups of clip-art scenes generated from sentences randomly sampled from the test set. Each group consists of three scenes generated by different models, and the ground truth reference scene. The human annotators are asked to determine whether a sentence is entailed given a corresponding clip-art scene. Each scene in this dataset is associated with three sentences that are used as the statements. Each sentence-scene pair is reviewed by three annotators to determine if the entailment is `true`, `false` or `uncertain`. We report results based on the ratio of the sentence-scene pairs marked as `true`.

To further analyze if our approach could capture finer-grained semantic alignments between textual descriptions and generated abstract scenes, we apply the predicate-argument semantic frame analysis of [4] on the corresponding triplets obtained from input sentences using a semantic parsing method as computed by [42]. We subdivide each sentence by the structure in the triplet as: `sub-pred` corresponding to sentences referring to only one object, `sub-pred-obj` corresponding to sentences referring to object pairs with semantic relations, `pred:loc` corresponding to sentences referring to locations, and `pred:pa` corresponding to sentences mentioning facial expressions.

For synthetic image generation, following [28], we conduct blind randomized A/B tests with human annotators on AMT. We compare our method with three state-of-the-art approaches, SG2IM [15], HDGAN [39] and AttnGAN [36]. We resize our generated images to the same resolutions as in these alternative methods, 64 $\times$ 64 for SG2IM [15], 256 $\times$ 256 for HDGAN [39] and AttnGAN [36]. For each text randomly selected from the test set, we present images generated from our method and a competing method and allow the user to choose the one which better represents the text. We collect results for 500 sentences, using 5 different annotators.

4.2. Results and Discussion

Abstract Scenes: Table 1 shows quantitative results on Abstract Scenes. Text2Scene (full) shows significant improvement over [42] and our variant that does not use attention on all the metrics except `U-obj Coord`. Human evaluation results on Table 2 confirm the quality of our outputs, where `Scores` are the percentage of scene-textual pairs with a `true` entailment; ($\geq$ 1) ($\geq$ 2) indicate if our method produced scenes that entailed at least one (or two) out of three statements. Text2Scene (full) outperforms the no-attention variant and the previous work under these experiments, including on statements with specific semantic information

Figure 7. Qualitative examples of the synthetic image composites. Please zoom in for details. We include more examples in the supplementary material.

such as `Obj-single`, `Obj-pair`, and `expression`, and is comparable on `location` statements. We also perform human evaluations on the reference scenes provided in the Abstract Scenes dataset as an upper bound on this task. Results also show that it is more challenging to generate the semantically related object pairs. Overall, the results also suggest that our proposed metrics correlate with human judgment on the task.

Figure 5 shows qualitative examples of our models on Abstract Scenes in comparison with baseline approaches and the reference scenes. These examples illustrate that Text2Scene is able to capture semantic nuances such as the spatial relation between two objects (e.g., the bucket and shovel are correctly placed in Jenny's hands in the last row) and object locations (e.g., Mike is on the ground near the swing set in the last row).

Extrinsic evaluation on the layout generation task (Table 4) shows the captions generated from our synthetic images perform close to those from the ground-truth layouts. Qualitative results in Figure 6 also show that our model learns important visual concepts such as the presence and count of the objects, and their spatial relations.

Synthetic Image Composites: Evaluation using automatic metrics shows that Text2Scene without any post-processing already outperforms all of the state-of-the-art methods by large margins except AttnGAN [36]. As our model adopts a composite image generation approach without adversarial training, gaps between adjacent patches may result in unnaturally shaded areas, which could hurt in some metrics. We observe that, after performing a regression-based inpainting [28], the composite outputs achieve consistent improvements on all the automatic metric scores. We posit that our model can be further improved by incorporating more robust post-processing or in combination with GAN-based methods. On the other hand, human evaluations show that our method significantly outperforms the alternative approaches including AttnGAN [36], demonstrating the potential of leveraging realistic image patches for text-to-image generation. In addition, as our model contains a patch retrieval module, it is important that the model does not generate a synthetic image by simply retrieving patches from a single training image. On average, each composite

image in our results contains 8.15 patches from 7.38 different source images, demonstrating that the model does not simply learn a global image retrieval. Fig. 7 shows qualitative examples of synthetic image composites. We also include examples of the generated images, their corresponding source images from which the patch segments are retrieved, and more qualitative results in the supplemental material.

5. Conclusions

This work presents a novel sequence-to-sequence model for generating compositional scene representations from visually descriptive language. We provide extensive quantitative and qualitative analysis of our model for different scene generation tasks on two distinctive datasets: Abstract Scenes [42] and COCO [22]. Experimental results demonstrate the capacity of our model to capture finer semantic meaning from descriptive text to generate complex scenes.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision (ECCV)*, pages 382–398. Springer, 2016. 6

[2] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Guided open vocabulary image captioning with constrained beam search. *Empirical Methods in Natural Language Processing (EMNLP)*, 2017. 2

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 6

[4] X. Carreras and L. Màrquez. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning*, pages 152–164. Association for Computational Linguistics, 2005. 7

[5] Q. Chen and V. Koltun. Image-to-image translation with conditional adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[6] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *European Conference on Computer Vision (ECCV)*, pages 15–29. Springer, 2010. 2

[7] J. P. Florian Schroff, Dmitry Kalenichenko. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4

[8] D. F. Fouhey and C. L. Zitnick. Predicting object dynamics in scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 1

[10] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem, and A. Kembhavi. Imagine this! scripts to compositions to videos. *European Conference on Computer Vision (ECCV)*, 2018. 3

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2016. 11, 12

[12] J. U. Holger Caesar and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[13] S. Hong, D. Yang, J. Choi, and H. Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3, 6

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[15] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3, 6, 7, 11, 13

[16] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015. 2

[17] J.-H. Kim, D. Parikh, D. Batra, B.-T. Zhang, and Y. Tian. Codraw: Visual dialog for collaborative drawing. *arXiv preprint arXiv:1712.05558*, 2017. 3

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015. 12

[19] P. Kuznetsova, V. Ordonez, T. Berg, and Y. Choi. Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association of Computational Linguistics*, 2(1):351–362, 2014. 2

[20] A. Lavie and A. Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. 6

[21] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004. 6

[22] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. *European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 5, 9

[23] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421. Association for Computational Linguistics, 2015. 4

[24] R. Mason and E. Charniak. Nonparametric method for data-driven image captioning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 592–598, 2014. 2

[25] V. Ordonez, X. Han, P. Kuznetsova, G. Kulkarni, M. Mitchell, K. Yamaguchi, K. Stratos, A. Goyal, J. Dodge, A. Mensch, et al. Large scale retrieval and generation of image descriptions. *International Journal of Computer Vision*, 119(1):46–59, 2016. 2

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 6

[27] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 11

[28] X. Qi, Q. Chen, J. Jia, and V. Koltun. Semi-parametric image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5, 7, 8, 12

[29] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1, 2, 6

[30] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 6

[31] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 3

[32] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015. 6

[33] R. Vedantam, X. Lin, T. Batra, C. Lawrence Zitnick, and D. Parikh. Learning common sense through visual abstraction. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2

[34] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 6

[35] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057. PMLR, 07–09 Jul 2015. 2, 4

[36] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 6, 7, 8, 13

[37] X. Yin and V. Ordonez. Obj2text: Generating visually descriptive language from object layouts. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2017. 6

[38] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 6

[39] Z. Zhang, Y. Xie, and L. Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 6, 7, 13

[40] X. Zhu and E. Grefenstette. Deep learning for semantic composition. In *ACL tutorial*, 2017. 2

[41] C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 5

[42] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the visual interpretation of sentences. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 2, 6, 7, 9

[43] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen. Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 18–26, 2015. 11

Supplementary Material

A. Network Architecture

Here we describe the network architectures for the components of our model in different tasks.

A.1. Text Encoder

We use the same text encoder for all our experiments, which consists of a bidirectional recurrent network with Gated Recurrent Units (GRUs). It takes a liner embedding of each word token as input and has a hidden dimension of 256 for each direction. We initialize the word embedding network with the pre-trained parameters from [27]. The parameters of this embedding network are kept fixed for the Abstract Scene and layout generation but finetuned for the synthetic image generation model.

A.2. Scene Encoder

The scene encoder Ω for the Abstract Scene generation is a pre-trained ResNet-34 [11]. Its parameters are fixed for all the experiments on the Abstract Scene dataset. For the layout and synthetic image generations, we develop our own scene encoders as the inputs at each step for these tasks are not RGB color images.

Table 6 and 7 show the architecture details. Here $|\mathcal{V}|$ is the size of the categorical vocabulary. In the layout generation task, $|\mathcal{V}|$ is 83, including 80 object categories in COCO and three special categorical tokens: sos, eos, pad. Here sos and eos indicate the start and end points of the sequence generation procedure. pad is a token for padding. For synthetic image generation, $|\mathcal{V}|$ is 98, including 80 object categories, 15 supercategories for stuffs in COCO and the special categorical tokens: sos, eos, pad.

As described in the main text of the paper, the input for synthetic image generation has a layer-wise structure where every three channels contain the color patches of a specific category from the background canvas image. In this case, the categorical information of the color patches can be easily learned. On the other hand, since the input is a large but sparse volume with very few non-zero values, to reduce the number of parameters and memory usage, we use a depthwise separable convolution as the first layer of Ω (index (2)), where each group of three channels (g3) is convolved to one single channel in the output feature map.

A.3. Convolutional Recurrent Module

The scene recurrent module for all our experiments is a convolutional GRU [43]. Each convolutional layer in this module have a 3×3 kernel with a stride of 1 and a hidden dimension of 512. We pad the input of each convolution so that the output feature map has the same spatial resolution as the input. The hidden state is initialized by spatially replicating the last hidden state from the text encoder.

Index	Input	Operation	Output Shape
(1)	-	Input	$\|\mathcal{V}\| \times 64 \times 64$
(2)	(1)	Conv(7×7, $\|\mathcal{V}\| \rightarrow 128$, s2)	$128 \times 32 \times 32$
(3)	(2)	Residual($128 \rightarrow 128$, s1)	$128 \times 32 \times 32$
(4)	(3)	Residual($128 \rightarrow 256$, s2)	$256 \times 16 \times 16$
(5)	(4)	Bilateral upsampling	$256 \times 28 \times 28$

Table 6. Architecture of our scene encoder Ω for layout generation. We follow the notation format used in [15]. Here $|\mathcal{V}|$ is the size of the categorical vocabulary. The input and output of each layer have a shape of $C \times H \times W$, where C is the number of channels and H and W are the height and width. The notation Conv($K \times K$, $C_{in} \rightarrow C_{out}$) represents a convolutional layer with $K \times K$ kernels, C_{in} input channels and C_{out} output channels. The notation s2 means the convolutional layer has a stride of 2. The notation Residual($C_{in} \rightarrow C_{out}$) is a residual module consisting of two 3×3 convolutions and a skip-connection layer. In the first residual block (index (3)), the skip-connection is an identity function and the first convolution has a stride of 1 (s1). In the second residual block (index (4)), the skip-connection is a 1×1 convolution with a stride of 2 (s2) and the first convolution also has a stride of 2 to downsample the feature map. Here all the convolutional layers are followed by a ReLU activation.

Index	Input	Operation	Output Shape
(1)	-	Input	$3\|\mathcal{V}\| \times 128 \times 128$
(2)	(1)	Conv(7×7, $3\|\mathcal{V}\| \rightarrow \|\mathcal{V}\|$, s2, g3)	$\|\mathcal{V}\| \times 64 \times 64$
(3)	(2)	Residual($\|\mathcal{V}\| \rightarrow \|\mathcal{V}\|$, s1)	$\|\mathcal{V}\| \times 64 \times 64$
(4)	(3)	Residual($\|\mathcal{V}\| \rightarrow 2\|\mathcal{V}\|$, s1)	$2\|\mathcal{V}\| \times 64 \times 64$
(5)	(4)	Residual($2\|\mathcal{V}\| \rightarrow 2\|\mathcal{V}\|$, s1)	$2\|\mathcal{V}\| \times 64 \times 64$
(6)	(5)	Residual($2\|\mathcal{V}\| \rightarrow 3\|\mathcal{V}\|$, s2)	$3\|\mathcal{V}\| \times 32 \times 32$
(7)	(6)	Residual($3\|\mathcal{V}\| \rightarrow 3\|\mathcal{V}\|$, s1)	$3\|\mathcal{V}\| \times 32 \times 32$
(8)	(7)	Residual($3\|\mathcal{V}\| \rightarrow 4\|\mathcal{V}\|$, s1)	$4\|\mathcal{V}\| \times 32 \times 32$

Table 7. Architecture of our scene encoder Ω for synthetic image generation. The notations are in the same format of Table 6. The first convolution (index (2)) is a depthwise separable convolution where each group of three channels (g3) is convolved to one single channel in the output feature map. All the convolutional layers are followed by a LeakyReLU activation with a negative slope of 0.2.

A.4. Object and Attribute Decoders

Table 8 shows the architectures for our object and attribute decoders. Ψ^o and Ψ^a are the spatial attention modules consisting of two convolutions. Θ^o is a two-layer perceptron predicting the likelihood of the next object using a softmax function. Θ^a is a four-layer convolution predicting the likelihoods of the location and attributes of the object. As explained in the main text of the paper, the output of Θ^a has $1 + \sum_k |R^k|$ channels, where $|R^k|$ denotes the discretized range of the k-th attribute, or the dimension of the query vector for patch retrieval in the synthetic image composites task. The first channel of the output from Θ^a predicts the location likelihoods which are normalized over the

Module	Index	Input	Operation	Output Shape
Ψ^o	(1)	-	Conv(3×3, 512→256)	256 × H × W
	(2)	(1)	Conv(3×3, 256→1)	1 × H × W
Ψ^a	(1)	-	Conv(3×3, 1324→256)	256 ×H × W
	(2)	(1)	Conv(3×3, 256→1)	1 × H × W
Θ^o	(1)	-	Linear((1324 + $\|\mathcal{V}\|$)→512)	512
	(2)	(1)	Linear(512→ $\|\mathcal{V}\|$)	$\|\mathcal{V}\|$
Θ^a	(1)	-	Conv(3×3, (1324+$\|\mathcal{V}\|$)→512)	512 × H × W
	(2)	(1)	Conv(3×3, 512→256)	256 × H × W
	(3)	(2)	Conv(3×3, 256→256)	256 × H × W
	(4)	(3)	Conv(3×3, 256→(1 + $\sum_k \|R^k\|$))	(1 + $\sum_k \|R^k\|$) × H × W

Table 8. Architectures for the object and attribute decoders. The notation Linear($C_{in} \rightarrow C_{out}$) represents a fully connected layer with C_{in} input channels and C_{out} output channels. All layers, except the last layer of each module, are followed by a ReLU activation.

spatial domain using a softmax function. The rest channels predict the attributes for every grid location. During training, the likelihoods from the ground-truth locations are used to compute the loss. At each step of the test time, the top-1 location is first sampled from the model. The attributes are then sampled from the corresponding location.

These architecture designs are used for all the three generation tasks. The only difference is the grid resolution (H, W). For abstract scene and layout generations, (H, W) = (28, 28). For synthetic image generation, (H, W) = (32, 32). Note that, although our model uses a fixed grid resolution, the composition can be performed on canvases of different resolutions.

A.5. Foreground Patch Embedding

The foreground segment representation we use is similar with the one in [28], where each segment P is represented by a tuple (P^{color}, P^{mask}, $P^{context}$). Here $P^{color} \in \mathbb{R}^{3\times H\times W}$ is a color patch containing the segment, $P^{mask} \in \{0,1\}^{1\times H\times W}$ is a binary mask indicating the foreground region of P^{color}, $P^{context} \in \{0,1\}^{|\mathcal{V}|\times H\times W}$ is a semantic map representing the semantic context around P within a bounding box, obtained from the semantic label map that originally contains the segment. The bounding box that encloses the context region is obtained by computing the bounding box of P^{color} and enlarging it by 50% in each direction.

Table 9 shows the architecture of our foreground patch embedding network. Here, the concatenation of (P^{color}, P^{mask}, $P^{context}$) is fed into a five-layer convolutional network which reduces the input into a 1D feature vector F_s (index (7)). As this convolutional backbone is relatively shallow, F_s is expected to encode the shape, appearance, and context, but may not capture the fine-grained semantic attributes of P. In our experiments, we find that incorporating the knowledge from the pre-trained deep features of P^{color} can help retrieve segments associated with strong semantics, such as the "person" segments. Therefore, we also use the pre-trained features F_d (index (8)) of P^{color} from

Index	Input	Operation	Output Shape
(1)	-	Input layout	($\|\mathcal{V}\|$ + 4) × 64 × 64
(2)	(1)	Conv(2 × 2, ($\|\mathcal{V}\|$ + 4) → 256, s2)	256 × 32 × 32
(3)	(2)	Conv(2 × 2, 256 → 256, s2)	256 × 16 × 16
(4)	(3)	Conv(2 × 2, 256 → 256, s2)	256 × 8 × 8
(5)	(4)	Conv(2 × 2, 256 → 256, s2)	256 × 4 × 4
(6)	(5)	Conv(2 × 2, 256 → 128, s2)	256 × 2 × 2
(7)	(6)	Global average pooling	256
(8)	-	Input patch feature	2048
(9)	(7)(8)	Linear((256 + 2048) → 128)	128

Table 9. Architecture of our foreground patch embedding network for synthetic image generation. All the convolutional layers are followed by a LeakyReLU activation with a negative slope of 0.2.

the mean pooling layer of ResNet152 [11], which has 2048 features. The final vector F_t is predicted from the concatenation of (F_s, F_d) by a linear mapping.

A.6. Inpainting Network

Our inpainting network has the same architecture as the image synthesis module proposed in [28], except that we exclude all the layer-normalization layers. To generate the simulated canvases on COCO, we follow the procedures proposed in [28], but make minor modifications: (1) we use the trained embedding patch features to retrieve alternative segments to stencil the canvas, instead of the intersection-over-union based criterion used in [28]. (2) we do not perform boundary elision for the segments as it may remove fine grained details of the segments such as human faces.

B. Optimization

For optimization we use Adam [18] with an initial learning rate of $5e-5$. The learning rate is decayed by 0.8 every 3 epochs. Models are trained until validation errors stop decreasing. For the Abstract Scene generation task, we set the hyperparameters (w_o, w_l, w_{pose}, $w_{expression}$, w_{size}, $w_{direction}$, w_a^O, w_a^A) to (8,2,2,2,1,1,1,1). For the layout generation task, we set the hyperparameters (w_o, w_l, w_{size}, w_{aratio}, w_a^O, w_a^A) to (5,2,2,2,1,0). For the synthetic image generation task, we set the hyperparameters (w_o, w_l, w_{size}, w_{aratio}, w_a^O, w_a^A, w_e, α) to (5,2,2,2,1,0,10,0.5). The hyperparameters are chosen to make the losses of different components comparable. Exploration of the best hyperparameters is left for future work.

C. User Study

We conduct two user studies on Amazon Mechanical Turk (AMT).

The first user study is to evaluate if the generated clip art scenes match the input textual descriptions. To this end, we randomly select 100 groups of images generated from the sentences in the test set. Each group consists of three images generated by different models, and the ground truth

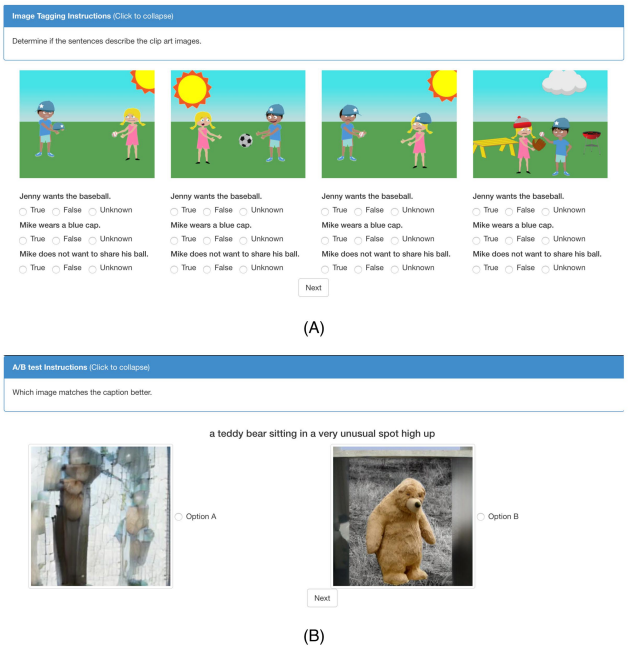

Figure 8. Screen shots of the user interfaces for our human subject studies on Amazon Mechanical Turk. (A) User interface for the evaluation study of the abstract scene generation experiment; (B) User interface for the evaluation study of the synthetic image generation experiment.

reference image. During the study, these images and the corresponding sentences are presented in random orders. The human annotators are asked to determine if the entailment between the generated scene and the sentence is `true`, `false` or `uncertain`. Each group of images is seen by three annotators. We ignore the `uncertain` responses and report the results using majority opinions. Figure 8 (A) shows the user interface of this study.

The second user study is on the synthetic image generation task, where we compare the generated images from our model and three state-of-the-art approaches:SG2IM [15], HDGAN [39], and AttnGAN [36]. In each round of the study, the human annotator is presented with one sentence and two generated images: one from our model, the other from an alternative approach. The orders of the images are randomized. We ask the human annotator to select the image which matches the sentence better. In total, we collect results for 500 sentences randomly selected from the test set, using five annotators for each. Figure 8 (B) shows the user interface of this study.

D. More qualitative examples

D.1. Abstract Scene

We present more qualitative examples in the Abstract Scene dataset in Fig. 9. The examples show that our model does not simply replicate the ground truth reference scenes, but generates dramatically different clip art scenes which still match the input textual descriptions.

D.2. Layout Generation

We present more qualitative examples for layout generation in Fig. 10. The examples include various scenes containing different object categories. Our model manages to learn important semantic concepts from the language, such as the presence and count of the objects, and their spatial relations.

D.3. Synthetic Image Generation

To demonstrate our model does not learn an image-level retrieval on the training set, we present in Fig. 11 the generated images and the corresponding source images from which the patch segments are retrieved for compositing. For each generated image, we show three source images for clarity. The examples illustrate that our model learns not only the presence and spatial layout of objects, but also the semantic knowledge that helps retrieve segments in similar contexts.

Fig. 12 shows more qualitative examples of our model for synthetic image generation.

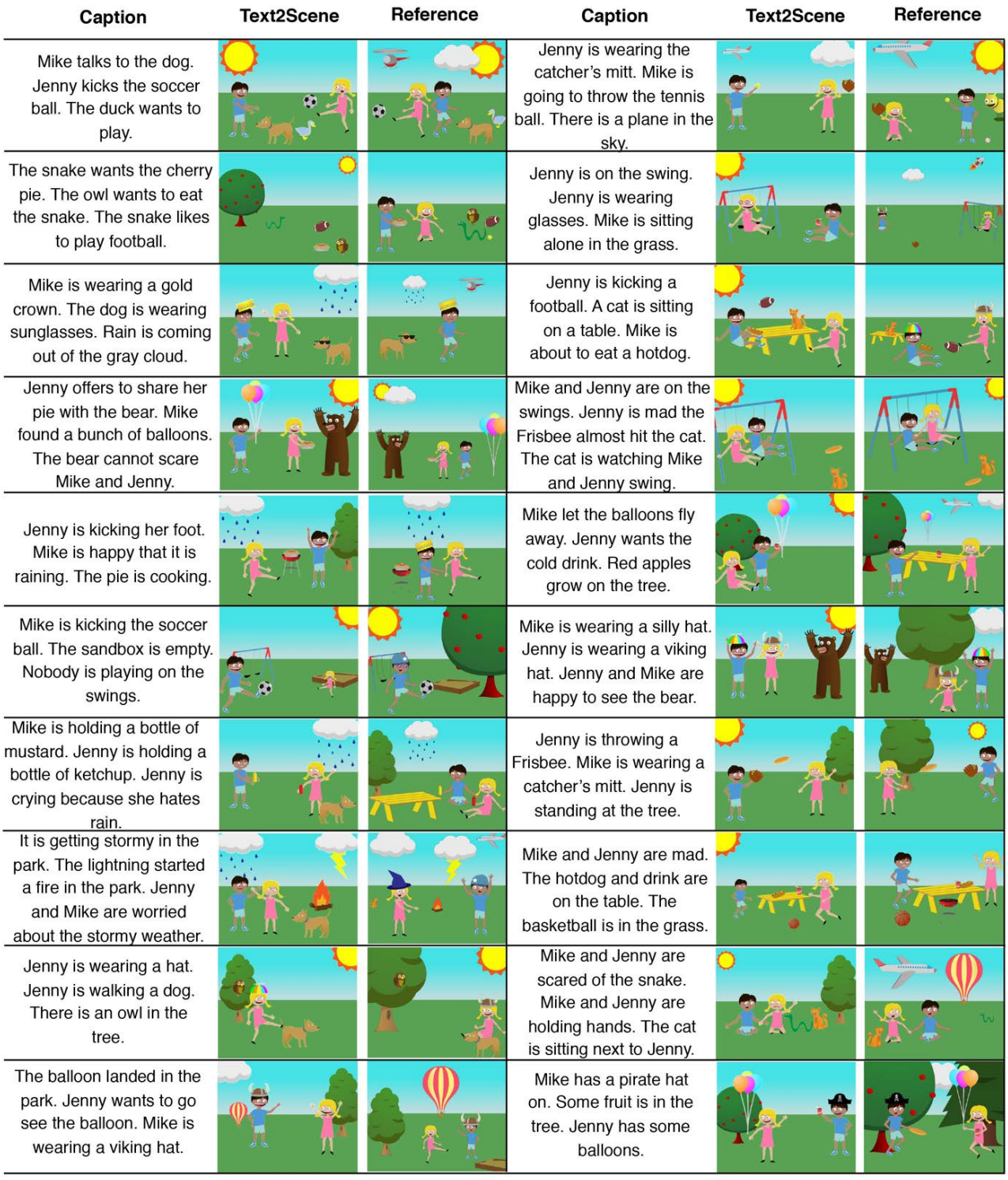

Figure 9. More qualitative examples of the abstract scene generation experiment.

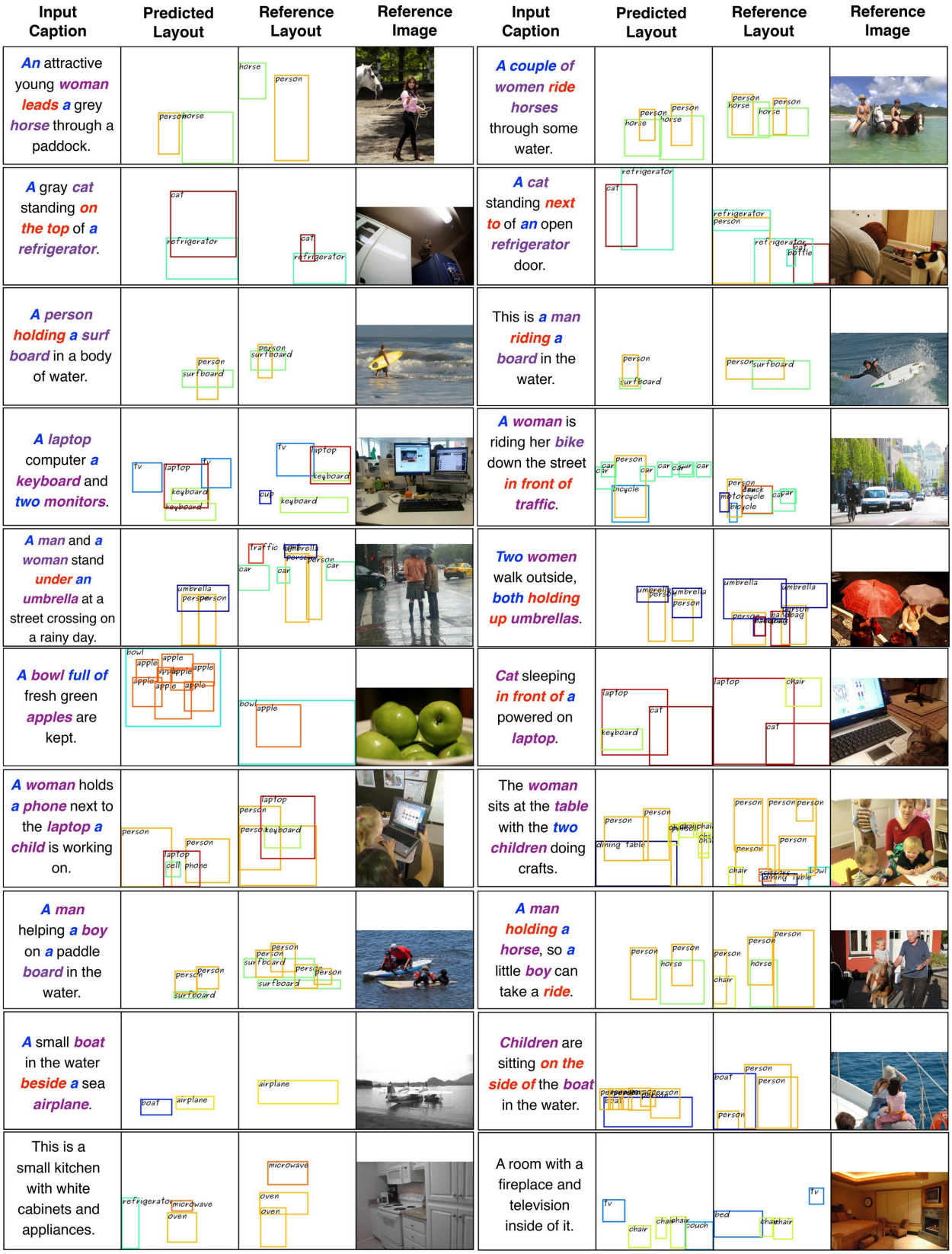

Figure 10. More qualitative examples of the layout generation experiment (best viewed in color). The appearances (purple), counts (blue), and spatial relations (red) of the objects are highlighted in the captions. The last row shows the cases when the layouts are underspecified in the input captions.

Figure 11. Example synthetic images and the source images from which the patch segments are retrieved for compositing. For each synthetic image, we show three source images for clarity.

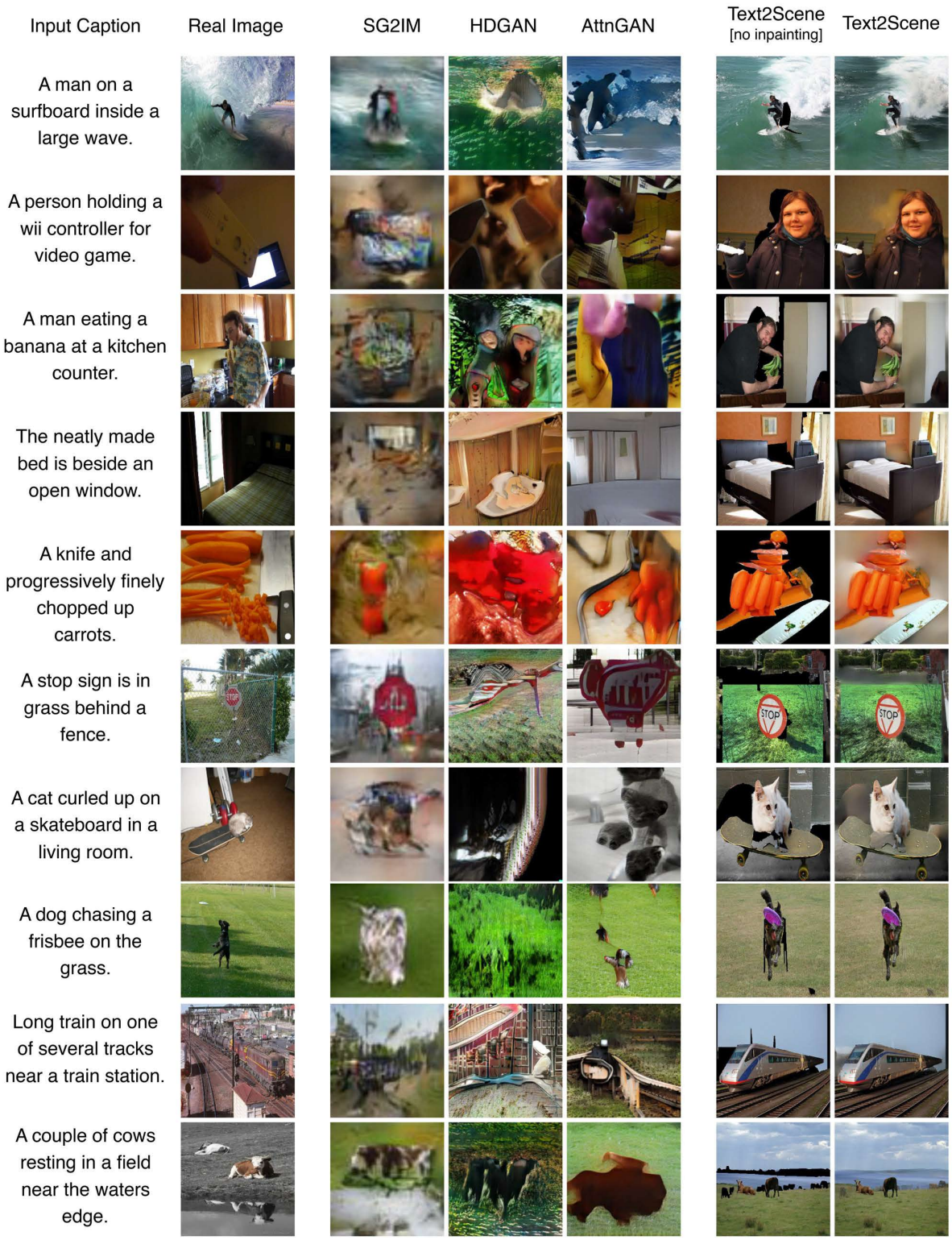

Figure 12. More qualitative examples of the synthetic image generation experiment.