Compositional Visual Generation with Composable Diffusion Models

Nan Liu[1]*, Shuang Li[2]*, Yilun Du[2]*
Antonio Torralba[2], and Joshua B. Tenenbaum[2]

[1] University of Illinois Urbana-Champaign
[2] Massachusetts Institute of Technology
nanliu4@illinois.edu, {lishuang,yilundu,torralba,jbt}@mit.edu

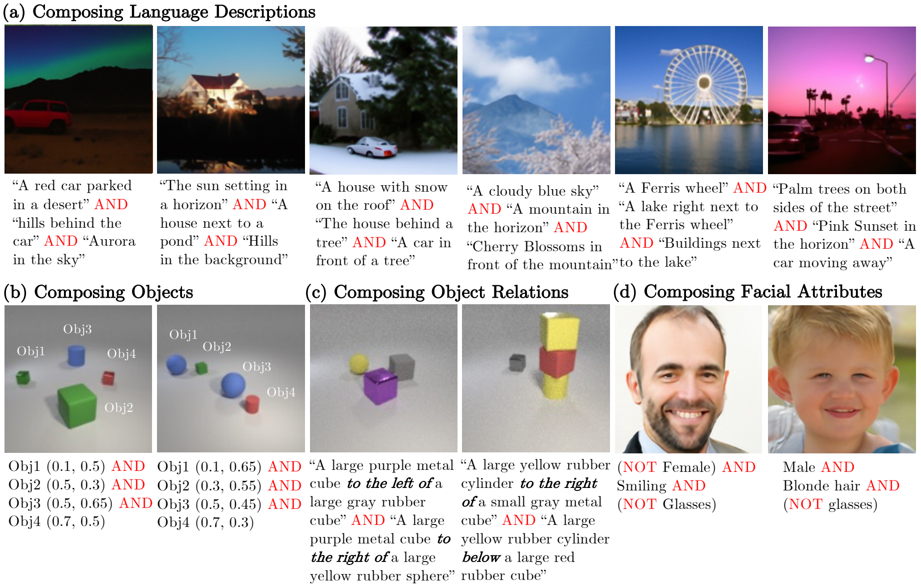
Abstract. Large text-guided diffusion models, such as DALLE-2, are able to generate stunning photorealistic images given natural language descriptions. While such models are highly flexible, they struggle to understand the composition of certain concepts, such as confusing the attributes of different objects or relations between objects. In this paper, we propose an alternative structured approach for compositional generation using diffusion models. An image is generated by composing a set of diffusion models, with each of them modeling a certain component of the image. To do this, we interpret diffusion models as energy-based models in which the data distributions defined by the energy functions may be explicitly combined. The proposed method can generate scenes at test time that are substantially more complex than those seen in training, composing sentence descriptions, object relations, human facial attributes, and even generalizing to new combinations that are rarely seen in the real world. We further illustrate how our approach may be used to compose pre-trained text-guided diffusion models and generate photorealistic images containing all the details described in the input descriptions, including the binding of certain object attributes that have been shown difficult for DALLE-2. These results point to the effectiveness of the proposed method in promoting structured generalization for visual generation.

Keywords: Compositionality, Diffusion Models, Energy-based Models, Visual Generation

1 Introduction

Our understanding of the world is highly compositional in nature. We are able to rapidly understand new objects from their components, or compose words into complex sentences to describe the world states we encounter [24]. We are able

* indicates equal contribution.
Correspondence to: Shuang Li <lishuang@mit.edu>, Yilun Du <yilundu@mit.edu>
Webpage: https://energy-based-model.github.io/Compositional-Visual-Generation-with-Composable-Diffusion-Models/



(a) Composing Language Descriptions

"A red car parked in a desert" AND "hills behind the car" AND "Aurora in the sky"

"The sun setting in a horizon" AND "A house next to a pond" AND "Hills in the background"

"A house with snow on the roof" AND "The house behind a tree" AND "A car in front of a tree"

"A cloudy blue sky" AND "A mountain in the horizon" AND "Cherry Blossoms in front of the mountain"

"A Ferris wheel" AND "A lake right next to the Ferris wheel" AND "Buildings next to the lake"

"Palm trees on both sides of the street" AND "Pink Sunset in the horizon" AND "A car moving away"

(b) Composing Objects

Obj1 (0.1, 0.5) AND Obj2 (0.5, 0.3) AND Obj3 (0.5, 0.65) AND Obj4 (0.7, 0.5)

Obj1 (0.1, 0.65) AND Obj2 (0.3, 0.55) AND Obj3 (0.5, 0.45) AND Obj4 (0.7, 0.3)

(c) Composing Object Relations

"A large purple metal cube ***to the left of*** a large gray rubber cube" AND "A large purple metal cube ***to the right of*** a large yellow rubber sphere"

"A large yellow rubber cylinder ***to the right of*** a small gray metal cube" AND "A large yellow rubber cylinder ***below*** a large red rubber cube"

(d) Composing Facial Attributes

(NOT Female) AND Smiling AND (NOT Glasses)

Male AND Blonde hair AND (NOT glasses)

Fig. 1: Our method allows compositional visual generation across a variety of domains, such as language descriptions, objects, object relations, and human attributes.

to make 'infinite use of finite means' [4], *i.e.*, repeatedly reuse and recombine concepts we have acquired in a potentially infinite manner. We are interested in constructing machine learning systems to have such compositional capabilities, particularly in the context of generative modeling.

Existing text-conditioned diffusion models such as DALLE-2 [35] have recently made remarkable strides towards compositional generation, and are capable in generating photorealistic images given textual descriptions. However, such systems are not fully compositional in nature and generate incorrect images when given more complex descriptions [28,45]. An underlying difficulty may be that such models encode text descriptions as fixed-size latent vectors. However, as textual descriptions become more complex, more information needs to be squeezed into the fixed-size vector. Thus it is impossible to encode arbitrarily complex textual descriptions.

In this work, we propose to factorize the compositional generation problem, using different diffusion models to capture different subsets of a compositional specification. These diffusion models are then explicitly composed together to jointly generate an image. By explicitly factorizing the compositional generative modeling problem, our method is able to generalize to significantly more complex combinations that are unseen during training.

Such an explicit form of compositionality has been explored before under the context of Energy-Based Models (EBMs) [7,8,26]. However, directly training EBMs has been proved to be unstable and hard to scale. We show that diffusion models can be interpreted as implicitly parameterized EBMs, which can be further composed for image generation, significantly improving training stability and image quality.

Our proposed method enables zero-shot compositional generation across different domains as shown in Figure 1. First, we illustrate how our approach may be applied to large pre-trained diffusion models, such as GLIDE [30], to compose multiple text descriptions. Next, we illustrate how our approach can be applied to compose objects and object relations, enabling zero-shot generalization to a larger number of objects. Finally, we illustrate how our framework can compose different facial attributes to generate human faces.

Contributions: In this paper, we introduce an approach towards compositional visual generation using diffusion models. First, we show that diffusion models can be composed by interpreting them as energy-based models and drawing on this connection, show how we may compose diffusion models together.

Second, we propose two compositional operators, conjunction and negation, on top of diffusion models that allow us to compose concepts in different domains during inference without any additional training. We show that the proposed method enables effective zero-shot combinatorial generalization. Finally, we evaluate our method on composing language descriptions, objects, object relations, and human facial attributes. Our method can generate high-quality images containing all the concepts and outperforms baselines by a large margin. For example, the accuracy of our method is 24.02% higher than the best baseline for composing three objects in the specified positions on the CLEVR dataset.

2 Related Work

Controllable Image Generation. Our work is related to existing work on controllable image generation. One type of approach towards controllable image generation specifies the underlying content of an image utilizing text through either GANs [49,50,2], VQ-VAEs [36], or diffusion models [30]. An alternative type of approach towards controllable image generation manipulates the underlying attributes in an image [41,48,52]. In contrast, we are interested in *compositionally controlling* the underlying content of an image at test time, generating images that exhibit compositions of multiple different types of image content. Thus, most relevant to our work, existing work has utilized EBMs to compose different factors describing a scene [7,32,8,26]. We illustrate how we may implement such probabilistic composition on diffusion models, achieving better performance.

Diffusion Models. Diffusion models have emerged as a promising class of generative models that formulates the data-generating process as an iterative denoising procedure [42,15]. The denoising procedure can be seen as parameterizing the gradients of the data distribution [44], connecting diffusion models to EBMs [25,10,33,12,11]. Diffusion models have recently shown great promise in image generation [6], enabling effective image editing [29,22], text conditioning [30,37,13], and image inpainting [39]. The iterative, gradient-based sampling of diffusion models lends itself towards flexible conditioning [6], enabling us to compose factors across different images. While diffusion models have been developed for image generation [43], they have further proven successful in the generation of waveforms [3], 3D shapes [51], decision making [17] and text [1], suggesting that our proposed composition operators may further be applied in such domains.

3 Background

3.1 Denoising Diffusion Models

Denoising Diffusion Probabilistic Models (DDPMs) are a class of generative models where generation is modeled as a denoising process. Starting from sampled noise, the diffusion model performs T denoising steps until a sharp image is formed. In particular, the denoising process produces a series of intermediate images with decreasing levels of noise, denoted as $\boldsymbol{x}_T, \boldsymbol{x}_{T-1}, ..., \boldsymbol{x}_0$, where $\boldsymbol{x}_T$ is sampled from a Gaussian prior and $\boldsymbol{x}_0$ is the final output image.

DDPMs construct a forward diffusion process by gradually adding Gaussian noise to the ground truth image. A diffusion model then learns to revert this noise corruption process. Both the *forward processes* $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and the *reverse process* $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ are modeled as the products of Markov transition probabilities:

$$q(\boldsymbol{x}_{0:T}) = q(\boldsymbol{x}_0) \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \qquad p_\theta(\boldsymbol{x}_{T:0}) = p(\boldsymbol{x}_T) \prod_{t=T}^{1} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t), \tag{1}$$

where $q(\boldsymbol{x}_0)$ is the real data distribution and $p(\boldsymbol{x}_T)$ is a standard Gaussian prior.

A *generative process* $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is trained to generate realistic images by approximating the reverse process through variational inference. Each step of the *generative process* is a Gaussian distribution with learned mean and covariance:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\mu_\theta(\boldsymbol{x}_t, t), \sigma_t^2) = \mathcal{N}(\boldsymbol{x}_t + \epsilon_\theta(\boldsymbol{x}_t, t), \sigma_t^2), \tag{2}$$

where $\boldsymbol{x}_{t-1}$ is parameterized by a mean $\mu_\theta(\boldsymbol{x}_t, t)$ represented by a perturbation $\epsilon_\theta(\boldsymbol{x}_t, t)$ to a noisy image $\boldsymbol{x}_t$. The goal is to remove the noise gradually by predicting a less noisy image at timestep $\boldsymbol{x}_{t-1}$ given a noisy image $\boldsymbol{x}_t$. To generate real images, we sample $\boldsymbol{x}_{t-1}$ from $t = T$ to $t = 1$ using the parameterized marginal distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, with an individual step corresponding to:

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t + \epsilon_\theta(\boldsymbol{x}_t, t) + \mathcal{N}(0, \sigma_t^2). \tag{3}$$

The generated images become more realistic over multiple iterations.

3.2 Energy Based Models

Energy-Based Models (EBMs) [10,9,12,33] are a class of generative models where the data distribution is modeled using an unnormalized probability density. Given an image $\boldsymbol{x} \in \mathbb{R}^D$, the probability density of image $\boldsymbol{x}$ is defined as:

$$p_\theta(\boldsymbol{x}) \propto e^{-E_\theta(\boldsymbol{x})}, \tag{4}$$

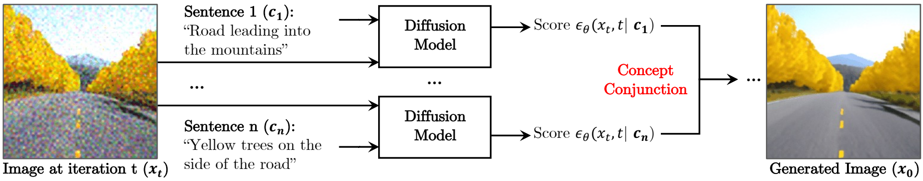

Sentence 1 (c_1):
"Road leading into the mountains"
...
Sentence n (c_n):
"Yellow trees on the side of the road"
Image at iteration t (x_t)
Diffusion Model
...
Diffusion Model
Score $\epsilon_\theta(x_t, t|\, c_1)$
Score $\epsilon_\theta(x_t, t|\, c_n)$
Concept Conjunction
...
Generated Image (x_0)

Fig. 2: **Compositional generation.** Our method can compose multiple concepts during inference and generate images containing all the concepts without further training. We first send an image from iteration t and each of the concept to the diffusion model to generate a set of scores $\{\epsilon_\theta(\boldsymbol{x}_t, t|\boldsymbol{c}_1), \dots, \epsilon_\theta(\boldsymbol{x}_t, t|\boldsymbol{c}_n)\}$. We then compose different concepts using the proposed compositional operators, such as conjunction, to denoise the generated images. The final image is obtained after T iterations.

where the energy function $E_\theta(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}$ is a learnable neural network.

A gradient based MCMC procedure, Langevin dynamics [10], is then used to sample from an unnormalized probability distribution to iteratively refine the generated image $\boldsymbol{x}$:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \frac{\lambda}{2}\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}_{t-1}) + \mathcal{N}(0, \sigma^2). \tag{5}$$

The procedure for sampling from diffusion models in Equation (3) is functionally similar to the sampling procedure used by EBMs in Equation (5). In both settings, images are iteratively refined starting from Gaussian noise, with a small amount of additional Gaussian noise added at each iterative step.

4 Our approach

In this section, we first introduce how we may interpret diffusion models as energy-based models in section 4.1 and then introduce how we compose diffusion models for visual generation in section 4.2.

4.1 Diffusion Models as Energy Based Models

The sampling procedure of diffusion models in Equation (3) and EBMs in Equation (5) are functionally similar. At a timestep t, in diffusion models, images are updated using a learned denoising network $\epsilon_\theta(\boldsymbol{x}_t, t)$ while in EBMs, images are updated using the gradient of the energy function $\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}_t) \propto \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})$, which is the score of the estimated probability distribution $p_\theta(\boldsymbol{x})$.

The denoising network $\epsilon_\theta(\boldsymbol{x}_t, t)$ is trained to predict the underlying score of the data distribution [47,43] when the number of diffusion steps increases to infinity. Similarly, an EBM is trained so that $\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}_t)$ corresponds to the score of the data distribution as well. In this sense, $\epsilon_\theta(\boldsymbol{x}_t, t)$ and $\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}_t)$ are fuctionally the same, and the underlying sampling procedure in Equation (3) and Equation (5) are equivalent. We may view a trained diffusion model $\epsilon_\theta(\boldsymbol{x}_t, t)$ as implicitly parameterizing an EBM by defining its data gradient $\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}_t)$ at each data point, and we will subsequently refer to $\epsilon_\theta(\boldsymbol{x}_t, t)$ as the score function. Such a parameterization enables us to leverage past work towards composing EBMs and apply it to diffusion models.

Composing EBMs. Previous EBMs [14,7] have shown good compositionality ability for visual generation. Given n independent EBMs, $E^1_\theta(\boldsymbol{x}), \cdots, E^n_\theta(\boldsymbol{x})$, the functional form of EBMs in Equation (4) enable us to compose multiple separate EBMs together to obtain a new EBM. The composed distribution can be represented as:

$$p_{\text{compose}}(\boldsymbol{x}) \propto p^1_\theta(\boldsymbol{x}) \cdots p^n_\theta(\boldsymbol{x}) \propto e^{-\sum_i E^i_\theta(\boldsymbol{x})} = e^{-E_\theta(\boldsymbol{x})}, \tag{6}$$

where $p^i_\theta \propto e^{-E^i_\theta(\boldsymbol{x})}$ is the probability density of image $\boldsymbol{x}$ (Equation (4)). Langevin dynamics is used to iteratively refine the generated image $\boldsymbol{x}$.

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \frac{\lambda}{2}\nabla_{\boldsymbol{x}}(\sum_i E^i_\theta(\boldsymbol{x}_{t-1})) + \mathcal{N}(0, \sigma^2). \tag{7}$$

Composing Diffusion Models. By leveraging the interpretation that diffusion models are functionally similar to EBMs, we may compose diffusion models in a similar way. The *generative process* and the score function of a diffusion model can be represented as $p^i_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ and $\epsilon^i_\theta(\boldsymbol{x}, t)$, respectively. If we treat the individual score function in diffusion models as the learned gradient of energy functions in EBMs, the composition of diffusion models has a score function of $\sum_i \epsilon^i_\theta(\boldsymbol{x}, t)$. Thus the *generative process* of composing multiple diffusion models becomes:

$$p_{\text{compose}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_t + \sum_i \epsilon^i_\theta(\boldsymbol{x}_t, t), \sigma^2_t). \tag{8}$$

A complication when parameterizing of a gradient field of EBM $\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}_t)$ with a learned score function $\epsilon_\theta(\boldsymbol{x}, t)$, is that the gradient field may not be conservative, and thus does not lead to a valid probability density. However, as discussed in [40], explicitly parameterizing the learned function $\epsilon_\theta(\mathbf{x}, t)$ as the gradient of EBM achieves similar performance as the non-conservative parameterization of diffusion models, suggesting this is not problematic.

4.2 Compositional Generation through Diffusion Models

Next, we discuss how we compose diffusion models for image generation. We aim to generate images conditioned on a set of concepts $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n\}$. To do this, we represent each concept $\boldsymbol{c}_i$ as an individual diffusion model, which are composed to generate images. Inspired by EBMs [7,26], we define two compositional operators, **conjunction (AND)** and **negation (NOT)**, to compose diffusion models. We learn a set of diffusion models representing the conditional image generation $p(\boldsymbol{x}|\boldsymbol{c}_i)$ given factor $\boldsymbol{c}_i$ and an unconditional image generation $p(\boldsymbol{x})$.

Concept Conjunction (AND). We aim to generate images containing certain attributes. Following [7], the conditional probability can be factorized as

$$p(\boldsymbol{x}|\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n) \propto p(\boldsymbol{x}, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n) = p(\boldsymbol{x}) \prod_i p(\boldsymbol{c}_i|\boldsymbol{x}). \tag{9}$$

Algorithm 1 Code for Composing Diffusion Models

```
Require Diffusion model $\epsilon_\theta(\boldsymbol{x}, t|\boldsymbol{c})$, scale $\alpha$, negation factor $\beta$, noises $\sigma_t$
// Code for conjunction
Initialize sample $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
for $t = T, \dots, 1$ do
    $\epsilon_i \leftarrow \epsilon_\theta(\boldsymbol{x}_t, t|\boldsymbol{c}_i)$    // compute conditional scores for each factor $\boldsymbol{c}_i$
    $\epsilon \leftarrow \epsilon_\theta(\boldsymbol{x}_t, t)$    // compute unconditional score
    $\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{x}_t + \epsilon + \alpha \sum_i (\epsilon_i - \epsilon), \sigma_t^2)$    // sampling
end for

// Code for negation
Initialize sample $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
for $t = T, \dots, 1$ do
    $\tilde{\epsilon}_j \leftarrow \epsilon_\theta(\boldsymbol{x}_t, t|\tilde{\boldsymbol{c}}_j)$    // compute conditional scores for negated factor $\tilde{\boldsymbol{c}}_j$
    $\epsilon_i \leftarrow \epsilon_\theta(\boldsymbol{x}_t, t|\boldsymbol{c}_i)$    // compute conditional scores for each factor $\boldsymbol{c}_i$
    $\epsilon \leftarrow \epsilon_\theta(\boldsymbol{x}_t, t)$    // compute unconditional score
    $\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{x}_t + \epsilon + \alpha \{-\beta(\tilde{\epsilon}_j - \epsilon) + \sum_i (\epsilon_i - \epsilon)\}, \sigma_t^2)$    // sampling
end for
```

We can represent $p(\boldsymbol{c}_i|\boldsymbol{x})$ using a combination of a conditional distribution $p(\boldsymbol{x}|\boldsymbol{c}_i)$ and an unconditional distribution $p(\boldsymbol{x})$, with both of them are parameterized as diffusion models $p(\boldsymbol{c}_i|\boldsymbol{x}) \propto \frac{p(\boldsymbol{x}|\boldsymbol{c}_i)}{p(\boldsymbol{x})}$. The expression of $p(\boldsymbol{c}_i|\boldsymbol{x})$ corresponds to the implicit classifier that represents the likelihood of $\boldsymbol{x}$ exhibiting factor $\boldsymbol{c}_i$. Substituting $p(\boldsymbol{c}_i|\boldsymbol{x})$ in Equation 9, we can rewrite Equation 9 as the probability distribution:

$$p(\boldsymbol{x}|\boldsymbol{c}_1, \dots, \boldsymbol{c}_n) \propto p(\boldsymbol{x}) \prod_i \frac{p(\boldsymbol{x}|\boldsymbol{c}_i)}{p(\boldsymbol{x})}. \tag{10}$$

We sample from this resultant distribution using Equation (8), with a new composed score function $\epsilon^*(\boldsymbol{x}_t, t)$:

$$\epsilon^*(\boldsymbol{x}_t, t) = \epsilon_\theta(\boldsymbol{x}_t, t) + \alpha \sum_i (\epsilon_\theta(\boldsymbol{x}_t, t|\boldsymbol{c}_i) - \epsilon_\theta(\boldsymbol{x}_t, t)), \tag{11}$$

where the constant α corresponds to a temperature scaling on $\frac{p(\boldsymbol{x}|\boldsymbol{c}_i)}{p(\boldsymbol{x})}$. We may then generate the composed sample using the following *generative process*:

$$p^*(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\boldsymbol{x}_t + \epsilon^*(\boldsymbol{x}_t, t), \sigma_t^2). \tag{12}$$

In the setting in which image generation is conditioned on a single concept, the above sampling procedure reduces to classifier-free guidance.

Concept Negation (NOT). In concept negation, we aim to generate image with the absence of a certain factor $\tilde{\boldsymbol{c}}_j$. We also need to generate images that look realistic. One easy way to do this is to make the generated images contain another factor $\boldsymbol{c}_i$. Following [7], concept negation can be represented as the composed probability distribution $p(\boldsymbol{x}|\text{not } \tilde{\boldsymbol{c}}_j, \boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_n) = \frac{\prod_i p(\boldsymbol{x}|\boldsymbol{c}_i)}{p(\boldsymbol{x}|\tilde{\boldsymbol{c}}_j)^\alpha}$. Following [7], we refactorize the joint probability distribution as:

$$p(\boldsymbol{x}|\text{not } \tilde{\boldsymbol{c}}_j, \boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n) \propto p(\boldsymbol{x}, \text{not } \tilde{\boldsymbol{c}}_j, \boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n) = p(\boldsymbol{x}) \frac{\prod_i p(\boldsymbol{c}_i|\boldsymbol{x})}{p(\tilde{\boldsymbol{c}}_j|\boldsymbol{x})^\beta}. \quad (13)$$

Using the implicit classifier factorization $p(\boldsymbol{c}_i|\boldsymbol{x}) \propto \frac{p(\boldsymbol{x}|\boldsymbol{c}_i)}{p(\boldsymbol{x})}$, we can rewrite the above expression as:

$$p(\boldsymbol{x}|\text{not } \tilde{\boldsymbol{c}}_j, \boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n) \propto p(\boldsymbol{x}) \frac{p(\boldsymbol{x})^\beta}{p(\boldsymbol{x}|\tilde{\boldsymbol{c}}_j)^\beta} \prod_i \frac{p(\boldsymbol{x}|\boldsymbol{c}_i)}{p(\boldsymbol{x})}. \quad (14)$$

Similarly, we may construct a new learned score $\epsilon^*(\boldsymbol{x}_t, t)$ using Equation (8) to sample from the *generative process* to represent this negated probability distribution at each timestep:

$$\epsilon^*(\boldsymbol{x}_t, t) = \epsilon_\theta(x_t, t) + \alpha\{-\beta(\epsilon_\theta(\boldsymbol{x}_t, t|\tilde{\boldsymbol{c}}_j) - \epsilon_\theta(\boldsymbol{x}_t, t)) + \sum_i (\epsilon_\theta(\boldsymbol{x}_t, t|\boldsymbol{c}_i) - \epsilon_\theta(\boldsymbol{x}_t, t))\}, \quad (15)$$

where the constant α corresponds to a temperature scaling on each implicit classifier $\frac{p(\boldsymbol{x}|\boldsymbol{c}_i)}{p(\boldsymbol{x})}$. We may then generate samples from this modified learned score using Equation 12.

Algorithm 1 provides the pseudo-code for composing diffusion models using concept conjunction and negation. Our method can compose pre-trained diffusion models during inference time without any additional training.

5 Experiment Setup

5.1 Datasets

CLEVR. CLEVR [18] is a synthetic dataset containing objects with different shapes, colors, and sizes. The training set consists of 30,000 images at 128×128 resolution. Each image contains $1 \sim 5$ objects and a 2D coordinate (x, y) label indicating that the image contains an object at (x, y). In our experiments, the 2D coordinate label is the coordinate of one random object in the image.

Relational CLEVR. Relational CLEVR [26] contains relational descriptions between objects in the image, such as "a red cube to the left of a blue cylinder". The training dataset contains $50,000$ images at 128×128 resolution. Each training image contains $1 \sim 5$ objects, and one label describing a relation between two objects. If there is only one object in the image, the second object in the relational description is null.

FFHQ. FFHQ [20] is a real world human face dataset. The original FFHQ dataset consists of 70,000 human face images without labels. [5] annotates three binary attributes, including *smile*, *gender*, and *glasses*, for the images using pre-trained classifiers. As a result, there are 51,067 images labeled by the classifiers.

5.2 Evaluation Metrics

Binary classification accuracy. During testing, we evaluate the performance of the proposed method and baselines on three different settings. The first test setting, **1 Component**, generates images conditioned on a single concept (matching the training distribution). The second and third test settings, **2 Components** and **3 Components**, generate images by composing two and three concepts respectively using the *conjunction* and *negation* operators. They are used to evaluate the models' generalization ability to new combinations.

For each task, we use the training data (real images) to train a binary classifier that takes an image and a concept, *e.g.* 'smiling', as input, and predicts whether the image contains or represents the concept. We then apply this classifier to a generated image, checking whether it faithfully captures each of the concepts. In each test setting, each method generates $5,000$ images for evaluation. The accuracy of the method is the percentage of generated images capturing all the concepts (See Appendix B).

Fréchet Inception Distance (FID) is a commonly used metric for evaluating the quality of generated images. It uses a pretrained inception model [46] to extract features for the generated images and real images and measure their feature similarity. Specifically, we use Clean-FID [34] to evaluate the generated images. FID is usually computed on $50,000$ generated images, but we use $5,000$ images in our experiments, thus causing our FID scores to be higher than usual.

6 Experiments

We compare the proposed method and baselines (section 6.1) on compositional generation on different domains. We show results of composing natural language descriptions (section 6.2), objects (section 6.3), object relational descriptions (section 6.4), and human facial attributes (Appendix A). Results analysis are shown in section 6.5.

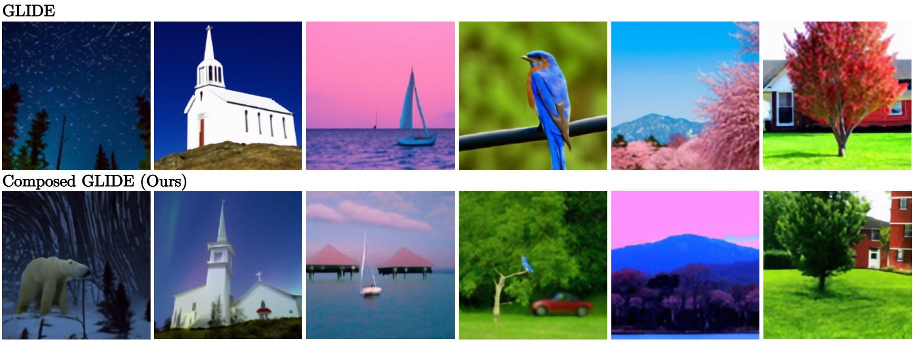
6.1 Baselines

We compare our method with baselines for compositional visual generation.

Energy-based models (EBM) [7] is the first paper using EBMs for compositional visual generation. They propose three compositional operators for composing different concepts. Our works is inspired by [7], but we compose diffusion models and achieve better results.

StyleGAN2 [21] is one of the state-of-the-art GAN methods for unconditional image generation. To enable compositional image generation, We optimize the latent code by decreasing the loss between a trained binary classifier and the given labels. We use the final latent code to generate images.

LACE [32] uses pre-trained classifiers to generate energy scores in the latent space of the pre-trained StyleGAN2 model. To enable compositional image synthesis, LACE uses compositional operators [7].



"A starry night sky" AND "A polar bear in a forest"

"A white church sitting on a hill" AND "Aurora in the sky"

"A pink sky in the horizon" AND "A sailboat at the sea" AND "Overwater bungalows"

"A blue bird on a tree" AND "A red car behind the tree" AND "A green forest in the background"

"A pink sky" AND "A blue mountain in the horizon" AND "Cherry Blossoms in front of the mountain"

"A green tree swaying in the wind" AND "A red brick house located behind a tree" AND "A healthy lawn in front of the house"

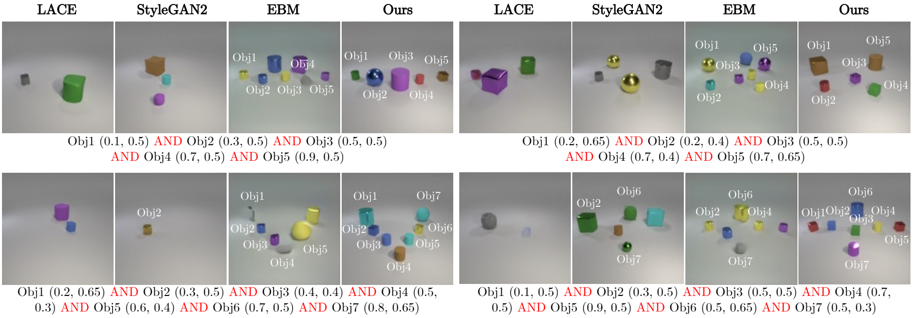
Fig. 3: **Composing Language Descriptions.** We develop *Composed GLIDE (Ours)*, a version of *GLIDE* [30] that utilizes our compositional operators to combine textual descriptions, without further training. We compare it to the original *GLIDE*, which directly encodes the descriptions as a single long sentence. Our approach more accurately captures text details, such as the "overwater bungalows" in the third example.

GLIDE [30] is a recent state-of-the-art text-conditioned diffusion model. For composing language descriptions, we use the classifier-free model released by OpenAI for comparison. For the rest tasks, we train the GLIDE model using the same data as our method.

6.2 Composing Language Descriptions

We first validate that our approach can compose natural language descriptions. We use the pre-trained text conditional diffusion models from *GLIDE* [30]. The image generation results of the released *GLIDE* model (a small model) is shown in Figure 3. We develop *Composed GLIDE (Ours)*, a version of GLIDE [30] that utilizes our compositional operators to combine textual descriptions, without further training. We compare this model to the original GLIDE model, which directly encodes the descriptions as a single long sentence.

In Figure 3, *GLIDE* takes a single long sentence as input, for example "A pink sky in the horizon, a sailboat at the sea, and overwater bungalows". In contrast, *Composed GLIDE (Ours)* composes several short sentences using the concept conjunction operator, *e.g.* "A pink sky in the horizon" AND "A sailboat at the sea" AND "Overwater bungalows". While both *GLIDE* and *Composed GLIDE (Ours)* can generate reasonable images containing objects described in the text prompt, our approach with the compositional operators can more accurately capture text details, such as the presence of "a polar bear" in the first example and the "overwater bungalows" in the third example.



Obj1 (0.1, 0.5) AND Obj2 (0.3, 0.5) AND Obj3 (0.5, 0.5) AND Obj4 (0.7, 0.5) AND Obj5 (0.9, 0.5)

Obj1 (0.2, 0.65) AND Obj2 (0.2, 0.4) AND Obj3 (0.5, 0.5) AND Obj4 (0.7, 0.4) AND Obj5 (0.7, 0.65)

Obj1 (0.2, 0.65) AND Obj2 (0.3, 0.5) AND Obj3 (0.4, 0.4) AND Obj4 (0.5, 0.3) AND Obj5 (0.6, 0.4) AND Obj6 (0.7, 0.5) AND Obj7 (0.8, 0.65)

Obj1 (0.1, 0.5) AND Obj2 (0.3, 0.5) AND Obj3 (0.5, 0.5) AND Obj4 (0.7, 0.5) AND Obj5 (0.9, 0.5) AND Obj6 (0.5, 0.65) AND Obj7 (0.5, 0.3)

Fig. 4: **Composing Objects.** Our method can compose multiple objects while baselines either miss or generate more objects.

Table 1: Quantitative evaluation of 128 × 128 image generation results on CLEVR. The binary classification accuracy (Acc) and FID scores are reported. Our method outperforms baselines on all the three test settings.

Models	**1 Component**		**2 Components**		**3 Components**	
	Acc (%) ↑	FID ↓	Acc (%) ↑	FID ↓	Acc (%) ↑	FID ↓
EBM [7]	70.54	78.63	28.22	65.45	7.34	58.33
StyleGAN2 [21]	1.04	51.37	0.04	23.29	0.00	19.01
LACE [32]	0.70	50.92	0.00	22.83	0.00	19.62
GLIDE [30]	0.86	61.68	0.06	38.26	0.00	37.18
Ours	**86.42**	**29.29**	**59.20**	**15.94**	**31.36**	**10.51**

6.3 Composing Objects

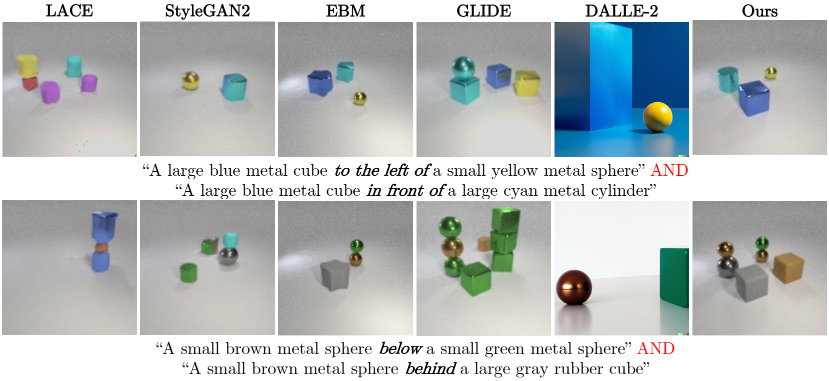
Given a set of 2D object positions, we aim to generate images containing objects at those positions.

Qualitative results. We compare the proposed method and baselines on composing objects in Figure 4. We only show the concept conjunction here because the object positions are not binary values, and thus negation of object positions is not interpretable. Given a set of object position labels, we compose them to generate images. Our model can generate images of objects at certain locations while the baseline methods either miss objects or generate incorrect objects.

Quantitative results. As shown in Table 1, our method outperforms baselines by a large margin. The binary classification accuracy of our method is 15.88% higher than the best baseline, EBM, in the *1 component* test setting and is 24.02 higher than EBM on the more challenging *3 Components* setting. Our method is more effective in zero-shot compositional generalization. In addition, our method can generate images with lower FID scores (more similar to the real images).

6.4 Composing Object Relations

Qualitative results. We further compare the proposed approach and baselines on composing object relational descriptions in Figure 5. Our model is trained to



"A large blue metal cube ***to the left of*** a small yellow metal sphere" AND
"A large blue metal cube ***in front of*** a large cyan metal cylinder"

"A small brown metal sphere ***below*** a small green metal sphere" AND
"A small brown metal sphere ***behind*** a large gray rubber cube"

Fig. 5: **Composing Visual Relations.** Image generation results on the Relational CLEVR dataset. Our model is trained to generate images conditioned on a single object relation, but during inference, our model can compose multiple object relations, generating better results than baselines.

Table 2: Quantitative evaluation of 128 × 128 image generation results on the Relational CLEVR dataset. The binary classification accuracy (Acc) and FID score on three test settings are reported. Although *EBM* performs well on the binary classification accuracy, its FID score is much lower than other methods. Our method achieves comparable or better results than baselines.

Models	**1 Component**		**2 Components**		**3 Components**	
	Acc (%) ↑	FID ↓	Acc (%) ↑	FID ↓	Acc (%) ↑	FID ↓
EBM [26]	**78.14**	44.41	**24.16**	55.89	**4.26**	58.66
StyleGAN2 [21]	20.18	**22.29**	1.66	30.58	0.16	31.30
LACE [32]	1.10	40.54	0.10	40.61	0.04	40.60
GLIDE [30]	32.68	57.48	7.48	59.47	2.14	61.52
Ours	60.40	29.06	21.84	**29.82**	2.80	**26.11**

generate images conditioned on a single object relation, but it can compose multiple object relations during inference without additional training. Both *LACE* and *StyleGAN2* fail to capture object relations in the input sentences, but *EBM* and our method can correctly compose multiple object relations. Our method generates higher-quality images compared with *EBM*, *e.g.* the object boundaries are sharper in our results than *EBM*. Surprisingly, *DALLE-2* and *GLIDE* can generate high-quality images, but they fail to understand object relations.

Quantitative results. Same as experiments in section 6.3, we evaluate the proposed method and baselines on three test settings in Table 2. We train a binary classifier to evaluate whether an image contains objects that satisfy the input relational description. For binary classification accuracy, our method outperforms *StyleGAN2 (CLIP)*, *LACE*, and *GLIDE* on all three test settings. *EBMs* perform well on composing relational descriptions, but their FID scores are much worse than other methods, *i.e.* their generated images are not realistic.

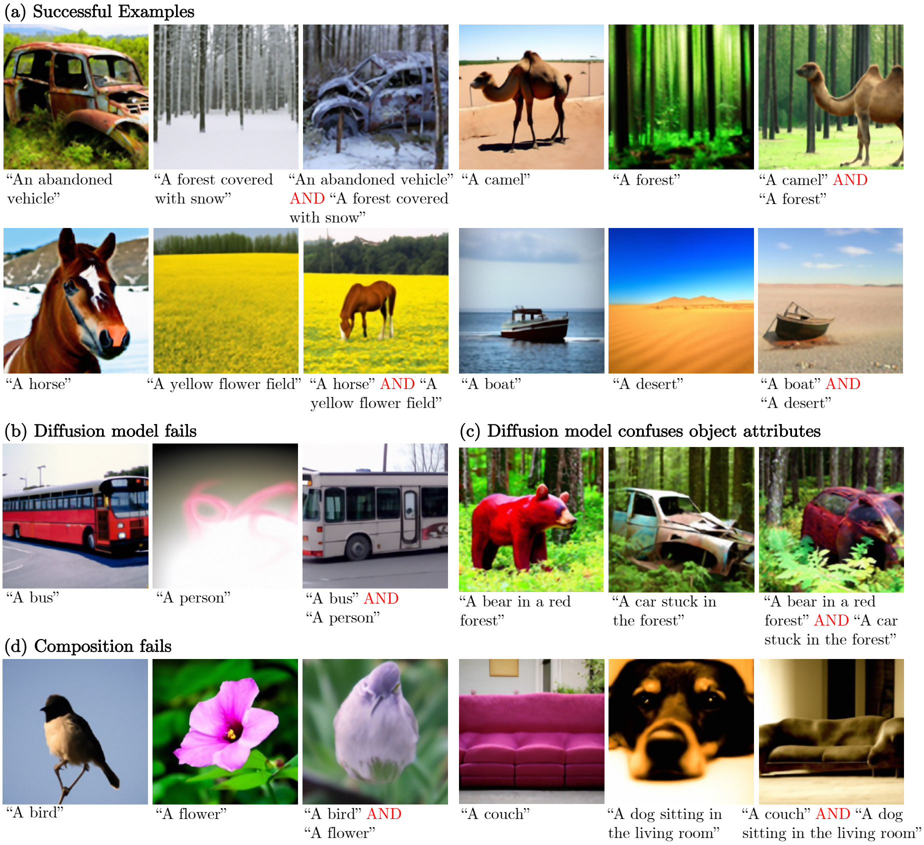

(a) Successful Examples
"An abandoned vehicle"
"A forest covered with snow"
"An abandoned vehicle" AND "A forest covered with snow"
"A camel"
"A forest"
"A camel" AND "A forest"
"A horse"
"A yellow flower field"
"A horse" AND "A yellow flower field"
"A boat"
"A desert"
"A boat" AND "A desert"
(b) Diffusion model fails
"A bus"
"A person"
"A bus" AND "A person"
(c) Diffusion model confuses object attributes
"A bear in a red forest"
"A car stuck in the forest"
"A bear in a red forest" AND "A car stuck in the forest"
(d) Composition fails
"A bird"
"A flower"
"A bird" AND "A flower"
"A couch"
"A dog sitting in the living room"
"A couch" AND "A dog sitting in the living room"

Fig. 6: **Qualitative results.** Successful examples (a) and failure examples (b-d) generated by the proposed method. There are three main types of failures: (b) The pre-trained diffusion model does not understand certain concepts, such as "person". (c) The pre-trained diffusion model confuses objects' attributes. (d) The composition fails. This usually happens when the objects are in the center of the images.

6.5 Results analysis

We show the image generation results conditioned on each individual sentence description, and our composition results in Figure 6. We provide four successful compositional examples, where the generated image contains all the concepts mentioned in the input sentences.

Failure cases. We observed three main failure cases of the proposed method. The first one is the pre-trained diffusion models do not understand certain concepts, such as "person" in (b). We used the pre-trained diffusion model, *GLIDE [30]*, which is trained to avoid generating human images. The second type of failure is because the diffusion models confuse the objects' attributes. In (c), the generated image contains "a red bear" while the input is "a bear in a red forest". The third type of failure is because the composition does not work, *e.g.* the “bird-shape and flower-color object” and the “dog-fur and sofa-shape object” in (d). Such failures usually happen when the objects are in the center of the images.

7 Conclusion

In this paper, we compose diffusion models for image generation. By interpreting diffusion models as energy-based models, we may explicitly compose them and generate images with significantly more complex combinations that are never seen during training. We propose two compositional operators, concept conjunction and negation, allowing us to compose diffusion models during the inference time without any additional training. The proposed composable diffusion models can generate images conditioned on sentence descriptions, objects, object relations, human facial attributes, and even generalize to new combinations that are rarely seen in the real world. These results demonstrate the effectiveness of the proposed method for compositional visual generation.

A limitation of our current approach is that while we are able to compose multiple diffusion models together, they are instances of the same model. We found limited success when composing diffusion models trained on different datasets. In contrast, compositional generation with EBMs [7] can successfully compose multiple separately trained models. Incorporating additional structures into diffusion models from EBMs [10], such as a conservative score field, may be a promising direction towards enabling compositions of separately trained diffusion models.

Acknowledgments. Shuang Li is supported by Raytheon BBN Technologies Corp. under the project Symbiant (reg. no. 030256-00001 90113), Mitsubishi Electric Research Laboratory (MERL) under the project Generative Models For Annotated Video, and CMU and Army Research Laboratory under the project Contrastive dissection to visualize the differences between synthetic and real trained representations (reg. no. 1130233-442111). Yilun Du is supported by a NSF Graduate Fellowship. We would like to thank David Bau for editting a preliminary draft of the manuscript and Mark Chen for providing DALLE-2 generations.

References

1. Austin, J., Johnson, D.D., Ho, J., Tarlow, D., van den Berg, R.: Structured denoising diffusion models in discrete state-spaces. In: Advances in Neural Information Processing Systems (2021)
2. Bau, D., Andonian, A., Cui, A., Park, Y., Jahanian, A., Oliva, A., Torralba, A.: Paint by word. arXiv preprint arXiv:2103.10951 (2021)
3. Chen, N., Zhang, Y., Zen, H., Weiss, R.J., Norouzi, M., Chan, W.: Wavegrad: Estimating gradients for waveform generation. arXiv preprint arXiv:2009.00713 (2020)
4. Chomsky, N.: Aspects of the Theory of Syntax. The MIT Press, Cambridge (1965), http://www.amazon.com/Aspects-Theory-Syntax-Noam-Chomsky/dp/0262530074
5. DCGM: Gender, age, and emotions extracted for flickr-faces-hq dataset (ffhq). https://github.com/DCGM/ffhq-features-dataset (2020)
6. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems **34** (2021)
7. Du, Y., Li, S., Mordatch, I.: Compositional visual generation with energy based models. Advances in Neural Information Processing Systems **33**, 6637–6647 (2020)
8. Du, Y., Li, S., Sharma, Y., Tenenbaum, J., Mordatch, I.: Unsupervised learning of compositional energy concepts. Advances in Neural Information Processing Systems **34** (2021)
9. Du, Y., Li, S., Tenenbaum, J., Mordatch, I.: Improved contrastive divergence training of energy based models. arXiv preprint arXiv:2012.01316 (2020)
10. Du, Y., Mordatch, I.: Implicit generation and generalization in energy-based models. arXiv preprint arXiv:1903.08689 (2019)
11. Gao, R., Song, Y., Poole, B., Wu, Y.N., Kingma, D.P.: Learning energy-based models by diffusion recovery likelihood. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=v_1Soh8QUNc
12. Grathwohl, W., Wang, K.C., Jacobsen, J.H., Duvenaud, D., Zemel, R.: Learning the stein discrepancy for training and evaluating energy-based models without sampling. In: International Conference on Machine Learning (2020)
13. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10696–10706 (2022)
14. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural computation **14**(8), 1771–1800 (2002)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
16. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021)
17. Janner, M., Du, Y., Tenenbaum, J., Levine, S.: Planning with diffusion for flexible behavior synthesis. In: International Conference on Machine Learning (2022)
18. Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2901–2910 (2017)
19. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)

20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
21. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020)
22. Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2426–2435 (June 2022)
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
24. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350**(6266), 1332–1338 (2015). https://doi.org/10.1126/science.aab3050, https://www.science.org/doi/abs/10.1126/science.aab3050
25. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predicting structured data **1**(0) (2006)
26. Liu, N., Li, S., Du, Y., Tenenbaum, J., Torralba, A.: Learning to compose visual relations. Advances in Neural Information Processing Systems **34** (2021)
27. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
28. Marcus, G., Davis, E., Aaronson, S.: A very preliminary analysis of dall-e 2. arXiv preprint arXiv:2204.13807 (2022)
29. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2021)
30. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
31. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. pp. 8162–8171. PMLR (2021)
32. Nie, W., Vahdat, A., Anandkumar, A.: Controllable and compositional generation with latent-space energy-based models. Advances in Neural Information Processing Systems **34** (2021)
33. Nijkamp, E., Hill, M., Han, T., Zhu, S.C., Wu, Y.N.: On the anatomy of mcmc-based maximum likelihood learning of energy-based models. arXiv preprint arXiv:1903.12370 (2019)
34. Parmar, G., Zhang, R., Zhu, J.Y.: On aliased resizing and surprising subtleties in gan evaluation. In: CVPR (2022)
35. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
36. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning. pp. 8821–8831. PMLR (2021)
37. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
38. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

39. Saharia, C., Chan, W., Chang, H., Lee, C.A., Ho, J., Salimans, T., Fleet, D.J., Norouzi, M.: Palette: Image-to-image diffusion models. arXiv preprint arXiv:2111.05826 (2021)
40. Salimans, T., Ho, J.: Should ebms model the energy or the score? In: Energy Based Models Workshop-ICLR 2021 (2021)
41. Shoshan, A., Bhonker, N., Kviatkovsky, I., Medioni, G.: Gan-control: Explicitly controllable gans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14083–14093 (2021)
42. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning. pp. 2256–2265. PMLR (2015)
43. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021)
44. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
45. Swimmer963: What dall-e 2 can and cannot do (May 2022), https://www.lesswrong.com/posts/uKp6tBFStnsvrot5t/what-dall-e-2-can-and-cannot-do
46. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
47. Vincent, P.: A connection between score matching and denoising autoencoders. Neural computation **23**(7), 1661–1674 (2011)
48. Xiao, T., Hong, J., Ma, J.: Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In: Proceedings of the European conference on computer vision (ECCV). pp. 168–184 (2018)
49. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1316–1324 (2018)
50. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 5907–5915 (2017)
51. Zhou, L., Du, Y., Wu, J.: 3D shape generation and completion through point-voxel diffusion. In: International Conference on Computer Vision (2021)
52. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: European conference on computer vision. pp. 592–608. Springer (2020)

Appendix

In this appendix, we first provide additional results in appendix A. We then show the details of training classifiers in appendix B. In appendix C and appendix D, we show more details of our approach and baselines, respectively. Finally, we provide the implementation details in appendix E.

A Additional Results

In this section, we first show results of composing human facial attributes in appendix A.1 as we described in the main paper Section 6. We then show more qualitative results in appendix A.2.

A.1 Composing Human Facial Attributes

Qualitative results. We compare the proposed method and baselines on compositing facial attributes in Figure 7. We find that *LACE* and *StyleGAN2* can generate high-fidelity images, but the generated images do not match the given label. For example, *StyleGAN2* generates humans without wearing glasses when the input labels contain "glasses". *LACE* generates males sometimes when the input is "NOT Male". The image quality of *EBM* is much worse than other methods. In contrast, our method can generate high-fidelity images, containing all the attributes in the input label.
Quantitative results. The results of our method and baselines on three test settings are shown in Table 3. Our method is comparable with the best baseline on each evaluation metric.

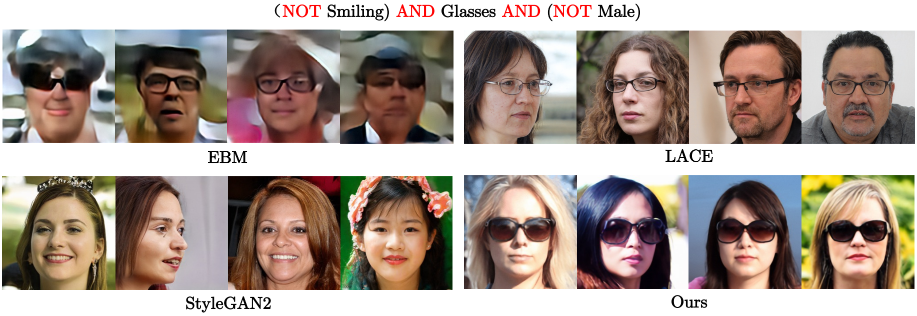
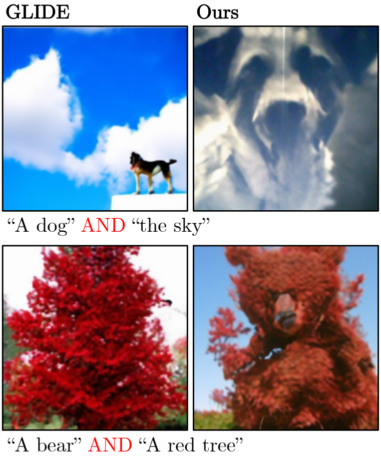


Fig. 7: **Composing Facial Attributes.** Image generation results on the FFHQ dataset. Our model is trained to generate images conditioned on a single human facial attribute, but during inference, our model can recursively compose multiple facial attributes using the proposed compositional operators. The baselines either fail to compose attributes (StyleGAN2 and LACE) or generate low-quality images (EBM).

A.2 More Qualitative Results

Table 3: Image generation results on FFHQ. The binary classification accuracy (Acc) and FID are reported. Our method achieves comparable results with the best baselines on three test settings.

Models	1 Component		2 Components		3 Components	
	Acc (%) ↑	FID ↓	Acc (%) ↑	FID ↓	Acc (%) ↑	FID ↓
EBM [7]	98.74	89.95	93.10	99.64	30.01	335.70
StyleGAN2 [21]	58.90	**18.04**	30.68	18.06	16.96	18.06
LACE [32]	97.60	28.21	**95.66**	36.23	**80.88**	34.64
GLIDE [30]	98.66	20.30	48.68	22.69	27.24	21.98
Ours	**99.26**	18.72	92.68	**17.22**	68.86	**16.95**



“A dog” AND “the sky”

“A bear” AND “A red tree”

Fig. 8: Our method (composing multiple sentences) generates different styles of images compare to *GLIDE* (directly encodes the descriptions as a single long sentence).

We provide more qualitative results of the proposed method on composing concepts using the conjunction operator. Figure 9, 10, 11 and 12 shows the results of compositing language descriptions. Figure 13 shows additional results on compositing objects on the CLEVR dataset. Our approach can reliably generate images conditioned on multiple concepts, even for combinations outside the training distribution

We further show results of compositing facial attributes on the FFHQ dataset in Figure 14. Our model is trained to generate images conditioned on a single human facial attribute, but it can compose multiple attributes during inference without further training using the conjunction and negation compositional operators. As shown in the fifth row of Figure 14, our model can compose *Not Male* and *Glasses* and generate images with females wearing glasses. The proposed compositional operators allow our model to compose facial attributes recursively.

Interesting cases. As shown in Figure 8, we find that our method, which combines multiple textual descriptions, can generate different styles of images compared to *GLIDE*, which directly encodes the descriptions as a single long sentence. Prompted with “a dog” and “the sky”, our method generates a dog-shaped cloud, whereas *GLIDE* generates a dog under the sky from the prompt “a dog and the sky”.

B Details of Binary Classifiers

We provide more details of the binary classifiers in this section.

CLEVR. CLEVR dataset consists of 30,000 image-label pairs. We split the dataset into training and validation subsets. There are $24,000$ data pairs used for training and $6,000$ data pairs used for validation. We train a binary classifier to evaluate whether there is an object appearing at a particular position of an image. The classifier achieves 99.05% accuracy on the validation set, which is used to evaluate the quality of generated images.

Relational CLEVR. Relational CLEVR [26] contains $50,000$ images at 128×128 resolution. We split the dataset into $40,000$ training data and $10,000$ validation data. Then we train a binary classifier to evaluate whether an image contains an object relational description. The classifier achieves 99.80% accuracy on the validation set.

FFHQ. We use $30,000$ image-label pairs from CelebA-HQ [19] to train a classifier for FFHQ generated images. We split the dataset into training and validation subsets using $80:20$ ratio. We select three attributes (*i.e. smiling*, *glasses*, and *gender*) to evaluate the compositionality ability of our approach and baselines. We thus train three binary classifiers to evaluate the *smiling*, *glasses*, and *gender* concepts respectively. Our classifiers achieve 95.01%, 99.20% and 97.49% accuracy on the validation sets of *smiling*, *glasses*, and *gender*.

To further verify the reliability of results obtained by the classifiers, we add human evaluation results and find that our method still outperforms baselines. We generated 300 facial images using our method and one of the best baselines (LACE), respectively. Given a concept combination, *e.g. Smiling AND (NOT Male)*, each method generates an image conditioned on this combination. We asked workers to select which image matches the input concepts the best. At 62% of the time, the workers think the images generated by our method are better.

C Details of Our Approach

Training. Our approach is implemented based on the code from [31,30]. Ho *et al.* [16] introduce a technique to train a conditional and an unconditional diffusion model at the same time by masking some labels as null labels. During training, we utilize the same approach. We randomly replace 10% of training labels as the null labels in our training to estimate the unconditional score and otherwise use conditional labels.

Inference. To generate images, we compute the unconditional and conditional scores for each label and use the combined score to sample a less noisy image at each timestep. To generate FFHQ images, we first generate images at 64×64 resolution and then upsample the images to 256×256. For CLEVR images, we generate images at 128×128 resolution directly.

Label Encoding. On the FFHQ dataset, we use three attributes, including *smile*, *glasses* and *gender*. For the *smile* and *glasses* attributes, label 1 indicates that the image contains the attribute, and label 0 indicates its absence. For the *gender* attribute, label 0 indicates "male", while label 1 represents "female". We use the embedding layer $nn.Embedding(7, d)$ to encode the attribute labels, including 6 attribute labels and 1 null class label. The labels are encoded as a d-dimension feature vector, which is then fused with the embedding of the iteration step t and image x_t. The fused features are sent to the U-Net [38] during training.

On the CLEVR dataset, we encode the (x, y) coordinates using a linear layer $nn.Linear(2, d)$, where 2 is the dimension of the (x, y) coordinates and d is the dimension of the hidden feature. The coordinates embedding is then fused with the embedding of the iteration step t and image x_t, which are further sent to the U-Net [38] during training.

D Details of Baselines

Energy-based models (EBMs). We train energy-based models using the codebase from [9], where we encode discrete labels and continuous labels using an embedding layer and a linear layer, respectively. We use the inference code from [7] to compose multiple concepts.

StyleGAN2. We train an unconditional StyleGAN2 on CLEVR, while we use an existing StyleGAN2 model trained on FFHQ. For training, we use the "config-f" setting provided by [21]. To enable image generation conditioned on multiple concepts, we train a binary classifier on each dataset. During inference, we optimize the underlying latent code to minimize each loss from the classifier conditioned on each individual label.

LACE. LACE [32] trains classifiers for image generation by using sampled images from StyleGAN2 and labels provided by the neural network. For CLEVR dataset, we firstly generate $10,000$ images using the same StyleGAN2 model that was trained on CLEVR in Section D. Then we modify the code to train a position annotator using a DenseNet model provided by LACE to label the positions of generated images. Lastly, we train a classifier conditioned on coordinates using their provided script. For FFHQ, we use their off-the-shelf pre-trained model for comparison. To enable image generation, we utilize their inference scripts.

GLIDE. We use the released GLIDE [30] model in our experiments. We develop Composed GLIDE (Ours), a version of GLIDE that utilizes our compositional operators to combine textual descriptions, without further training. We compare it to the original GLIDE, which directly encodes the descriptions as a single long sentence. [30] also released a upsample model to upsample the generated images to a resolution of 256×256. We use the upsample model for both the GLIDE and Composed GLIDE (Ours).

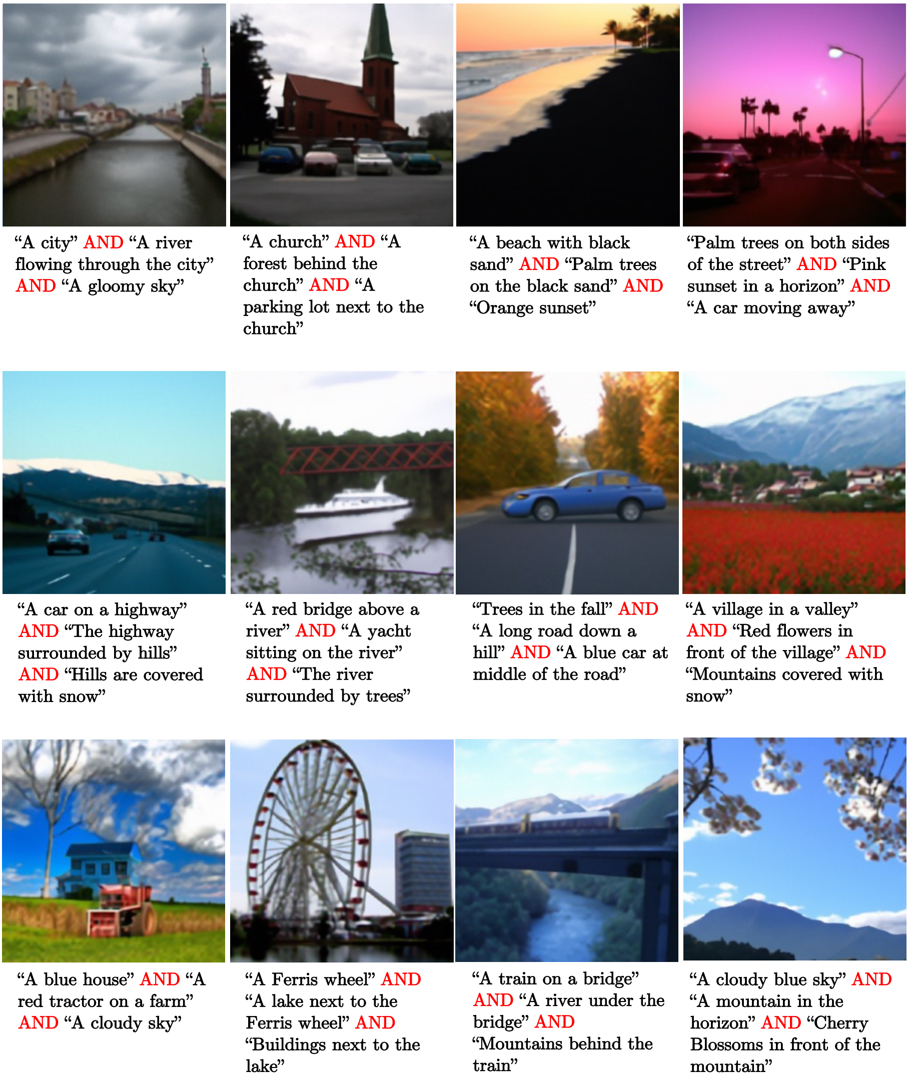
E Implementation Details

EBMs. In our experiments, we use the same setting to train models on different datasets. We use the Adam optimizer [23] with a learning rate of 10^{-4}. For MCMC sampling, we use a step size of 300 and 80 iterations. On each dataset, the model is trained for two days on a single Tesla 32GB GPU.

StyleGAN2. We train the StyleGAN2 model for 2 days on CLEVR using a single Tesla 32GB GPU. It takes 2 hours to train binary classifiers for each dataset. We use the Adam optimizer [23] with $\beta_1 = 0$, $\beta_2 = 0.99$, and $\epsilon = 10^{-8}$ to train the StyleGAN2 model (more details can be found in the codebase from [21]). We use the Adam optimizer with $\beta_1 = 0$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ to train the classifiers. We use the pre-trained model provided by [21] on the FFHQ dataset.

LACE. LACE uses the pre-trained model provided by [21] on the FFHQ dataset as well. For CLEVR, we use the same StyleGAN2 model as described in Section E. It takes less than 10 minutes to train the classifier on each dataset using a single Tesla 32GB GPU.

Our Approach. To train diffusion models on both CLEVR and FFHQ, we use $1,000$ diffusion steps, and the cosine noise schedule. We use the AdamW optimizer [27] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We train the diffusion models on CLEVR and FFHQ for 7 days ($750,000$ iterations) and 2 days ($250,000$ iterations), respectively, using a single Tesla 32GB GPU.



"A city" AND "A river flowing through the city" AND "A gloomy sky"

"A church" AND "A forest behind the church" AND "A parking lot next to the church"

"A beach with black sand" AND "Palm trees on the black sand" AND "Orange sunset"

"Palm trees on both sides of the street" AND "Pink sunset in a horizon" AND "A car moving away"

"A car on a highway" AND "The highway surrounded by hills" AND "Hills are covered with snow"

"A red bridge above a river" AND "A yacht sitting on the river" AND "The river surrounded by trees"

"Trees in the fall" AND "A long road down a hill" AND "A blue car at middle of the road"

"A village in a valley" AND "Red flowers in front of the village" AND "Mountains covered with snow"

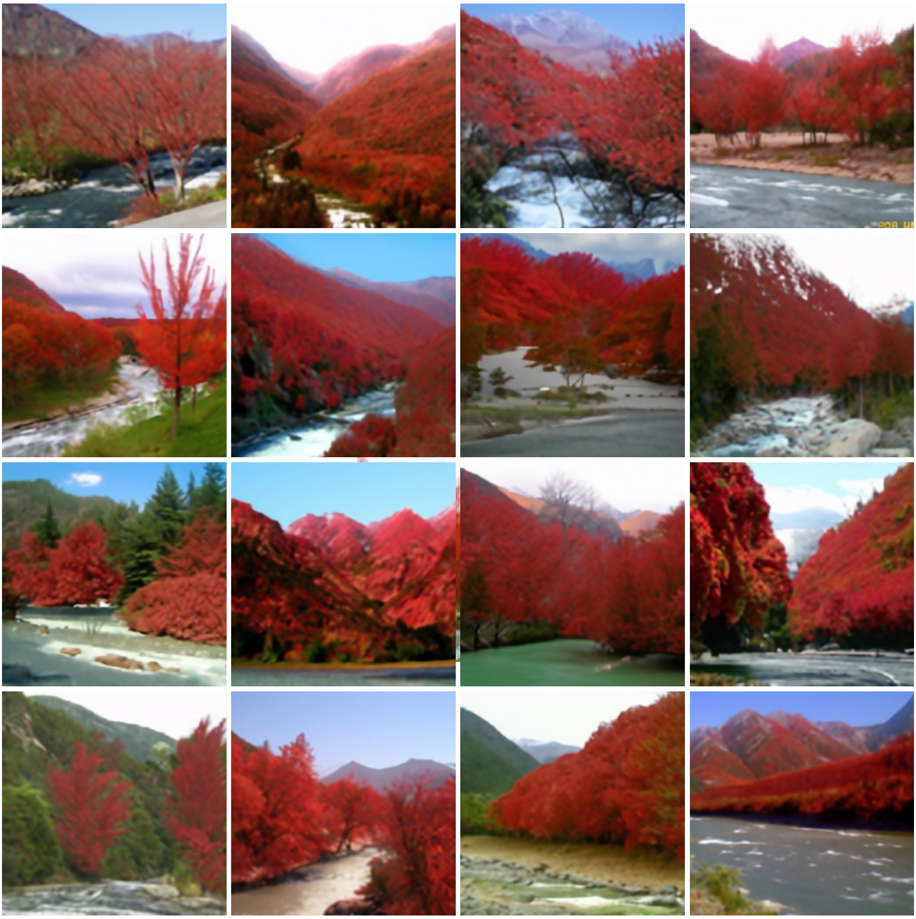
"A blue house" AND "A red tractor on a farm" AND "A cloudy sky"

"A Ferris wheel" AND "A lake next to the Ferris wheel" AND "Buildings next to the lake"

"A train on a bridge" AND "A river under the bridge" AND "Mountains behind the train"

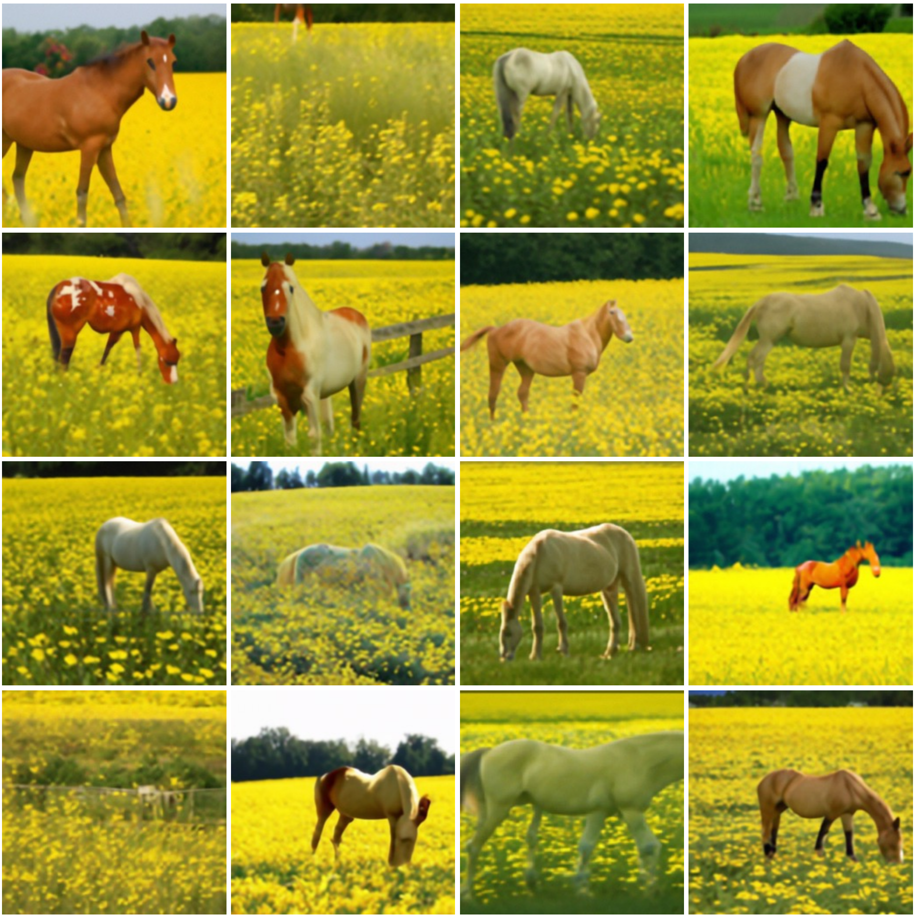
"A cloudy blue sky" AND "A mountain in the horizon" AND "Cherry Blossoms in front of the mountain"

Fig. 9: **Composing Language Descriptions**. We provide more qualitative results of *Composed GLIDE (Ours)*, a version of GLIDE [30] that utilizes our compositional operators to combine textual descriptions, without further training.



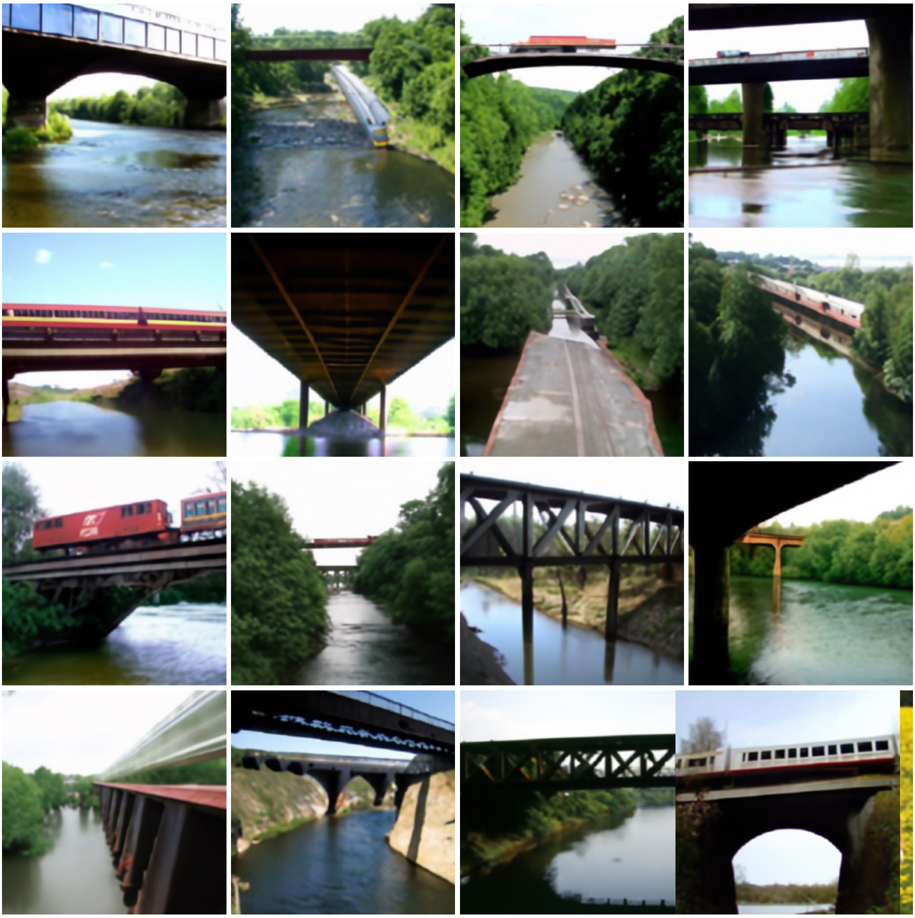
“A river leading into mountains” AND “red trees on the side”

Fig. 10: **Composing Language Descriptions**. Images generated by our method, *Composed GLIDE (Ours)*.



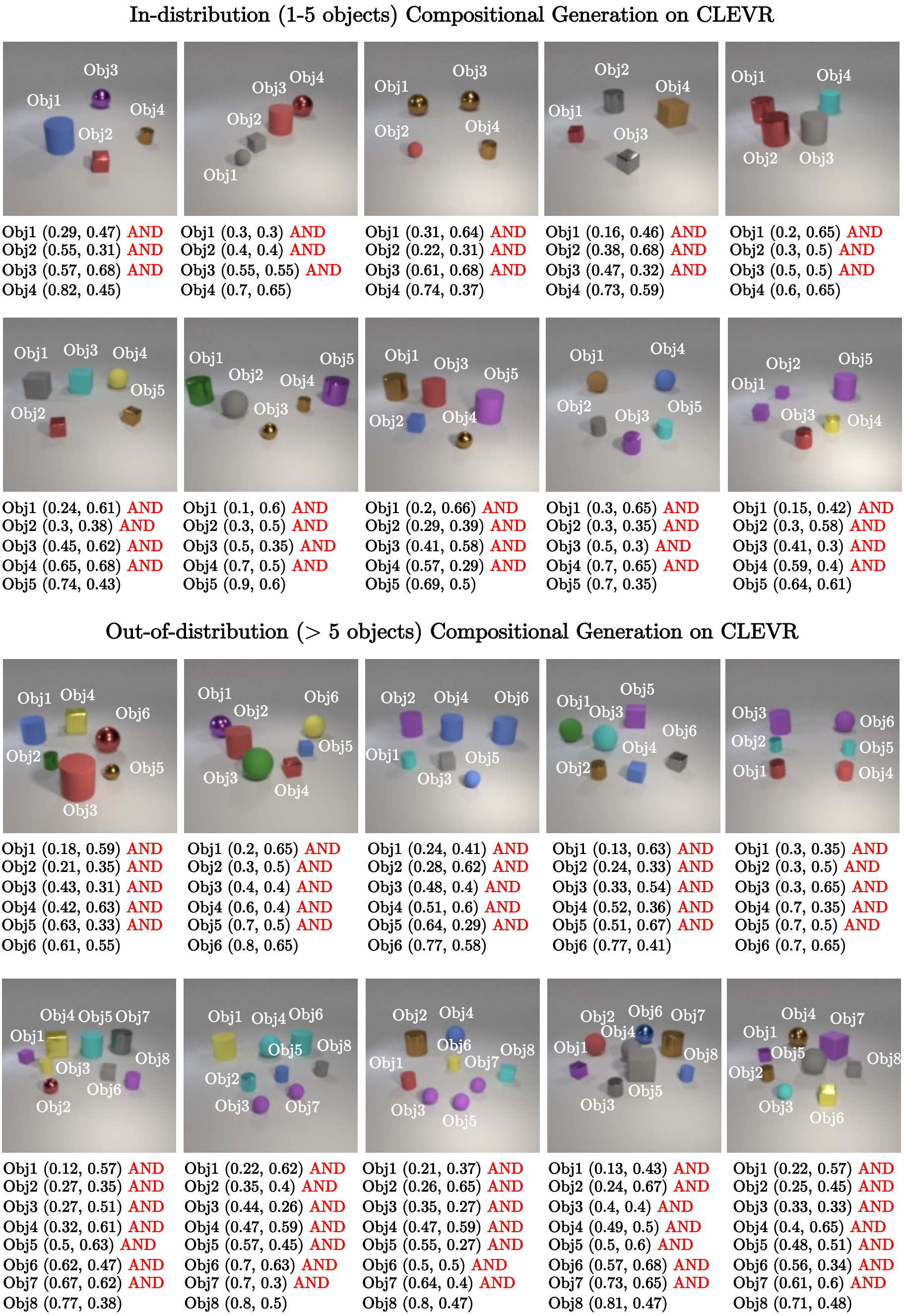
“A horse” AND “a yellow flower field”

Fig. 11: **Composing Language Descriptions**. Images generated by our method, *Composed GLIDE (Ours)*.



“A train on a bridge” AND “A river under the bridge”

Fig. 12: **Composing Language Descriptions**. Images generated by our method, *Composed GLIDE (Ours)*.


In-distribution (1-5 objects) Compositional Generation on CLEVR
Obj3 Obj1 Obj4 Obj2
Obj4 Obj3 Obj2 Obj1
Obj3 Obj1 Obj2 Obj4
Obj2 Obj4 Obj1 Obj3
Obj1 Obj4 Obj2 Obj3
Obj1 Obj3 Obj4 Obj5 Obj2
Obj1 Obj5 Obj2 Obj4 Obj3
Obj1 Obj3 Obj5 Obj2 Obj4
Obj1 Obj4 Obj2 Obj3 Obj5
Obj2 Obj5 Obj1 Obj3 Obj4
Out-of-distribution (> 5 objects) Compositional Generation on CLEVR
Obj1 Obj4 Obj6 Obj2 Obj5 Obj3
Obj1 Obj2 Obj6 Obj5 Obj3 Obj4
Obj2 Obj4 Obj6 Obj1 Obj5 Obj3
Obj1 Obj5 Obj3 Obj6 Obj4 Obj2
Obj3 Obj6 Obj2 Obj5 Obj1 Obj4
Obj4 Obj5 Obj7 Obj1 Obj8 Obj3 Obj6 Obj2
Obj1 Obj4 Obj6 Obj5 Obj8 Obj2 Obj3 Obj7
Obj2 Obj4 Obj6 Obj8 Obj1 Obj7 Obj3 Obj5
Obj2 Obj6 Obj7 Obj4 Obj1 Obj8 Obj5 Obj3
Obj4 Obj7 Obj1 Obj5 Obj8 Obj2 Obj3 Obj6

Obj1 (0.29, 0.47) AND
Obj2 (0.55, 0.31) AND
Obj3 (0.57, 0.68) AND
Obj4 (0.82, 0.45)

Obj1 (0.3, 0.3) AND
Obj2 (0.4, 0.4) AND
Obj3 (0.55, 0.55) AND
Obj4 (0.7, 0.65)

Obj1 (0.31, 0.64) AND
Obj2 (0.22, 0.31) AND
Obj3 (0.61, 0.68) AND
Obj4 (0.74, 0.37)

Obj1 (0.16, 0.46) AND
Obj2 (0.38, 0.68) AND
Obj3 (0.47, 0.32) AND
Obj4 (0.73, 0.59)

Obj1 (0.2, 0.65) AND
Obj2 (0.3, 0.5) AND
Obj3 (0.5, 0.5) AND
Obj4 (0.6, 0.65)

Obj1 (0.24, 0.61) AND
Obj2 (0.3, 0.38) AND
Obj3 (0.45, 0.62) AND
Obj4 (0.65, 0.68) AND
Obj5 (0.74, 0.43)

Obj1 (0.1, 0.6) AND
Obj2 (0.3, 0.5) AND
Obj3 (0.5, 0.35) AND
Obj4 (0.7, 0.5) AND
Obj5 (0.9, 0.6)

Obj1 (0.2, 0.66) AND
Obj2 (0.29, 0.39) AND
Obj3 (0.41, 0.58) AND
Obj4 (0.57, 0.29) AND
Obj5 (0.69, 0.5)

Obj1 (0.3, 0.65) AND
Obj2 (0.3, 0.35) AND
Obj3 (0.5, 0.3) AND
Obj4 (0.7, 0.65) AND
Obj5 (0.7, 0.35)

Obj1 (0.15, 0.42) AND
Obj2 (0.3, 0.58) AND
Obj3 (0.41, 0.3) AND
Obj4 (0.59, 0.4) AND
Obj5 (0.64, 0.61)

Obj1 (0.18, 0.59) AND
Obj2 (0.21, 0.35) AND
Obj3 (0.43, 0.31) AND
Obj4 (0.42, 0.63) AND
Obj5 (0.63, 0.33) AND
Obj6 (0.61, 0.55)

Obj1 (0.2, 0.65) AND
Obj2 (0.3, 0.5) AND
Obj3 (0.4, 0.4) AND
Obj4 (0.6, 0.4) AND
Obj5 (0.7, 0.5) AND
Obj6 (0.8, 0.65)

Obj1 (0.24, 0.41) AND
Obj2 (0.28, 0.62) AND
Obj3 (0.48, 0.4) AND
Obj4 (0.51, 0.6) AND
Obj5 (0.64, 0.29) AND
Obj6 (0.77, 0.58)

Obj1 (0.13, 0.63) AND
Obj2 (0.24, 0.33) AND
Obj3 (0.33, 0.54) AND
Obj4 (0.52, 0.36) AND
Obj5 (0.51, 0.67) AND
Obj6 (0.77, 0.41)

Obj1 (0.3, 0.35) AND
Obj2 (0.3, 0.5) AND
Obj3 (0.3, 0.65) AND
Obj4 (0.7, 0.35) AND
Obj5 (0.7, 0.5) AND
Obj6 (0.7, 0.65)

Obj1 (0.12, 0.57) AND
Obj2 (0.27, 0.35) AND
Obj3 (0.27, 0.51) AND
Obj4 (0.32, 0.61) AND
Obj5 (0.5, 0.63) AND
Obj6 (0.62, 0.47) AND
Obj7 (0.67, 0.62) AND
Obj8 (0.77, 0.38)

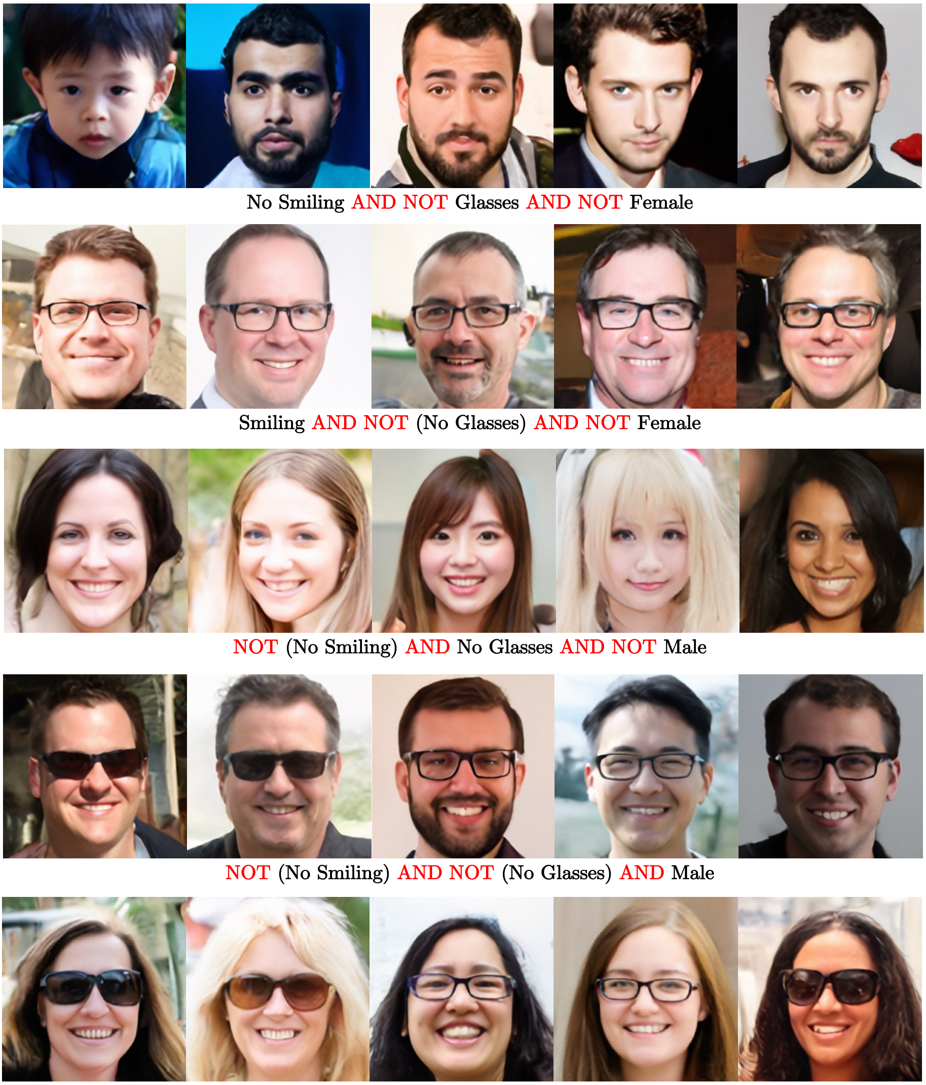
Obj1 (0.22, 0.62) AND
Obj2 (0.35, 0.4) AND
Obj3 (0.44, 0.26) AND
Obj4 (0.47, 0.59) AND
Obj5 (0.57, 0.45) AND
Obj6 (0.7, 0.63) AND
Obj7 (0.7, 0.3) AND
Obj8 (0.8, 0.5)

Obj1 (0.21, 0.37) AND
Obj2 (0.26, 0.65) AND
Obj3 (0.35, 0.27) AND
Obj4 (0.47, 0.59) AND
Obj5 (0.55, 0.27) AND
Obj6 (0.5, 0.5) AND
Obj7 (0.64, 0.4) AND
Obj8 (0.8, 0.47)

Obj1 (0.13, 0.43) AND
Obj2 (0.24, 0.67) AND
Obj3 (0.4, 0.4) AND
Obj4 (0.49, 0.5) AND
Obj5 (0.5, 0.6) AND
Obj6 (0.57, 0.68) AND
Obj7 (0.73, 0.65) AND
Obj8 (0.81, 0.47)

Obj1 (0.22, 0.57) AND
Obj2 (0.25, 0.45) AND
Obj3 (0.33, 0.33) AND
Obj4 (0.4, 0.65) AND
Obj5 (0.48, 0.51) AND
Obj6 (0.56, 0.34) AND
Obj7 (0.61, 0.6) AND
Obj8 (0.71, 0.48)

Fig. 13: **Composing Objects.** During inference, our model can generate images that contain multiple objects by composing their probability distributions using the conjunction operator. Note that the training set only contains images with fewer than 5 objects, but our model can compose more than 5 objects during inference.



No Smiling AND NOT Glasses AND NOT Female

Smiling AND NOT (No Glasses) AND NOT Female

NOT (No Smiling) AND No Glasses AND NOT Male

NOT (No Smiling) AND NOT (No Glasses) AND Male

Smiling AND NOT (No Glasses) AND NOT Male

Fig. 14: **Composing Facial Attributes.** During inference, our model can generate images that contain multiple attributes by composing their probability distributions using the negation and conjunction operators.