Tell, Draw, and Repeat: Generating and modifying images based on continual linguistic instruction

Alaaeldin El-Nouby[1,4,*] Shikhar Sharma[2] Hannes Schulz[2] Devon Hjelm[2,3,5]
Layla El Asri[2] Samira Ebrahimi Kahou[2] Yoshua Bengio[3,5,6] Graham W. Taylor[1,4,6]

[1] University of Guelph [2] Microsoft Research [3] Montreal Institute for Learning Algorithms
[4] Vector Institute for Artificial Intelligence [5] University of Montreal [6] Canadian Institute for Advanced Research

Abstract

Conditional text-to-image generation is an active area of research, with many possible applications. Existing research has primarily focused on generating a single image from available conditioning information in one step. One practical extension beyond one-step generation is a system that generates an image iteratively, conditioned on ongoing linguistic input or feedback. This is significantly more challenging than one-step generation tasks, as such a system must understand the contents of its generated images with respect to the feedback history, the current feedback, as well as the interactions among concepts present in the feedback history. In this work, we present a recurrent image generation model which takes into account both the generated output up to the current step as well as all past instructions for generation. We show that our model is able to generate the background, add new objects, and apply simple transformations to existing objects. We believe our approach is an important step toward interactive generation.

1. Introduction

Vision is one of the most important ways in which humans experience, interact with, understand, and learn about the world around them. Intelligent systems that can generate images and video for human users have a wide range of applications, from education and entertainment to the pursuit of creative arts. Such systems also have the potential to serve as accessibility tools for the physically impaired; many modern and creative works are now generated or edited using digital graphic design tools, and the complexity of these tools can lead to inaccessibility issues, particularly with people with insufficient technical knowledge or resources. A system that can follow speech- or text-based instructions and then perform a corresponding image editing task could improve accessibility substantially. These benefits can easily extend to other domains of image generation such as gaming, animation, creating visual teaching material, etc. In this paper, we take a step in this exciting research direction by introducing the *neural visual artist* task.

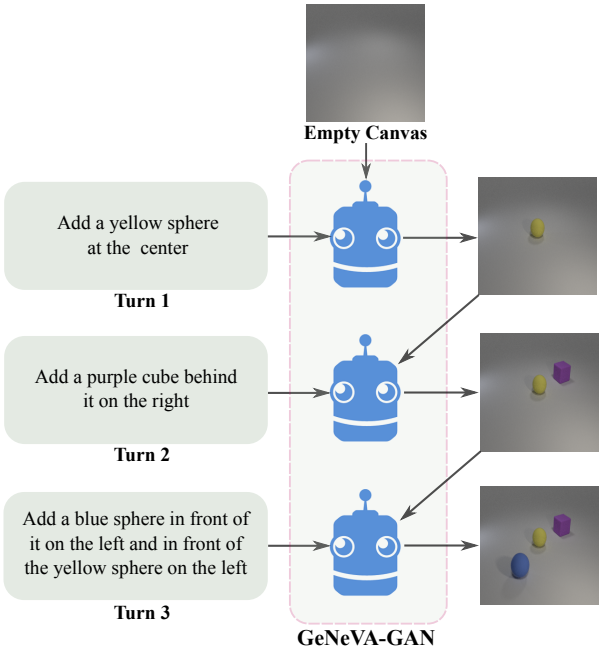


Figure 1. We present the Generative Neural Visual Artist (GeNeVA) task. Starting from an empty canvas, a *Drawer* (GeNeVA-GAN) iteratively constructs a scene based on a series of instructions and feedback from a *Teller*.

Conditional generative models allow for generation of images from other input sources, such as labels [1] and dialogue [2]. Image generation conditioned on natural language is a difficult yet attractive goal [3, 4, 5, 6]. Though these models are able to produce high quality images for simple datasets, such as birds, flowers, furniture, etc., good caption-conditioned generators of complex datasets, such as Microsoft Common Objects in Context (MS COCO) [7] are nonexistent. This lack of good generators may be due to the limited information content of captions, which are not rich enough to describe an entire image [2]. Combining object annotations with the intermediate steps of generating bounding boxes and object masks before generating the final images can improve results [5].

*Work was performed during an internship with Microsoft Research.

Instead of constructing images given a caption, we focus on learning to *iteratively* generate images based on continual linguistic input. We call this task the Generative Neural Visual Artist (GeNeVA), inspired by the process of gradually transforming a blank canvas to a scene. Systems trained to perform this task should be able to leverage advances in text-conditioned single image generation.

1.1. GeNeVA Task and Datasets

We present an example dialogue for the GeNeVA task in Figure 1, which involves a Teller giving a sequence of linguistic instructions to a Drawer for the ultimate goal of image generation. The Teller is able to gauge progress through visual feedback of the generated image. This is a challenging task because the Drawer needs to learn how to map complex linguistic instructions to realistic objects on a canvas, maintaining not only object properties but relationships between objects (e.g., relative location). The Drawer also needs to modify the existing drawing in a manner consistent with previous images and instructions, so it needs to remember previous instructions. All of these involve understanding a complex relationship between objects in the scene and how those relationships are expressed in the image in a way that is consistent with all instructions given.

For this task, we use the synthetic Collaborative Drawing (CoDraw) dataset [8], which is composed of sequences of images along with associated dialogue of instructions and linguistic feedback (Figure 2). Also, we introduce the Iterative CLEVR (i-CLEVR) dataset (Figure 4), a modified version of the Compositional Language and Elementary Visual Reasoning (CLEVR) [9] dataset, for incremental construction of CLEVR scenes based on linguistic instructions. Offloading the difficulty of generating natural images by using two well-studied synthetic datasets allowed us to better assess progress on the GeNeVA task and improve the iterative generation process. While photo-realistic images will undoubtedly be more challenging to work with, our models are by no means restricted to synthetic image generation. We expect that insights drawn from this setting will be crucial to success in the natural image setting.

The most similar task to GeNeVA is the task proposed by the CoDraw [8] authors. They require a model to build a scene by placing the clip art images of the individual objects in their correct positions. In other words, the model predictions will be in coordinate space for their task, while for a model for the GeNeVA task they will be in pixel space. Natural images are in scope for the GeNeVA task, where Generative Adversial Networks (GANs) are currently state-of-the-art. Non-pixel-based approaches will be limited to placing pre-segmented specific poses of objects. For such approaches, it will be extremely difficult to obtain a pre-segmented set of all possible poses of all objects e.g., under different lighting conditions. Additionally, a pixel-based model does not necessarily require object-labels so it can easily scale without such annotation.

1.2. Contributions

Our primary contributions are summarized as follows:

- We introduce the GeNeVA task and propose a novel recurrent GAN architecture that specializes in plausible modification of images in the context of an instructional history.
- We introduce the i-CLEVR dataset, a sequential version of CLEVR [9] with associated linguistic descriptions for constructing each CLEVR scene, and establish a baseline for it.
- We propose a relationship similarity metric that evaluates the model's ability to place objects in a plausible position compatible with the instructions.
- We demonstrate the importance of iterative generation for complex scenes by showing that our approach outperforms the non-iterative baseline.

Our experiments on the CoDraw and i-CLEVR datasets show that our model is capable of generating images that incrementally build upon the previously generated images and follow the provided instructions. The model is able to learn complex behaviors such as drawing new objects, moving objects around in the image, and re-sizing these objects. In addition to reporting qualitative results, we train an object localizer and measure precision, recall, F1 score, and our proposed relational similarity metric by comparing detections on ground-truth vs. generated images.

2. Related Work

GANs [10] represent a powerful family of generative models whose benefits and strengths extend to conditional image generation. Several approaches for conditioning exist, such as conditioning both the generator and discriminator on labels [1], as well as training an auxiliary classifier as part of the discriminator [11]. Closer to GeNeVA text-based conditioning, Reed et al. [3] generate images conditioned on the provided captions. Zhang et al. [4] proposed a two-stage model called StackGAN, where the first stage generated low resolution images conditioned on the caption, and the second stage generated a higher resolution image conditioned on the previous image and the caption. Hong et al. [5] proposed a three-step generation process where they use external segmentation and bounding box annotation for MS COCO to first generate bounding boxes, then a mask for the object, and then the final image. Building upon StackGAN, AttnGAN [6] introduced an attentional generator network that enabled the generator to synthesize different spatial locations in the image, conditioned on an attention mechanism over words in the caption. It also introduced an image-text similarity module which encouraged generating images more relevant to the provided caption.

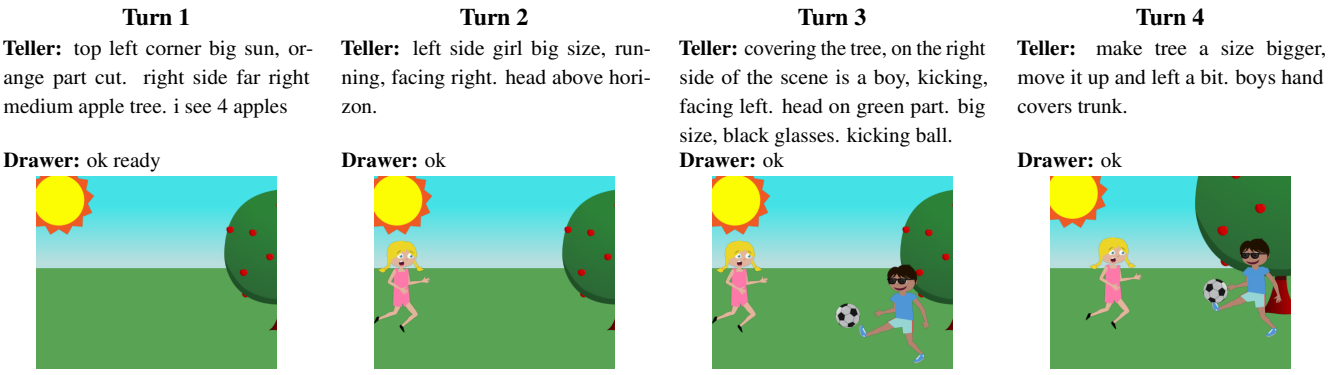


Figure 2. An example from the CoDraw [8] dataset. Each example from the dataset involves a conversation between a Teller and a Drawer. The Teller has access to a final image and has to iteratively provide text instructions and feedback to the Drawer to guide them to draw the same image. The Drawer updates the image on receiving instructions or feedback. In the original CoDraw setup, the Drawer predicted the position and attributes of objects to compose a scene. In GeNeVA, we task systems with generating the images directly in pixel space.

Departing from purely caption data, Sharma et al. [2] proposed a non-iterative model called ChatPainter that generates images using dialogue data. ChatPainter conditions on captions from MS COCO and a Recurrent Neural Network (RNN)-encoded dialogue relevant to the caption (obtained from the Visual Dialog (VisDial) [12] dataset) to generate images. The authors showed that the question answering-based dialogue captured richer information about the image than just the caption, which enabled ChatPainter to generate superior images compared to using captions alone. Since the VisDial dialogues were collected separately from the MS COCO dataset, there are no intermediate incremental images for each turn of the dialogue. The model, thus, only reads the entire dialogue and generates a single final image, so this setup diverges from a real-life sketch artist scenario where the artist has to keep making changes to the current sketch based on feedback.

There has also been recent work in performing recurrent image generation outside of text-to-image generation tasks. Yang et al. [13] perform unsupervised image generation in recursive steps, first generating a background, subsequently conditioning on it to generate the foreground and the mask, and finally using an affine transformation to combine the foreground and background. Lin et al. [14] tackle the image compositing task of placing a foreground object on a background image in a natural location. However, this approach is limited to fixed object templates, and instead of generating images directly, the model recursively generates parameters of transformations to continue applying to an object template until the image is close enough to natural image manifold. Their approach also does not modify existing objects in the image. Both of these approaches aim to generate a single final image without incorporating any external feedback. To the best of our knowledge, the proposed model is the first of its kind that can recursively generate and modify intermediate images based on continual text instructions such that every generated image is consistent with past instructions.

3. Methods

In this section, we describe a conditional recurrent GAN model for the GeNeVA task. An overview of the model architecture is shown in Figure 3.

3.1. Model

During an n-step interaction between a Teller and a Drawer, the Teller provides a drawing canvas x_0 and a sequence of instructions $Q = (q_1, \ldots, q_n)$. For every turn in the conversation, a conditioned generator G outputs a new image

$$\widetilde{x}_t = G(z_t, h_t, f_{G_{t-1}}), \tag{1}$$

where z_t is a noise vector sampled from a normal distribution $\mathcal{N}(0, 1)$ of dimension N_z. G is conditioned on two variables, h_t and $f_{G_{t-1}}$, where h_t is a context-aware condition and $f_{G_{t-1}}$ is context-free.

The context-free condition $f_{G_{t-1}} = E_G(\widetilde{x}_{t-1})$ is an encoding of the previously generated image $\widetilde{x}_{t-1}$ using an encoder E_G, which is a shallow Convolutional Neural Network (CNN). Assuming square inputs, the encoder produces low resolution feature maps of dimensions $(K_g \times K_g \times N_g)$.

The context-aware condition h_t needs to have access to the conversation history such that it can learn a better encoding of the instruction in the context of the conversation history up to time $t-1$.

Each instruction q_t is encoded using a bi-directional Gated Recurrent Unit (GRU) on top of GloVe word embeddings [15]. This instruction encoding is denoted by d_t.

We formulate h_t as a recursive function R, which takes the instruction encoding d_t as well as the previous condition h_{t-1} as inputs. We implement R with a second GRU, which yields h_t with dimension N_c:

$$h_t = R(d_t, h_{t-1}). \tag{2}$$

The context-free condition $f_{G_{t-1}}$ represents the prior given to the model by the most recently generated image (i.e. a representation of the current canvas). On the other

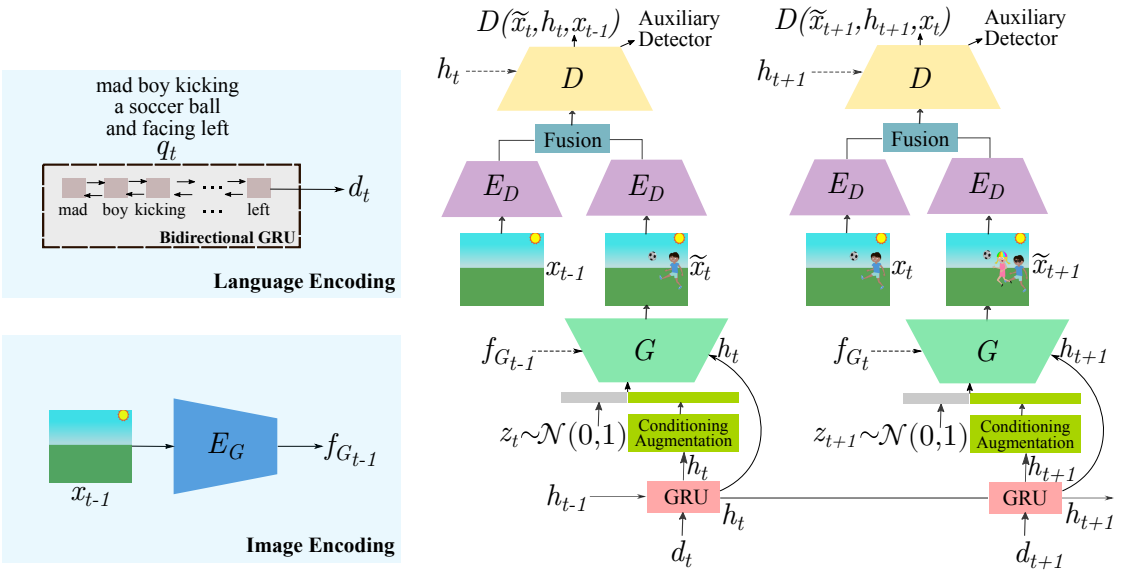


Figure 3. An overview of the GeNeVA-GAN architecture. For each time step t, the instruction q_t is encoded into d_t using a Bi-directional GRU. The previous time-step generated image $\tilde{x}_{t-1}$ (teacher-forcing at training time with ground truth x_{t-1}) is encoded into $f_{G_{t-1}}$ using E_G. A GRU outputs a context-aware condition h_t as a function of d_t and the previous condition h_{t-1}. The generator G generates an image $\tilde{x}_t$ conditioned on h_t and $f_{G_{t-1}}$. $f_{G_{t-1}}$ is concatenated to feature maps from G with the same spatial dimensions while h_t is used as the input for conditional batch normalization. The image from the current time-step (ground truth x_t or generated $\tilde{x}_t$) and the previous time-step ground-truth image are encoded using E_D. The features from both images are fused and then passed as input to a discriminator D. Finally, D is conditioned using the context-aware condition h_t. An auxiliary objective of detecting all the objects in the scene is also added to D.

hand, the context-aware condition h_t represents the additions or modification the Teller is describing in the new image. In our model, the context-aware condition is concatenated with the noise vector z_t after applying conditioning augmentation [4], as shown in Figure 3. Similar to Miyato and Koyama [16], it is also used in applying conditional batch normalization to all of the generator's convolutional layers. The context-free condition $f_{G_{t-1}}$ is concatenated with the feature maps from the generator's intermediate layer L_{f_G} which has the same spatial dimensions as $f_{G_{t-1}}$.

Since we are modeling iterative modifications of images, having a discriminator D that only distinguishes between real and generated images at each step will not be sufficient. The discriminator should also identify cases where the image is modified incorrectly with respect to the instruction or not modified at all. To enforce this, we introduce three modifications to the discriminator. First, an image encoder E_D is used to encode the current time step image (real or generated) *and the previous time-step ground-truth image* as shown in Figure 3. The output feature maps of dimensions $(K_d \times K_d \times N_d)$ are passed through a fusion layer. We experiment with element-wise subtraction and concatenation of feature maps as different options for fusion. The fused features are passed through a discriminator D. Passing a fused representation of both the current and the previous images to the discriminator encourages it to focus on the quality of the additions and modifications, not only the overall image quality. This provides a better training signal for the generator. Additionally, the context-aware condition h_{t-1} is used as a condition for D through projection similar to [16].

Second, for the discriminator loss, in addition to labelling real images as positive examples and generated images as negative examples, we add a term for the combination of [real image, wrong instruction], similar to Reed et al. [3]. Finally, we add an auxiliary objective [11] of detecting all objects in the scene at the current time step.

The generator and discriminator are trained alternately to minimize the adversarial hinge loss [17, 18, 19]. The discriminator minimizes

$$L_D = L_{D_{\text{real}}} + \frac{1}{2}(L_{D_{\text{fake}}} + L_{D_{\text{wrong}}}) + \beta L_{\text{aux}}, \qquad (3)$$

where

$$L_{D_{\text{real}}} = -\mathbb{E}_{(x_t, c_t) \sim p_{\text{data}(0:T)}}[\min(0, -1 + D(x_t, c_t))]$$
$$L_{D_{\text{wrong}}} = -\mathbb{E}_{(x_t, \hat{c_t}) \sim p_{\text{data}(0:T)}}[\min(0, -1 - D(x_t, \hat{c_t}))]$$
$$L_{D_{\text{fake}}} = -\mathbb{E}_{z_t \sim \mathcal{N}, c_t \sim p_{\text{data}(0:T)}}[\min(0, -1 - D(G(z_t, \tilde{c}_t), c_t))],$$

with $\tilde{c}_t = \{h_t, f_{G_{t-1}}\}$ and $c_t = \{h_t, x_{t-1}\}$. Finally, $\hat{c}_t$ is the same as c_t but with a wrong instruction and T is the length of the instruction sequence Q.

The loss function for the auxiliary task is a binary cross entropy over all the N possible objects at that time step,

$$L_{\text{aux}} = \sum_{i=0}^{N} -\left(y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\right),$$

where y_i is a binary label for each object indicating whether it is present in the scene at the current time step. Note that

we do not index the loss with t to simplify notation. A linear layer of dimension N is added to the last discriminator layer before applying projection conditioning with h_t. A sigmoid function is applied to each of the N outputs yielding p_i, the model detection prediction for object i.

The generator loss term is

$$L_G = -\mathbb{E}_{z_t \sim p_z, c_t \sim p_{data}(0:T)} D(G(z_t, \tilde{c}_t), c_t) + \beta L_{\text{aux}}. \tag{4}$$

Additionally, to help with training stability, we apply zero-centered gradient penalty regularization to the discriminator's parameters Φ on the real data alone with weighting factor γ as suggested by Mescheder et al. [20],

$$\text{GPReg}(\Phi) = \frac{\gamma}{2} \mathbb{E}_{p_{D(x)}} [\|\nabla D_\Phi(x)\|^2]. \tag{5}$$

3.2. Implementation Details

The network architecture for the generator and discriminator follows the ResBlocks architecture as used by Miyato and Koyama [16]. Following SAGAN [19], we add a self-attention layer to the intermediate layers with spatial dimensions of 16×16 for the discriminator and the generator. We use spectral normalization [18] for all layers in the discriminator.

For the training dynamics, the generator and discriminator parameters are updated every time step, while the parameters of E_G, R and the text encoder are updated every sequence. The text encoder and the network R are trained with respect to the discriminator objective only.

We add layer normalization [21] to the text encoding GRU, as well as the the GRU implementing R. We add batch normalization [22] to the output of the image encoder E_G. We found that adding these normalization methods was important for gradient flow to all modalities.

For training, we used teacher forcing by using the ground truth images x_{t-1} instead of the generated image $\widetilde{x}_{t-1}$, but we use $\widetilde{x}_{t-1}$ during test time. We use the Adam optimizer [23] for the GAN, with learning rates of 0.0004 for the discriminator and the 0.0001 for the generator, trained with an equal number of updates. We use Adam as well for the text encoder with learning rate of 0.003, and for the GRU with learning rate of $3 \cdot 10^{-4}$.

In our experiments the following hyper-parameters worked the best, $N_z = 100$, $N_c = 1024$, $K_g = 16$, $N_g = 128$, $K_d = 16$, $N_d = 256$, $\gamma = 10$, and $\beta = 20$. More details are provided in the appendix.

4. Datasets

For the GeNeVA task, we require a dataset that contains textual instructions describing drawing actions, along with corresponding ground truth images for each instruction. To the best of our knowledge, the only such dataset publicly available is CoDraw. Additionally, we create a new dataset called i-CLEVR, specifically designed for this task.

4.1. CoDraw

CoDraw [8] is a recently released clip art-like dataset. It consists of scenes, which are sequences of images of children playing in a park. The children have different poses and expressions and the scenes include other objects such as trees, tables, and animals. There are 58 object types in total. Corresponding to every scene, there is a conversation between a Teller and a Drawer (both Amazon Mechanical Turk workers) in natural language. The Drawer updates the canvas based on the Teller's instructions. The Drawer can ask questions as well for clarification. The dataset consists of 9,993 scenes of varying length. An example of such a scene is shown in Figure 2. The initial drawing canvas x_0 for CoDraw provided to the Drawer consists of the background having just the sky and grass.

Pre-processing In some instances of the original dataset, the Drawer waited for multiple Teller turns before modifying the image. In these cases, we concatenate consecutive turns into a single turn until the Drawer modifies the image. We also concatenate turns until a new object has been added or removed. Thus every turn has an image in which the number of objects has changed since the last turn.

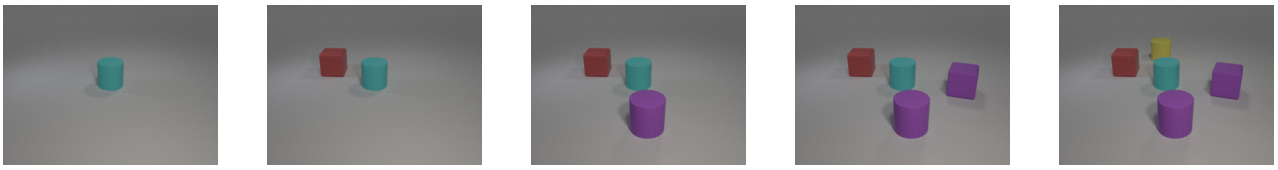
We treat the concatenated utterances of the Drawer and the Teller at time step t as the instruction, injecting a special delimiting token between the Teller and Drawer. The Teller and Drawer text contains several spelling mistakes and we run the Bing Spell Check API[1] over the entire dataset to make corrections. For words that are not present in the GloVe vocabulary, we use the "unk" word embedding from GloVe. We use the same train-valid-test split proposed in the original CoDraw dataset.

4.2. i-CLEVR

CLEVR [9] is a programmatically generated dataset that is popular in the Visual Question Answering (VQA) community. CLEVR consists of images of collections of objects with different shapes, colors, materials and sizes. Each image is assigned complex questions about object counts, attributes or existence. We build on top of the open-source generation code[2] for CLEVR to create Iterative CLEVR (i-CLEVR). Each example in the dataset consists of a sequence of 5 (image, instructions) pairs. Starting from an empty canvas (background), each instruction describes an object to add the canvas in terms of its shape and color. The instruction also describes where the object should be placed relative to existing objects in the scene. To make the task more complex and force the model to make use of context, we refer to the most recently added object by "it" instead of stating its attributes.

[1] https://azure.microsoft.com/en-us/services/cognitive-services/spell-check/

[2] https://github.com/facebookresearch/clevr-dataset-gen



Add a cyan cylinder at the center

Add a red cube behind it on the left

Add a purple cylinder in front of it on the right and in front of the cyan cylinder

Add a purple cube behind it on the right and in front of the red cube on the right

Add a yellow cylinder behind the purple cylinder on the left and behind the red cube on the right

Figure 4. Example sequence of image-instruction pairs from the i-CLEVR dataset.

An example from the i-CLEVR dataset is presented in Figure 4. The initial drawing canvas x_0 for i-CLEVR consists of the empty background. A model is tasked with learning how to add the object with the correct attributes in a plausible position, based on the textual instruction. More details about the dataset generation can be found in the appendix.

The i-CLEVR dataset consists of 10,000 sequences, totalling 50,000 images and instructions. The training split contains 6,000 sequences, while the validation and testing splits have 2,000 sequences each.

5. Experiments

In this section, we first define our evaluation metrics, and then describe the experiments carried out on the CoDraw and i-CLEVR datasets.

5.1. Evaluation Metrics

Standard metrics used for evaluating GAN models such as the Inception Score or Fréchet Inception Distance (FID) only capture how realistic the generations look relative to real images. They cannot detect if the model is correctly modifying the images according to the GeNeVA task instructions. A good evaluation metric for this task needs to identify if all the objects that were described by the Teller are present in the generated images. It should also check that the objects' positions and relationships match the instructions. To capture all of these constraints, we train an object localizer on the training dataset. For every example, we compare the detections of this localizer on the real images and the generated ones. We present the precision, recall, and F1-score for this object detection task. We also construct a graph where the nodes are objects present in the images and edges are positional relationships: left, right, behind, front. We compare the graphs constructed from the real and the generated images to test the correct placement of objects, without requiring the model to draw the objects in the same exact locations (which would have defied its generative nature).

The **object detector and localizer** is based on the Inception-v3 architecture. We modify the last layer for object detection and replace it with two heads. The first head is a linear layer with a sigmoid activation function to serve as the object detector. It is trained with a binary cross-entropy loss. The second head is a linear layer where we regress all the objects' coordinates. This head is trained with an L_2-loss with a mask applied to only compute loss over objects that occur in the ground truth image provided in the dataset. We initialize the model using pre-trained weights trained over the ILSVRC12 (ImageNet) dataset and fine-tune on the CoDraw or i-CLEVR datasets.

Relational Similarity To compare the arrangement of objects qualitatively, we use the above object detector/localizer to determine the type and position of objects in the ground truth and the generated image. We estimate a scene graph for each image, in which the detected objects and the image center are the vertices. The directed edges are given by the left-right and front-back relations between the vertices. To compute a relational similarity metric on scene graphs, we determine how many of the ground truth relations are present in the generated image:

$$\text{rsim}(E_{G_{\text{gt}}}, E_{G_{\text{gen}}}) = \text{recall} \times \frac{|E_{G_{\text{gen}}} \cap E_{G_{\text{gt}}}|}{|E_{G_{\text{gt}}}|} \quad (6)$$

where "recall" is the recall over objects detected in the generated image w.r.t objects detected in the ground truth image. $E_{G_{\text{gt}}}$ is the set of relational edges for the ground truth image corresponding to vertices common to both ground truth images and generated images, and $E_{G_{\text{gen}}}$ is the set of relational edges for the generated image corresponding to vertices common to both ground truth images and generated images. The graph similarity for the complete dataset is reported by taking the mean across time-steps of individual examples and then a mean over the entire dataset. This metric is a lower bound on the actual relational accuracy, as it penalizes relations based on how the objects are positioned in the ground truth image. The same instructions may, however, permit different relationship graphs. We present some examples of low-scoring to high-scoring images on this metric in the appendix.

5.2. Ablation Study

We experimented with different variations of our architecture to test the effect of each component. We define the different instantiations of our architecture as follows:

Model	CoDraw				i-CLEVR			
	Precision	Recall	F1-Score	rsim($E_{G_{\text{gt}}}, E_{G_{\text{gen}}}$)	Precision	Recall	F1-Score	rsim($E_{G_{\text{gt}}}, E_{G_{\text{gen}}}$)
Non-iterative	50.60	43.42	44.96	22.33	25.49	20.95	22.63	11.52
Baseline	55.61	42.31	48.05	25.31	69.09	56.38	62.08	45.19
Mismatch	62.47	48.95	54.89	32.74	71.15	60.57	65.44	50.21
G prior	60.78	49.37	54.48	33.60	82.80	77.22	79.91	63.93
Aux	54.78	51.51	53.10	33.83	83.63	75.63	79.43	55.36
D Concat	66.38	51.27	57.85	33.57	88.47	83.35	85.83	70.22
D Subtract	**66.64**	**52.66**	**58.83**	**35.41**	**92.39**	**84.72**	**88.39**	**74.02**

Table 1. Results of the GeNeVA-GAN ablation study on the CoDraw and i-CLEVR datasets.

Model	$L_{D_{\text{wrong}}}$	$f_{G_{t-1}}$	L_{aux}	D Fusion	
				concat	subtract
Baseline	✗	✗	✗	✗	✗
Mismatch	✓	✗	✗	✗	✗
G prior	✓	✓	✗	✗	✗
Aux	✓	✓	✓	✗	✗
D Concat	✓	✓	✓	✓	✗
D Subtract	✓	✓	✓	✗	✓

Table 2. Description of the components present in each model we test in the ablation study.

- **Baseline** The simplest version of our model. The discriminator loss only includes the adversarial terms L_{fake} and L_{real}. The generator is only conditioned using the context-aware condition: $\widetilde{x}_t = G(z, h_t)$.
  As for the discriminator, it has no access to the previous time-step image features. Only $\tilde{x}_t$ is encoded using E_D and then passed to the discriminator D without any fusion operations.
- **Mismatch** The L_{wrong} term is added to the discriminator loss. The rest of the model is similar to the baseline.
- ***G* prior** For this model, we condition the generator on the context-free condition $f_{G_{t-1}}$ in addition to h_t as in equation (1).
- **Aux** In this model we add the L_{aux} term to both the generator and discriminator losses. The loss functions for this model follow equations (3) and (4).
- ***D* Concat** In this model, the discriminator's input is the fused features from x_{t-1} and x_t (or $\tilde{x}_t$) encoded using E_D. The fusion is a simple concatenation across the channels dimension.
- ***D* Subtract** This is the same as "*D* Concat" except for the fusion operation, which is an element-wise subtraction between the feature maps.
- **Non-iterative** The non-iterative baseline uses the same model as the "Mismatch" baseline. All the input instructions are concatenated into one instruction and the final image is generated in a single-step.

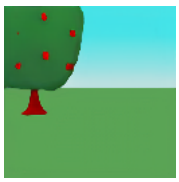
A summary of the components that are present for each model we test in the ablation study is provided in Table 2.

5.3. Results

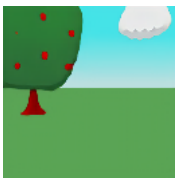
Quantitative Results We present the results of the ablation study in Table 1. As expected, among the iterative models, the *Baseline* model has the weakest performance on all the metrics for both datasets. This is due to the fact that it needs to construct a completely new image from scratch every time-step; there is no consistency enforced between successive generations. As for the *Mismatch* model, despite suffering from the same problems as the *Baseline*, training D to differentiate between wrong and right (image, instruction) pairs leads to generated images that better match the instructions. This is clear in Table 1 as the performance improves on all metrics compared to the *Baseline*.

The *G prior* model tries to enforce consistency between generations by using the context-free condition $f_{G_{t-1}}$. Adding this condition leads to a significant improvement to all the metrics for the i-CLEVR dataset. However, for the CoDraw dataset, it shows a less significant improvement to recall and relational similarity, while precision degrades. These results can be explained by the fact that i-CLEVR has much more complex relationships between objects and the instructions have a strong dependence on the existing objects in the scene. Therefore, the model benefits from having access to how the objects were placed in the most recent iteration. As for CoDraw, the relationships among objects are relatively simpler. Nevertheless, adding the context-aware condition helps with placing the objects correctly as shown by the improvement in the relational similarity metric. A possible drawback from using the context-free condition is that it is harder to recover from past mistakes, especially if it has to do with a large objects. This drawback can explain the drop in precision.

For the *Aux* model, it had different effects on the two datasets. For CoDraw, it helped improve recall and relational similarity, but caused a significant decrease in precision. For i-CLEVR, it helped improve precision with hurting the recall and relational similarity. This different behavior for each dataset can be explained by the types of objects that are present. While for CoDraw, there are objects that are almost always present like the girl or the boy, for i-CLEVR



Drawer: ready
Teller: large apple tree left side trunk start 2 3 way up green and about 1 and 1 4 inches from left side

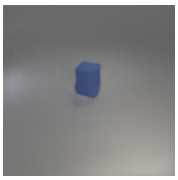


Teller: big cloud right side almost touching apple tree 1 2 inch up into blue

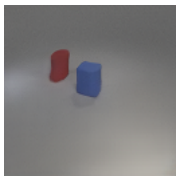
Teller: below the cloud full size girl her head touches top of green hands over head centered under cloud

Teller: boy 1 1 2 to left of girl under right side of tree same height as girl facing right hands out to right holding a football in both

Teller: sandbox medium size lower left corner facing right left side off screen lower right corner is equal to end of tree trunk
Drawer: yes



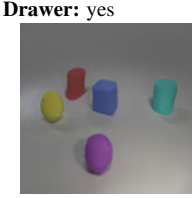
Add a blue cube at the center



Add a red cylinder behind it on the left

Add a purple sphere in front of it on the right and in front of the blue cube

Add a cyan cylinder behind it on the right and in front of the red cylinder on the right



Add a yellow sphere in front of the red cylinder on the left and on the left of the blue cube

Figure 5. Example images generated by our model (D Subtract) on CoDraw (top row) and i-CLEVR (bottom row); shown with the provided instructions. We scale images from both datasets to 128x128 in a pre-processing step.

there is high randomness in objects presence. Adding the auxiliary objective encourages the model to make sure the frequent objects are present, leading to the increase in recall while hurting precision. Finally, we observe that giving D access to the previous image x_{t-1} shows improvement on almost all the metrics for both datasets. We also observe that subtraction fusion consistently performs better than concatenation fusion and outperforms all other models for both datasets. This indicates that encouraging the discriminator to focus on the modifications gives a better training signal for the generator.

The *Non-iterative* model performs worse than all of the iterative models. This is likely because the language encoder has difficulty understanding dependencies and object relationships in a lengthy concatenated instruction. The benefit of using an iterative model is more visible in the i-CLEVR dataset since in it, the spatial relationships are always defined in terms of existing objects. This makes it very difficult to comprehend all the relationships across different turns in a single step. By having multiple steps, iterative generation makes this task easier. The results of this experiment make a case for iterative generation in complex text-conditional image generation tasks that have traditionally been performed non-iteratively.

Qualitative Results: We present some examples of images generated by our model in Figure 5. Due to space constraints, more example images are provided in the appendix. On CoDraw, we observe that the model is able to generate scenes consistent with the conversation and generation history and gets most of the coarse details correct, such as large objects and their relative positions. But it has difficulty in capturing fine-grained details, such as tiny objects, facial expressions, and object poses. The model also struggles when a single instruction asks to add several objects at once. For i-CLEVR, the model captures spatial relationships and colors very accurately as demonstrated in Figure 5. However, in some instances, the model fails to add the fifth object when the image is already crowded and there is no space left to add it without moving the others. We also experimented with using an intermediate ground truth image as the inital image at test time and the model was able to generalize and place objects correctly in that scenario as well. The results of this experiment are presented in the appendix.

6. Conclusion and Future Work

We presented a recurrent GAN model for the GeNeVA task and show that the model is able to draw reasonable images for the provided instructions iteratively. It also significantly outperforms the non-iterative baseline. We presented an ablation study to highlight the contribution of different components. Since this task can have several plausible solutions and no existing metric can capture all of them, we proposed a relational similarity metric to capture the possible relationships. For future research directions, having a system that can also ask questions from the user when it needs clarifications would potentially be even more useful. Collecting photo-realistic images, transitions between such images, and annotations in the form of instructions for these transitions is prohibitively expensive; hence, no photo-realistic dataset appropriate for this task publicly exists. Such datasets are needed to scale this task to photo-realistic images.

Acknowledgements

We thank Philip Bachman for valuable discussions.

References

[1] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv:1411.1784 [cs.AI], 2014.

[2] S. Sharma, D. Suhubdy, V. Michalski, S. E. Kahou, and Y. Bengio, "Chatpainter: Improving text to image generation using dialogue," in *International Conference on Learning Representations (ICLR) Workshop*, 2018.

[3] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International Conference on Machine Learning (ICML)*, 2016.

[4] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *International Conference on Computer Vision (ICCV)*, 2017.

[5] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical text-to-image synthesis," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

[8] J.-H. Kim, D. Parikh, D. Batra, B.-T. Zhang, and Y. Tian, "Codraw: Visual dialog for collaborative drawing," arXiv:1712.05558 [cs.CV], 2017.

[9] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "Inferring and executing programs for visual reasoning," in *International Conference on Computer Vision (ICCV)*, 2017.

[10] 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 27*, 2014.

[11] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *International Conference on Machine Learning (ICML)*, 2017.

[12] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual Dialog," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] J. Yang, A. Kannan, D. Batra, and D. Parikh, "LR-GAN: Layered recursive generative adversarial networks for image generation," in *International Conference on Learning Representations (ICLR)*, 2017.

[14] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "ST-GAN: Spatial transformer generative adversarial networks for image compositing," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[16] T. Miyato and M. Koyama, "cGANs with projection discriminator," in *International Conference on Learning Representations (ICLR)*, 2018.

[17] J. H. Lim and J. C. Ye, "Geometric GAN," arXiv:1705.02894 [stat.ML], 2017.

[18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[19] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," arXiv:1805.08318 [stat.ML], 2018.

[20] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?" in *International Conference on Machine learning (ICML)*, 2018.

[21] J. Ba, R. Kiros, and G. E. Hinton, "Layer normalization," arXiv:1607.06450 [stat.ML], 2016.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Journal of Machine Learning Research (JMLR)*, 2015.

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

[24] Blender Online Community, "Blender - a 3d modelling and rendering package," 2016. [Online]. Available: http://www.blender.org

Appendix

A. Generation Examples

We present selected examples generated using our best model (D Subtract) on two datasets. Examples generated for CoDraw are presented in Figure 6 and examples generated for i-CLEVR are presented in Figure 7. We also present random examples from all the models present in the ablation study for a qualitative comparison on the CoDraw dataset. These are shown in Figure 8 (Baseline), Figure 9 (Mismatch), Figure 10 (G prior), Figure 11 (Aux), Figure 12 (D Concat), Figure 13 (D Subtract), and Figure 14 (Non-iterative).

B. Generalization to new background images

GeNeVA-GAN was trained using the empty background image as the initial image. We ran an experiment where we used a different image (intermediate ground truth image from the test set containing objects) as the initial image. We present generated examples from this experiment in Figure 15. The model is able to place the desired object at the correct location with the correct color and shape over the provided image. This shows that the model is capable of generalizing to a background it was not trained on and it can understand the existing objects from just the initial image without any instruction history for placing them.

C. i-CLEVR Dataset Generation

To generate the image for each step in the sequence, an object with random attributes is rendered to the scene using Blender [24]. We ensure that all objects have a unique combination of attributes. Each object can have one of 3 shapes (cube, sphere, cylinder) and one of 8 colors. In contrast to CLEVR, we have a fixed material and size for objects. For the first image in the sequence, the object placement is fixed to the image center. For all the following images, the objects are placed in a random position while maintaining visibility (not completely occluded) and at a minimum distance from other objects.

To generate instructions, we use a simple text template that depends on the instruction number. For example, the second instruction in the sequence will have the following template:

"Add a [object color] [object shape] [relative position: depth] it on the [relative position: horizontal]"

From the third instruction onward, the object position is described relative to two objects. These two objects are chosen randomly from the existing objects in the scene.

D. Qualitative evaluation of the Relational Similarity (rsim) metric

We provide generated image examples with scores spread out between the minimum value (0) and maximum value (1) on the rsim metric in Figure 16. This is to provide readers with a more intuitive understanding of how the metric captures which spatial relationships match between the ground truth and the generated image.

E. Additional implementation details

We use 300-dimensional GloVe[3] word embeddings for representing the words in each instruction q_t. These word embeddings are encoded using a bi-directional-GRU to obtain a 1024-dimensional instruction encoding d_t. All state dimensions for the higher level GRU R are set to 1024. The output of the conditioning augmentation module is also 1024-dimensional.

The code for this project was implemented in PyTorch. For the generator and discriminator optimizers, "betas" was set to $(0.0, 0.9)$ and weight decay was set to 0. The learning rates for the image encoding modules were set to 0.006. Gradient norm was clipped at 50. For each training experiment, we used a batch size of 32 over 2 NVIDIA P100 GPUs.

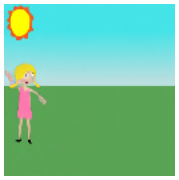
F. Additional language encoder experiments

We experimented with using skip-thought encoding for sentences instead of training the bi-directional-GRU encoder over GloVe embeddings. For the paper, we chose to use the latter since it performed better.

We also experimented with passing the previous image through the language encoder, but observed that it was easier for the model to generate an accurate image when the previous image features are passed to the Generator directly.

[3] http://nlp.stanford.edu/data/glove.840B.300d.zip

Drawer: ready to draw ? **Teller:** Medium sun is on the left corner fully visible.



Teller: Below sun sits a mad girl with legs on front she faces right and hand touches the left border a 1 2 head is above horizon. **Drawer:** ok.

Teller: The girl is big. A fire is on front feet of girl like 1 2 “. **Drawer:** ok.

Teller: A grill is just next to fire the grill is a little lower than top flame. **Drawer:** ok.

Teller: A small pine is on right side 1 4 “ left side is cut also the tip is cut. **Drawer:** ok

Teller: In top left hand corner is medium sun cut off on top and on side. **Drawer:** I am a patient worker ready to start.

Teller: In middle of screen is a medium pine tree trunk starts dead middle of screen. **Drawer:** Got it.

Teller: A large boy is sitting cross legged almost in left corner slightly higher and to right he is facing right . **Drawer:** ok.

Teller: Drink in right hand hot dog in left, to left of hot dog is a fire. **Drawer:** ok.

Teller: On right side is small girl angry face running her foot is cut off head just touches horizon. **Drawer:** ok.

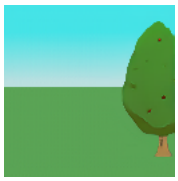
Teller: Small hot air balloon in upper left corner touching left edge very top of balloon off top edge. **Drawer:** ok.

Teller: Med tree with hole in trunk at right side about 1 3 of it below horizon and right edge of it off screen. **Drawer:** ok.

Teller: Big crying girl sitting on ground legs outstretched facing left top of her head touching bottom left corner of tree trunk. **Drawer:** ok.

Teller: Big standing boy arms in air facing right toes touching bottom edge of page slightly left of center. **Drawer:** ok.

Teller: Soccer ball on ground between boy and girl about level with boy ’s hips . **Drawer:** ok.



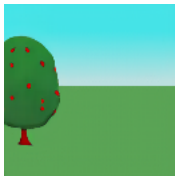
Drawer: Ready. **Teller:** Medium bushy tree right side top 2 3 above horizon right side touching right side of tree.

Teller: Left side medium slide facing right half ladder cut off at left one step is in sky. **Drawer:** Done.

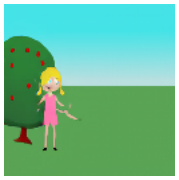
Teller: head covering bottom of slide part bum by bottom sitting legs out happy girl facing right. **Drawer:** Done.

Teller: By her back hand is a medium beach ball her hand is touching the right side of it from hand holding medium party ballo. **Drawer:** Done.

Teller: Actual balloons part is over horizon back of head touching left side of tree running happy boy facing girl wearing rainbow hat. **Drawer:** Done.



Drawer: Ready. **Teller:** On the left is a small apple tree slightly cut at the left edge. **Drawer:** Done.



Teller: In front of the tree facing right stands a happy girl waving **Drawer:** Done.

Teller: In the middle of the field standing facing right his hands to the right. **Drawer:** Who is standing in the middle and what expression?

Teller: Is a boy he is angry. **Drawer:** Done. **Teller:** On the right is a medium pine tree slightly cut at the right and at the top. **Drawer:** Done.

Teller: In front of the tree is a dog facing left. **Drawer:** Done.

Drawer: Start. **Teller:** Bushy tree in middle. **Drawer:** Ok.

Teller: Full sun right corner. **Drawer:** Yes.

Teller: Soccer ball bottom right corner. **Drawer:** Done.

Teller: Plane top left corner . **Drawer:** Ok.

Teller: Boy between sun and soccer ball. **Drawer:** Boy direction. **Teller:** Hands out facing left. **Drawer:** Ok.

Figure 6. Generation examples from our best model (D Subtract) for the CoDraw dataset.

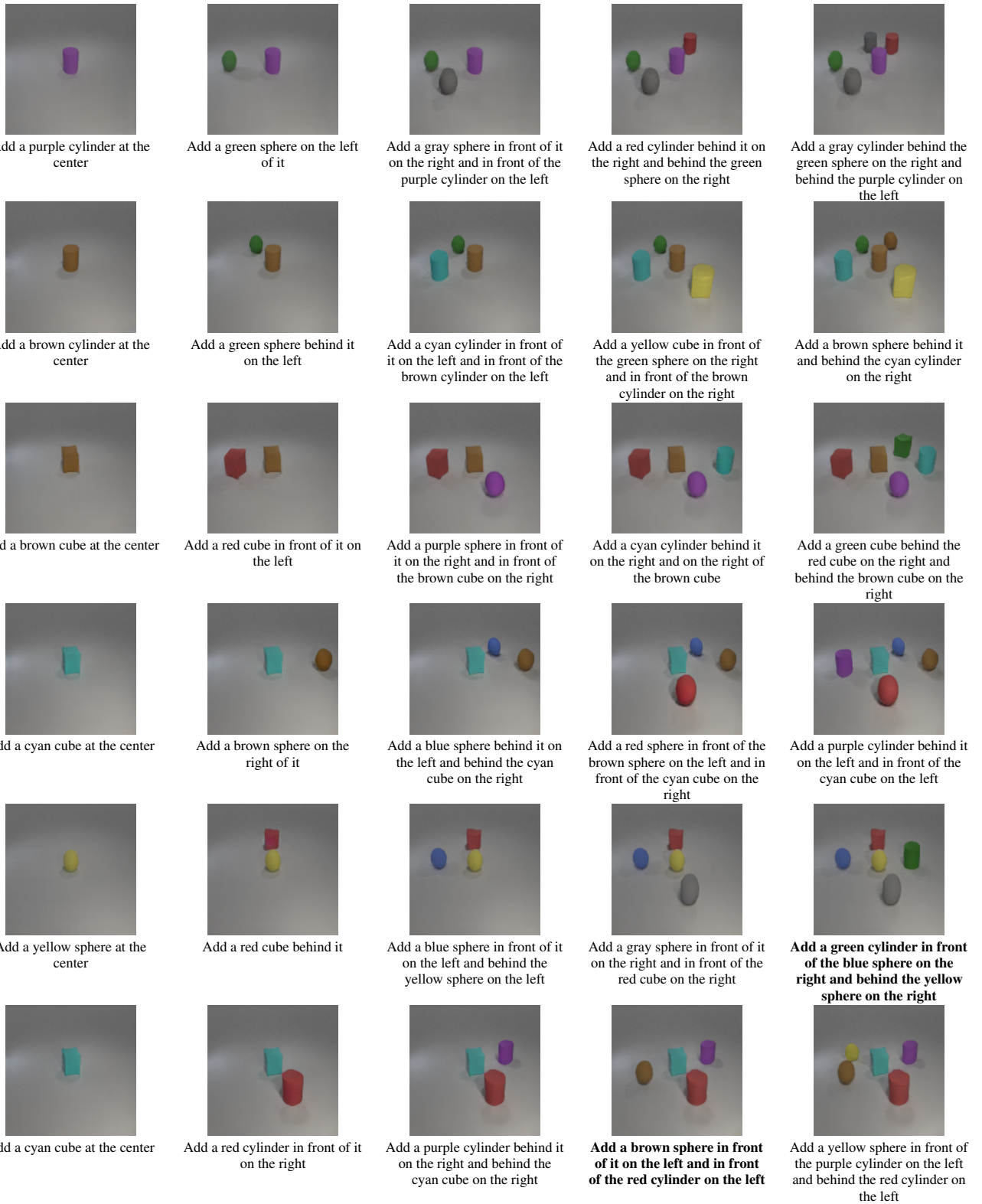
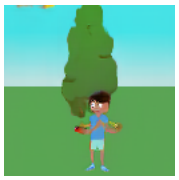


Figure 7. Generation examples from our best model (D Subtract) for the i-CLEVR dataset. Instructions where the model made a mistake are marked in bold.

Teller: small pine on right most of tree cut off on right and top big shocked running mike on right facing left his elbow on the left edge of tree

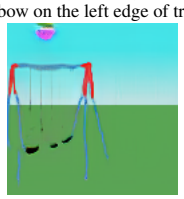


Teller: big sun in middle 1 2 cut off on top small oak tree on left hole facing right 1 4 from top

Teller: big slide on left facing right slide is in front of tree

Teller: big angry sitting with legs out jenny is on ground in front of end of slide

Teller: big cat is under slide mike 's head is over horizon jenny is 1 2 below horizon



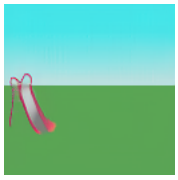
Drawer: ready when you are describe away **Teller:** large swing on the left facing right

Teller: looks like a happy girl standing on the 2nd swing facing right no teeth

Teller: sad facing right feet almost at the bottom an inch from the left **Drawer:** are his arms out or left one up

Teller: one arm in the air **Drawer:** got it **Teller:** there is a sun on the top right a little bit of it is seen it 's behind the tree

Teller: under the tree is a bee facing left **Drawer:** got it



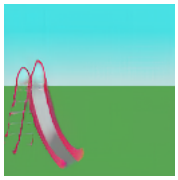
Teller: there is a small slide on the left side facing right tip of the top of the slide is above horizon

Teller: the boy with sad face is sliding down two legs kicked out there 's a baseball near the end of the slide **Drawer:** ready

Teller: the boy is probably medium size there is also a medium girl on the right edge smiling and jumping

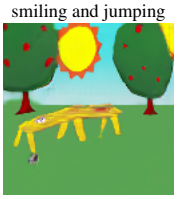
Teller: she is facing left and the top of her head is touching the horizon she has a baseball glove on her left hand

Teller: there is an airplane facing right at the top right of the picture let me know when you 're ready for me to check



Drawer: ready **Teller:** medium slide on left of screen facing right about 1 2 " from left surprised boy sitting at bottom of slide

Teller: small tree in back of slide to right a bit and then an apple tree to the right of that tree about 1 2 " apart



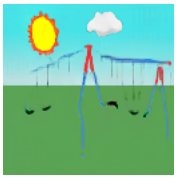
Teller: sun between the trees bottom left of sun blocked by tree on left **Drawer:** got it

Teller: surprised girl on right of screen about 1 2 inch from right border hands in the air facing left **Drawer:** got it

Teller: tennis ball beneath her left foot **Drawer:** got it

Teller: left side big swing set left behind leg cut off edge and left corner small sun covered partially by small plain cloud

Teller: right side medium sand basket dune facing left in front of dune sad cross leg sitting crying girl facing right **Drawer:** ready

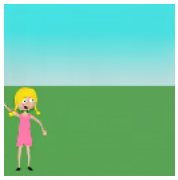


Teller: next to girl cat sitting looking at her her hand is hidden by cat 's head

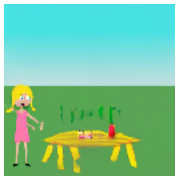
Teller: in front of the sandbox near the girl in the corner boy sad cross leg looking right

Teller: behind boy and behind girl 's hand spring bee body hidden by boy facing right and i will check

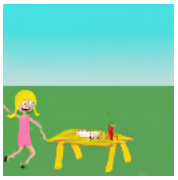
Figure 8. Random selection of examples generated by our Baseline model for the CoDraw dataset.



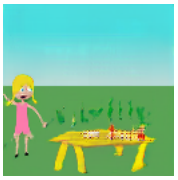
Drawer: ready **Teller:** 1 girl happy running facing right 0 2 inch from bottom to top and 0 2 inches from left to right with a chef



Teller: hat on her a table 1 and a half inches from left to right 1 2 inches from bottom to top with a pizza in the middle



Teller: and and a hot dog facing left to the right of the pizza a fire 0 1 inches from bottom to top 0 4 inches from right to left and above a



Teller: tent facing left and top of the tent is above the horizon line 0 1 inches and right is cover a little bit and above

Teller: a air balloon small like 1 2 inches from right to left and 0 2 inches from top to bottom and that 's it

Drawer: what 's in the picture **Teller:** in the top right is a sun covered partially by clouds **Drawer:** is it a large sun

Teller: in the middle left is a helicopter **Drawer:** which way is it facing

Teller: yes a large sun **Teller:** heli is facing to the right **Teller:** tail is to the left **Teller:** on bottom right is a girl in pink with left arm raised

Teller: on his mouth is an o shape **Teller:** to the right of the boy is a dog with a blue collar

Teller: above the boys left hand in the blue sky is a yellow ball

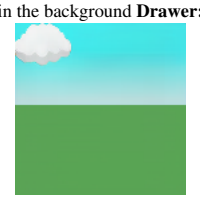
Drawer: may you please tell the first thing to draw **Teller:** there is a small tree on the right side of the scene sort of in the background **Drawer:**

Teller: there is a bear to the left but in the foreground of the tree **Drawer:** what next

Teller: the bear is small **Drawer:** **Teller:** the girl is to the left facing left looking at the bear with her leg out scared facing right **Drawer:**

Teller: there is a small dog below the girl and a angry boy to left facing left with a racket in the left hand **Drawer:** what is next

Teller: there is a small helicopter at the top in the sky in between the boy and girl **Drawer:** right above the bear



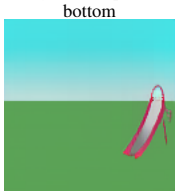
Teller: there is a cloud in the top left corner it is cut off on the top and left sides two puffs on the right and 3 puffs on the bottom

Teller: an inch from the right of the cloud are small balloons the orange balloon is on the right

Teller: airplane 1 4 inch to the right of the balloon the nose of the plane is in line with the yellow balloon it faces left

Teller: angry boy sitting below the plane facing left about 1 2 inch grass above and below him

Teller: a girl sits to the left of the boy facing him feet almost touching surprised wearing viking hat top brown part of hat touches horizon



Drawer: ready **Teller:** a small bear close to the right side one finger off picture small sliding on his left side bear left foot touching the sliding

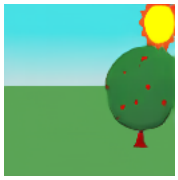
Teller: table medium in the center front

Teller: medium pine tree behind the table one inch

Teller: sad girl standing far left part hand missing sad face medium

Teller: boy sitting on her right side look mad

Figure 9. Random selection of examples generated by our Mismatch model for the CoDraw dataset.



Teller: medium sun on your right hand with a small half apple tree under it

Teller: small snake left of apple tree snake facing left

Teller: small bushy tree is on your left hand side hole facing left 1 2 inch from side 1 inch from top

Teller: large table in left hand corner slanted right north with medium owl on right end facing right

Teller: bushy tree looks like it 's sitting on table

Drawer: i 'm ready **Teller:** boy flying a kite

Teller: wearing blue t shirt **Teller:** blue shoes **Teller:** lighter blue shorts **Drawer:** got it **Teller:** has black hair **Drawer:** i have all of him **Teller:** sunny outside

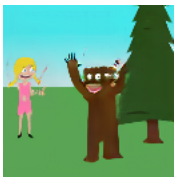
Teller: a girl next to a grill

Teller: i am sorry only one girl in the scene **Teller:** next to a girl **Teller:** on the grill **Drawer:** is a burger **Teller:** there are hamburgers

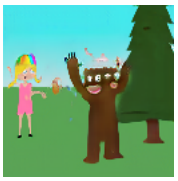
Teller: 3 of them **Teller:** she is wearing pink overall holding ketchup in her hand and she is wearing glasses on

Drawer: ready when you are **Teller:** right side medium pine tree cut in half by right edge and cut top **Drawer:** got it

Teller: left from pine tree big size bear facing left head above horizon



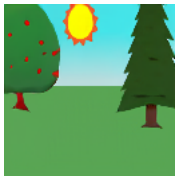
Teller: left side running angry big size girl facing right horn hat holding football in right hand our right



Teller: now behind girl they overlap running angry boy facing right purple glasses witch hat on facing right half body above horizon **Drawer:** got it

Teller: movie girl down a bit move boy right a bit lines with girl next to girls what is a frisbee

Drawer: what is in the scenery of the image **Teller:** medium pine tree on right trunk is about 1 inch from bottom and trunk is 1 2 from right top is cut off on top and right **Drawer:** what 's next



Teller: on left is medium apple tree cut off on left trunk is halfway down grass big sun in center of two trees **Drawer:** what 's next

Teller: on the right side of apple tree trunk is sad boy sitting legs out facing right hes 1 inch from bottom very close to trunk **Drawer:** what 's next

Teller: to right of boy align with his hand is a fire to right of fire is girl kneeling smiling one arm up girl and boy small **Drawer:** what 's next

Teller: girls right hand overlaps pine tree a little left side of sun is overlapped by apple tree below fire is ketchup left mustard right **Drawer:** next

Teller: ready **Drawer:** what what is the first object and location **Teller:** small table middle of green **Drawer:** next

Teller: there is a pine to the left med size upper peck can not see **Drawer:** next

Teller: med sun in upper right corner **Drawer:** any of it cut off

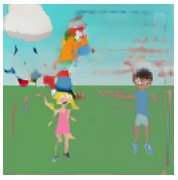
Teller: no sun is whole straight down from sun is a boy standing with a laugh on face **Drawer:** facing left

Teller: yes boy is almost at bottom of screen **Drawer:** what else **Teller:** there is a girl standing at corner of table looks likes she is running smiling **Drawer:** next

Figure 10. Random selection of examples generated by our G prior model for the CoDraw dataset.

Drawer: go **Teller:** large rain cloud left corner touches side cut off on top drops almost touch grass **Drawer:** next

Teller: large rocket on right tip of cloud flying left with very small girl sad legs out sitting on its upper wing **Drawer:** sitting on rocket the rocket is middle scene



Teller: rocket overcloud large regular cloud on right side cut off on top and side a bit surprised boy legs out facing left under cloud **Drawer:** next

Teller: cat facing boy 1 2 inch to left of his feet **Drawer:** next

Teller: i will check and send adjustments **Drawer:** yes i do n't have the girl tell me where is the girl **Teller:** she is sitting on the rockets upper wing her back arm is under the window **Drawer:** check

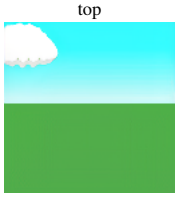
Drawer: where is jenny and mike **Teller:** on left hand side of the screen 2 inches above bottom is a pine tree cut off at top

Teller: straight down from tree is jenny legs crossed facing right right arm in the air

Teller: next to jenny is mike same level standing facing right with arms out mouth open

Teller: on the right hand side inch from the bottom is a duck facing jenny and mike

Teller: straight above duck is soccer ball



Teller: large cloud in left corner top and left are off screen

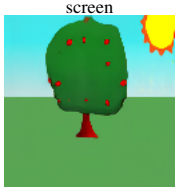
Teller: large apple tree right side top of trunk lines up with horizon right side and top of screen

Teller: in front of trunk little over to the right of trunk by right side is a soccer ball

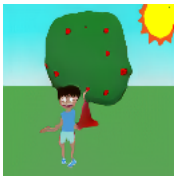
Teller: left side large girl angry face sitting cross legged facing right

Teller: neck on horizon line she 's holding a baseball in her up hand and wearing rainbow hat

Drawer: what do we have **Teller:** med sun right corner **Drawer:** and



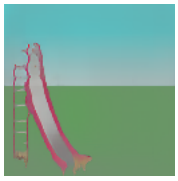
Teller: middle of green with trees half in blue half in green is a apple tree med size **Drawer:** and



Teller: med boy standing on right of tree **Drawer:** face expression and where are his hands **Teller:** he has a tennis ball in right hand he is smiling showing teeth **Drawer:** and

Teller: right arm sticking out across from him is a girl almost to the left edge left arm down right hand with a ball glove **Drawer:** smiling please give more elaborations

Teller: yes smiling looking right she is standing her middle is on the line of green **Drawer:** and



Teller: large slide to the left facing right with owl sitting on top of platform **Drawer:** top of handles or where we stand

Teller: small cloud in top center a bit to the right owl is on the platform **Drawer:** go

Teller: there is a medium to small apple tree on right half in blue half in green right side of tree cut off a bit **Drawer:** go

Teller: a dog directly under tree facing left **Drawer:** size

Teller: a girl standing in front of slide arms up smiling dog looks small

Figure 11. Random selection of examples generated by our Aux model for the CoDraw dataset.

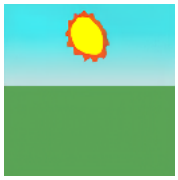
Teller: large pine tree on right trunk 1 4 inch from bottom cut off top and right

Teller: large sand box on left mound on left half bottom edge cut off and left corner cut off mound fully visible

Teller: middle of sandbox large girl running right big toothy smile

Teller: bottom left corner of sandbox large cat looking right top right corner sandbox is baseball

Teller: girl is wearing pirate cap will check when ready



Teller: there is a large sun in the left hand corner

Teller: there is a small girl facing left she has one leg in the air 3 inches from the right 1 2 an inch from the bottom

Teller: small duck facing girl bill at her foot like she is going to kick the duck but move the duck 1 4 inch away about an inch from bottom

Teller: soccer ball to the left of the duck to the left 1 4 inch and up 2 inches

Teller: boy like he is running facing the duck and girl they playing keep away from duck his arms up and looks like he is running mouth open **Teller:** he has glasses on

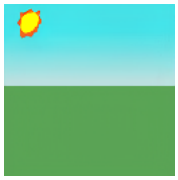
Teller: top left corner is large sun half of it hidden **Drawer:** what else **Teller:** top right corner is large cloud half of it hidden **Drawer:**

Teller: on left side of screen is a large girl standing with her arms in front of her facing right her eyes are even with the horizon line **Drawer:**

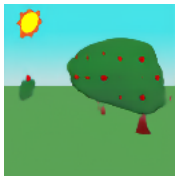
Teller: girl is wearing a chefs hat sunglasses and is holding a pink shovel in her right hand **Drawer:**

Teller: on right side of screen is large boy one hand on hip facing the girl with angry expression his neck is even with horizon line **Drawer:** got it what else

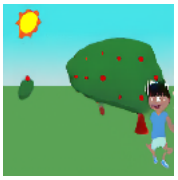
Teller: between them is a large grill **Drawer:**



Teller: big sun on left only top cut a little bit the yellow part



Teller: on right sun small apple tree cut top



Teller: on right of tree happy mike hand front stand facing left head touches the tree

Teller: below sun small sand mike is medium box mound on left left corner is hidden a small duck inside the sandbox

Teller: medium jenny sits on right side of sandbox crossed legs and one hand up facing right head cover right corner of sandbox

Teller: sun top right almost touching top 1 " from left medium size

Teller: big bear at left facing right arm pits at horizon

Teller: medium to small size girl about 3 " away from bear facing left angry arms out holding a bat in right hand head in the blue area

Teller: boy behind her about 1 5 " away arms up startled glove on left hand rainbow hat on his eyes slightly below horizon

Teller: move boy down about an inch face him to left and his striped hat and we are good

Figure 12. Random selection of examples generated by our *D* Concat model for the CoDraw dataset.

Teller: big cloud top left side **Drawer:** got it

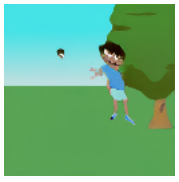
Teller: on the right side is a swing big size **Drawer:** any parts cut off

Teller: girl on the left side neg horizon one arm up facing right happy face **Drawer:** got it

Teller: one part cut from swing just a bit from the right **Drawer: Teller:** next to the right of the cloud is a basketball a bit over the cloud **Drawer:**

Teller: a boy is on the swing the right sit legs cross surprised face facing left color hat baseball glove **Drawer:** got it

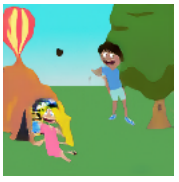
Drawer: ready **Teller:** big oak on right hole facing right hole almost touching horizon **Drawer:** if its large it is huge how much is cut off on the right



Teller: smiling big hands out mike on right left trunk point touching his back hair above horizon

Teller: small hot balloon on left 1 4 from top 1 in from left big tent on left facing right cut off slightly on back top above horizon

Teller: smiling big hands out jenny is in front of left opening tent



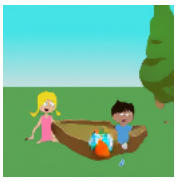
Teller: she has an owl sitting on her left wrist her hand is in the dark opening **Drawer:** she is facing him and large owl right

Drawer: go **Teller:** small bushy tree facing left owl on right middle of tree

Teller: to the left of tree is a medium sandbox mound on right close to bottom **Drawer:** next

Teller: girl sitting in left corner indian style smiling one arm up **Drawer:** next

Teller: boy in right corner sitting indian style smiling with arms open both facing right **Drawer:** next



Teller: under boys right hand is cup in sand straw to left to left of cup is medium beach ball

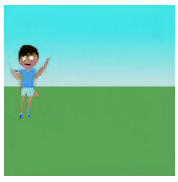
Teller: tree hole facing left cut off from right side a little bit top hiding a bit of the sun

Teller: bumblebee with ear touching the bottom left of tree trunk facing to the right side **Drawer:** got it

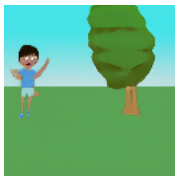
Teller: girl sitting smiling facing right hand behind her one inch from side wearing crown

Teller: crown almost touches the horizon **Drawer:** got it **Teller:** boy faces girl sitting smiling his feet r half inch from hers and raised up a little he wears a beanie with top of it just at horizon

Teller: duck between the two with ducks feet level to boys top foot

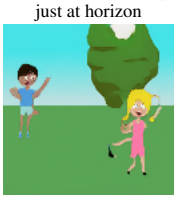


Teller: on the left an inch from the edge is a boy **Drawer:** what is he doing **Teller:** he is facing right standing one hand up teeth showing and holding a racket with one hand that is in the air

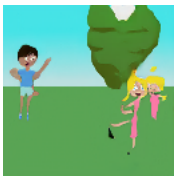


Teller: next to him a medium tree hole facing left head touches the tip of last branch truck aligns with his waist

Teller: on the right 1 inch from edge is a girl sad looking left one hand in the air head aligns with the boy 's **Drawer:** her left hand cut off



Teller: above her is a small cloud right above her can i check



Teller: no about 2 cm from the edge the hand **Teller:** move her 1 more cm from the edge she is holding a yellow small ball in the hand in air

Figure 13. Random selection of examples generated by our D Subtract model for the CoDraw dataset.

Drawer: what is there **Teller:** big thunderbolt on left bolt facing right touching on left edge close to top **Drawer:** and **Teller:** big raindrop cloud to the right of thunderbolt cloud cutting it off on right side a little big shocked jenny with arms up **Drawer:** where is she **Teller:** head right below horizon sad big sitting mike with legs facing right is beside her **Drawer:** and **Teller:** he 's wearing a star hat soccer ball is covering his left foot and shin **Drawer:** and

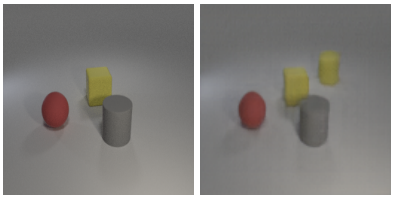
Teller: ready **Drawer:** and ready **Teller:** upper right corner large sun with right edges a bit cut off and top cut off **Teller:** under sun happy boy standing facing left with right arm up his shoulders just above horizon line **Teller:** he is wearing a pirate hat it touches on of the sun tips on the left side **Teller:** happy girl kicking on left side she is about 1 5 inches in from left side her mouth is at the horizon line **Drawer:** got it **Teller:** just a tiny bit off of girls kicking foot is a beach ball a cloud is over the girl towards the right center **Drawer:** is the cloud on the right or the sun you said sun upper right corner

Teller: boy left side kicking leg facing right his half torso aligns with horizon he is shocked **Drawer:** go **Teller:** finger away from his leg soccer ball its bottom part touches horizon **Teller:** right side medium tree 1 4 cut off right side and trunk half way in grass with slight cut off as well right side hole facing left **Drawer:** go **Teller:** plain cloud top middle top part cut off big size in front of tree dog its legs behind completely cut off and it 's facing left **Teller:** near dog is a big cat its tail cover 's dog 's front leg slightly and facing right and then girl sitting smiling facing right

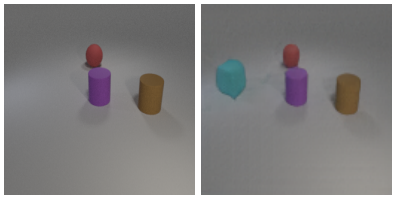
Drawer: ready **Teller:** top left facing left one 1 4 inch from side blade touch top small helicopter facing left **Drawer:** what 's in the left the helicopter **Teller:** nothing it is a 1 4 inch from side flying left **Drawer:** got it it 's tiny right **Teller:** yes **Teller:** below copter is large boy facing right arms out mouth open neck at horizon **Teller:** right of boy his top hand is on first plank is a large picnic table left top corner is highest point pie is there in corner **Drawer:** which side is the pie **Teller:** right of pie is large girl facing left standing with smile no teeth one arm up and one down pie top left corner **Drawer:** where is she to the horizon and she is in front of the table **Teller:** girl in front of table nose at horizon top right corner of table is ketchup **Drawer:** got it **Teller:** 1 2 inch from right side and 1 2 inch from horizon is large grill

Drawer: ready **Teller:** there is a medium in the center of the sky just below the top edge **Drawer:** medium cloud **Teller:** oh sorry medium sun the medium cloud is down and to the right in the sky **Teller:** there is a small oak tree on the left an inch away from the left edge hole facing right 2 3s of the leaves are above the horizon **Teller:** on the right side the kids are both medium sized and facing left jenny is happy jumping half inch from the right edge

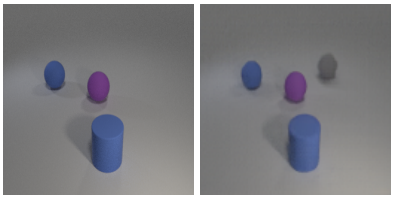
Figure 14. Random selection of examples generated by our Non-iterative model for the CoDraw dataset.



(a) **Left:** Initial Image **Right:** Final Image **Instruction:** Add a yellow cylinder behind the gray cylinder on the right and behind the yellow cube on the right

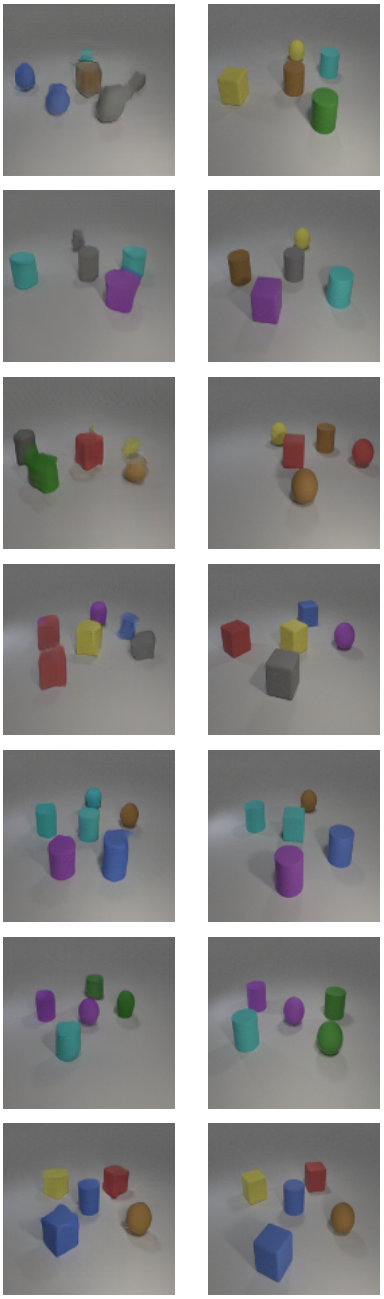


(b) **Left:** Initial Image **Right:** Final Image **Instruction:** Add a cyan cube behind the brown cylinder on the left and behind the purple cylinder on the left



(c) **Left:** Initial Image **Right:** Final Image **Instruction:** Add a gray sphere behind the blue cylinder on the right and behind the purple sphere on the right

Figure 15. When GeNeVA-GAN is provided with an initial image different from the background image used during training, it still adds the desired object with the right properties at the correct location. The model was not trained in this setting and the success of this experiment demonstrates that it has learnt to preserve the existing canvas, understand the existing objects, and add new objects with the correct relationships to existing objects.



Objects detected in generated image: cube gray, cube brown, sphere gray, sphere blue
Objects detected in ground truth image: cube yellow, sphere yellow, cylinder green, cylinder brown, cylinder cyan
Recall: 0.00
rsim: 0.00
Explanation: None of the correct objects are drawn.

Objects detected in generated image: cube gray, cylinder gray, cylinder purple, cylinder cyan
Objects detected in ground truth image: cube purple, sphere yellow, cylinder gray, cylinder brown, cylinder cyan
Recall: 0.4
rsim: 0.25
Explanation: The cyan and gray cylinders are the only two objects detected in the generated image from the five ground truth objects detected.

Objects detected in generated image: cube red, cube green, sphere brown, cylinder gray
Objects detected in ground truth image: cube red, sphere red, sphere brown, sphere yellow, cylinder brown
Recall: 0.4
rsim: 0.35
Explanation: The red cube and brown sphere are detected common to both images. Most of the relationships of these two and the center are correct.

Objects detected in generated image: cube gray, cube red, cube yellow, sphere purple
Objects detected in ground truth image: cube gray, cube red, cube blue, cube yellow, sphere purple
Recall: 0.8
rsim: 0.45
Explanation: Only the blue cube is not detected in the generated image. Several spatial relationships of the common objects and the center are incorrect.

Objects detected in generated image: sphere brown, sphere cyan, cylinder blue, cylinder purple, cylinder cyan
Objects detected in ground truth image: cube cyan, sphere brown, cylinder blue, cylinder purple, cylinder cyan
Recall: 0.8
rsim: 0.675
Explanation: Cyan cube detected in ground truth image is missing from the generated image. Most spatial relationships of the common objects and center are correct.

Objects detected in generated image: sphere green, sphere purple, cylinder green, cylinder purple, cylinder cyan
Objects detected in ground truth image: sphere green, sphere purple, cylinder green, cylinder purple, cylinder cyan
Recall: 1.0
rsim: 0.76
Explanation: All the objects are detected correctly but some of the spatial relationships are incorrect.

Objects detected in generated image: cube red, cube blue, cube yellow, sphere brown, cylinder blue
Objects detected in ground truth image: cube red, cube blue, cube yellow, sphere brown, cylinder blue
Recall: 1.00
rsim: 1.00
Explanation: All the objects are detected correctly and are in the correct relative positions.

Figure 16. **Column 1:** Generated final image; **Column 2:** Ground truth final image; **Column 3:** List of objects detected in the generated and ground truth image, the recall on object detection, the value of the relational similarity (rsim) metric. The examples have been selected to qualitatively show examples with diverse score values between the minimum (0) and the maximum (1) values of the rsim metric.