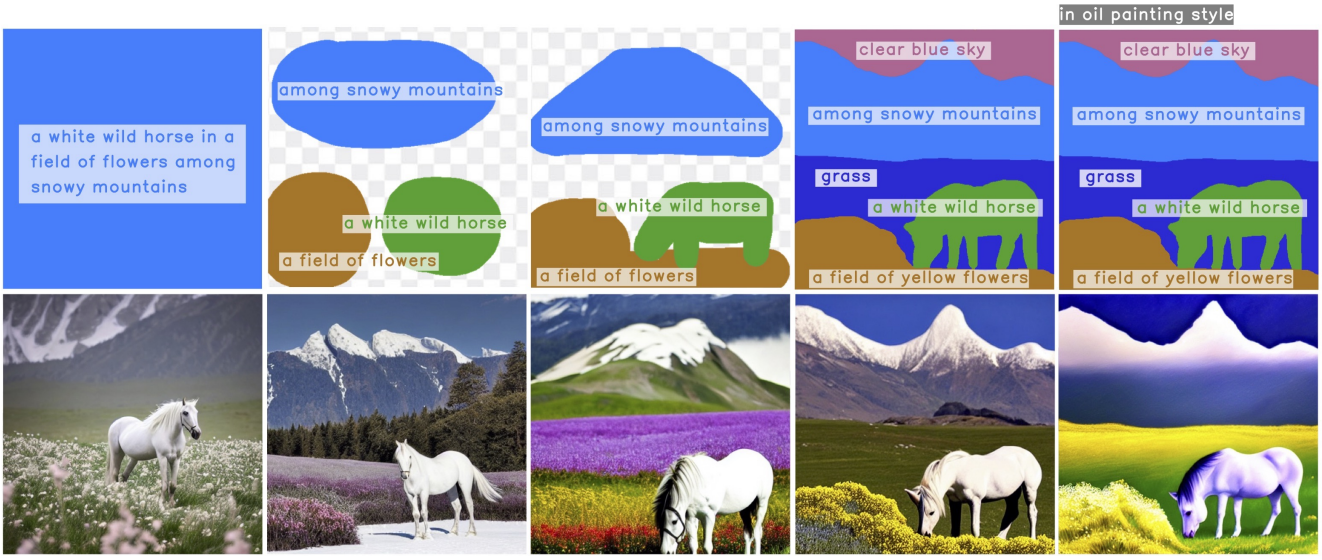
SceneComposer: Any-Level Semantic Image Synthesis

Yu Zeng[1], Zhe Lin[2], Jianming Zhang[2], Qing Liu[2], John Collomosse[2], Jason Kuen[2], Vishal M. Patel[1]
{yzeng22,vpatel36}@jhu.edu, {zlin,jianmzha,qingl,collomos,kuen}@adobe.com
[1]Johns Hopkins University [2]Adobe Research



(a) (b) (c) (d) (e)

Figure 1. Examples of image synthesis from any-level semantic layouts. (a) The coarsest layout, *i.e.* at the 0-th precision level, is equivalent to a text input; (d) the finest layout, *i.e.* at the highest level, is close to an accurate segmentation map; (a)-(d) intermediate level layouts (from coarse to fine), the shape control becomes tighter with increasing levels. (e) We can specify different precision levels for different components, *e.g.* to include a 0-th level style indicator while the remaining regions are of higher levels.

Abstract

We propose a new framework for conditional image synthesis from semantic layouts of any precision levels, ranging from pure text to a 2D semantic canvas with precise shapes. More specifically, the input layout consists of one or more semantic regions with free-form text descriptions and adjustable precision levels, which can be set based on the desired controllability. The framework naturally reduces to text-to-image (T2I) at the lowest level with no shape information, and it becomes segmentation-to-image (S2I) at the highest level. By supporting the levels in-between, our framework is flexible in assisting users of different drawing expertise and at different stages of their creative workflow. We introduce several novel techniques to address the challenges coming with this new setup, including a pipeline for collecting training data; a precision-encoded mask pyramid and a text feature map representation to jointly encode precision level, semantics, and composition information; and a multi-scale guided diffusion model to synthesize images. To evaluate the proposed method, we collect a test dataset containing user-drawn layouts with diverse scenes and styles. Experimental results show that the proposed method can generate high-quality images following the layout at given precision, and compares favorably against existing methods. Project page https://zengxianyu.github.io/scenec/

1. Introduction

Recently, deep generative models such as StyleGAN [24, 25] and diffusion models [9, 19, 49] have made a significant breakthrough in generating high-quality images. Image generation and editing technologies enabled by these models have become highly appealing to artists and designers by helping their creative workflows. To make image generation more controllable, researchers have put a lot of effort into conditional image synthesis and introduced models using various types and levels of semantic input such as object categories, text prompts, and segmentation

Table 1. Difference from related conditional image synthesis works. T2I: text to image, S2I: segmentation to image, ST2I: Scene-based text to image [13], Box2I: bounding box layout to image [50].

Setting	Open-domain layout	Shape control	Sparse layout	Coarse shape	Level control
T2I	✓	✗	✗	✗	✗
S2I	✗	✓	✗	✗	✗
ST2I	✗	✓	✗	✗	✗
Box2I	✗	✗	✓	✗	✗
Ours	✓	✓	✓	✓	✓

maps *etc*. [23, 35, 36, 43, 44, 67].

However, existing models are not flexible enough to support the full creative workflow. They mostly consider fixed-level semantics as the input [63], *e.g*. image-level text descriptions in text-to-image generation (T2I) [35, 39, 41, 43, 44], or pixel-level segmentation maps in segmentation-to-image generation (S2I) [23, 36, 67]. Recent breakthroughs on T2I such as DALLE2 [39] and StableDiffusion [1, 43] demonstrate extraordinary capabilities of generating high-quality results. They can convert a rough idea into visual messages to provide inspirations at the beginning of the creative process, but provide no further control over image composition. On the other hand, S2I allows users to precisely control the image composition. As it is extremely challenging to draw a detailed layout directly, S2I is more useful for later creative stages given initial designs. For real-world use cases, it is highly desirable to have a model which can generate images from not only pure text or segmentation maps, but also intermediate-level layouts with coarse shapes.

To this end, we propose a new unified conditional image synthesis framework to generate images from a semantic layout at any combination of precision levels. It is inspired by the typical coarse-to-fine workflow of artists and designers: they first start from an idea, which can be expressed as a text prompt or a set of concepts (Fig. 1 (a)), then tend to draw the approximate outlines and refine each object (Fig. 1 (a)-(d)). More specifically, we model a semantic layout as a set of semantic regions with free-form text descriptions. The layout can be sparse and each region can have a precision level to control how well the generated object should fit to the specified shape. The framework reduces to T2I when the layout is the coarsest (Fig. 1 (a)), and it becomes S2I when the layout is a segmentation map (Fig. 1 (d)). By adjusting the precision level, users can achieve their desired controllability (Fig. 1 (a)-(d)). This framework is different from the existing works in many aspects, as summarized in Table 1.

This new setup comes with several challenges. First, it is non-trivial to encode open-domain layouts in image synthesis frameworks. Second, to handle hand-drawn layouts of varying precision, we need an effective and robust way to inject the precision information into the layout encoding. Third, there is no large-scale open-domain layout/image dataset. To generate high-quality images and generalize to novel concepts, a large and diverse training dataset is crucial.

We introduce several novel ideas to address these challenges. First, we propose a text feature map representation for encoding a semantic layout. It can be seen as a spatial extension of text embedding or generalization of segmentation masks from binary to continuous space. Second, we introduce a precision-encoded mask pyramid to model layout precision. Inspired by the classical image pyramid models [2, 6, 47, 62], we relate shape precision to levels in a pyramid representation and encode precision by dropping out regions of lower precision levels. In other words, the l-th level of the mask pyramid is a sub-layout (subset of regions) consisting of semantic regions with precision level no less than l. By creating a text feature map for each sub-layout, we obtain a text feature pyramid as a unified representation of semantics, composition, and precision. Finally, we feed the text feature pyramid to a multi-scale guided diffusion model to generate images. We fulfill the need for training data by collecting them from two sources: (1) large-scale image-text pairs; (2) a relatively small pseudo layout/image dataset using text-based object detection and segmentation. With this multi-source training strategy, both text-to-image and layout-to-image can benefit from each other synergistically.

Our contributions are summarized as follows:

- A unified framework for diffusion-based image synthesis from semantic layouts with any combination of precision control.
- Novel ideas to build the model, including precision-encoded mask pyramid and pyramid text feature map representation, and multi-scale guided diffusion model, and training with multi-source data.
- A new real-world user-drawn layout dataset and extensive experiments showing the effectiveness of our model for text-to-image and layout-to-image generation with precision control.

2. Related Work

Deep generative models. In recent years, there has been significant progress in image generation using deep generative models. Some of these approaches attempt to learn the image distribution by optimizing a likelihood-based objective function. Autoregressive models (ARMs) [11, 55–57] treat images as sequences of pixels or tokens in a learned dictionary to define a tractable density model, which can be optimized by maximizing the likelihood. Variational Autoencoders (VAEs) [26, 54] use an intractable density function and train the model by maximizing the variational lower bound. Diffusion-based models [9, 19, 43, 48, 49] are also trained by optimizing the variational lower bound. Unlike VAEs which map an image into the latent space using a learnable encoder, diffusion-based models use a fixed diffusion process to transform an image into Gaussian noise. Generative Adversarial Networks (GANs) [5, 14, 24, 25] do not define the density function explicitly; instead, generative

models are trained through an adversarial learning process against a discriminator. We base our method on diffusion models due to their ability to generate high-quality images and stability in training.

Layout/Segmentation-to-image generation refers to the task of generating images from a spatial arrangement of semantic concepts, *e.g*. a segmentation map or a set of bounding boxes. Segmentation-based methods (*e.g*. [8,23,36,45,52,59,60]) have been proposed to allow users to precisely control the image composition, but are less flexible as they require a dense and accurate segmentation map and only allow a fixed set of categories. Bounding box-based methods can generate images from a coarse layout of bounding boxes. Bounding box-based generation was studied as a standalone task first by Zhao *et al*. [66] and has been an active research area [3, 17, 30, 58] since then. To convey more information than object categories, more expressive bounding box descriptions have been introduced, such as attributes [3, 34], relations [3], and free-form descriptions [12]. Although bounding boxes are easier to draw than segmentation maps, they provide no control over object shapes and orientations. In this work, we use coarse shapes with free-form text descriptions and precision levels to represent a layout, providing more flexibility and control for different synthesis needs.

Text-to-image/Multi-modal image generation. T2I aims to generate images from a free-form text description, *i.e*. image captions. Earlier approaches are usually based on conditional GANs [41, 65]. Recently, autoregressive models based on image quantization techniques [10, 40, 64] and diffusion models [1,35,39,43,44] have shown surprising results with large-scale models and datasets. To gain more control over the spatial composition, some studies have tried to incorporate layout in the form of key points [42], bounding boxes [18], or object shapes [13, 21, 27, 29, 37], as an additional input. Some studies focused on image synthesis from multi-modal input including text, segmentation maps, edge maps [22, 61] *etc*. A work [4] concurrent to ours proposes a training-free paint-with-word method that enables users to specify the object locations by manipulating the attention matrices in cross-attention layers. These approaches use the layout as an extra signal complementary to captions. In contrast, we unify image captions and layout using a multi-level framework that models captions as a special case of layout with the lowest precision level. From a practical perspective, our framework provides a more flexible control mechanism with coarse shapes and precision controllability.

3. Proposed Method

Our method aims to generate images from a layout consisting of a set of semantic regions of varying precision levels. Formally, an input layout can be viewed as a list of tuples $\{(M_i, t_i, c_i)\}_{i=1}^{n}$, where M_i, t_i, c_i indicate the segmentation mask, text description, and the precision level of the i-th region. We let $\boldsymbol{M}, \boldsymbol{t}, \boldsymbol{c}$ denote the sets of masks, texts and precision levels of all regions, respectively. The precision level variable $c_i \in \mathbb{N}_{\leq L}$ indicates how precisely the generated content should follow the mask M_i. A smaller value of c_i indicates a less precise control allowing more deviation from the mask M_i. $c_i = 0$ indicates the coarsest level where the i-th mask will be ignored. When all $c_i = 0$, the problem converts into T2I. Fig. 2 shows an overview of the proposed method. To generate an image at resolution $2^L \times 2^L$, we first formulate a precision-encoded mask pyramid $\{\boldsymbol{M}^l\}_{l=0}^{L}$ which represents each mask at the given precision level (Sec. 3.1). Then we combine the mask pyramid with the text descriptions $\boldsymbol{t}$ to form a text feature pyramid $\{\boldsymbol{Z}^l\}_{l=0}^{L}$. It contains a $2^l \times 2^l$ text feature map at each level, which can be seen as an extension of the one-hot label map encoding in S2I (Sec. 3.2). Finally, a multi-level guided diffusion model takes the feature pyramid as input to generate an image (Sec. 3.3).

3.1. Precision-Encoded Mask Pyramid

For a user-drawn coarse shape, it is challenging to model the exact precision, since the type and the amount of error vary across different users. To simplify the problem, we relate precision to resolutions and propose a precision-encoded mask pyramid to encode the shape and precision information simultaneously. Given a mask and a precision level, a high precision level corresponds to using all details of the masks at a high resolution, while a low precision level means we can only trust the mask at a low resolution.

More specifically, given a set of masks $\boldsymbol{M} = [m_1, .., m_n]$ and the precision levels $\boldsymbol{c} = [c_1, ..., c_n]$, we construct the mask pyramid by representing each mask M_i at resolutions up to 2^{c_i}. In other words, we resize each mask M_i to multiple resolutions and drop out the masks at resolutions higher than 2^{c_i}, as illustrated in Fig. 3. Formally, the l-th layer of the mask pyramid, denoted as $\boldsymbol{M}^l$, is computed as follows,

$$M_i^l = \text{resize}(M_i, 2^l) \cdot \mathbb{1}_{c_i \geq l}, \tag{1}$$

where $\text{resize}(M_i, 2^l)$ resizes a mask M_i to $2^l \times 2^l$ resolution through image interpolation and binarization. The indicator function $\mathbb{1}_{c_i \geq l}$ returns 1 when $c_i \geq l$ and 0 otherwise. In our implementation, we set the range of precision levels to $[0, 3, 4, 5, 6]$. We observed that for a free-hand drawn layout, 8×8 masks at the level 3 are easy to draw while being fairly informative, whereas 64×64 masks at the level 6 usually capture enough details, as illustrated in Fig. 4.

3.2. Text Feature Pyramid

The mask pyramid encodes the shape and the precision information. To generate an image, we also need semantic information. We introduce a text feature pyramid for this purpose. Each level of the text feature pyramid is a $2^l \times 2^l$ text feature map $\boldsymbol{Z}^l$ obtained by combining the masks $\boldsymbol{M}^l$

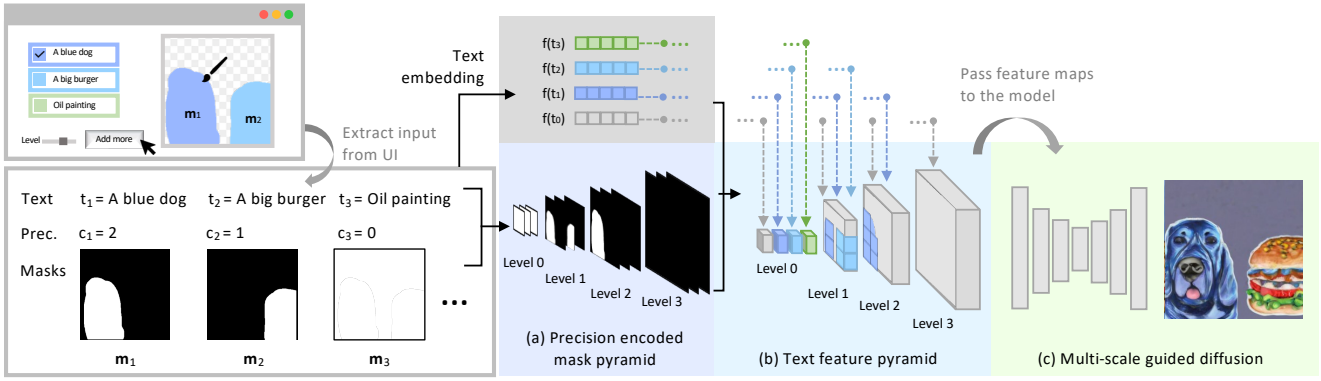


Figure 2. An overview of the proposed method. We provide an intuitive interface where users can easily define a layout using a semantic brush associated with a free-form text description and adjustable precision level. The masks, regional descriptions, and precision levels $\{(M_i, t_i, c_i)\}_{i=1}^{n}$ are jointly encoded into a text feature pyramid, and then translated into an image by a multi-scale guided diffusion model.

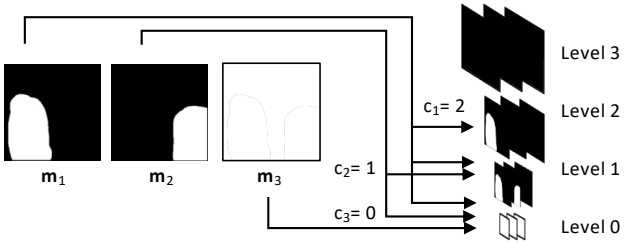
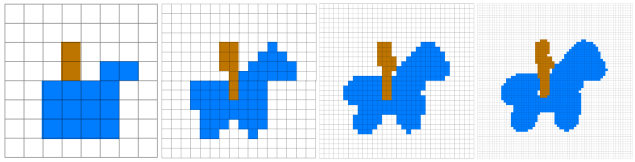


Figure 3. An example of the precision-encoded mask pyramid. The first level has two non-zero masks corresponding to m_1, m_2 as $c_1, c_2 \geq 1$; the second level only has m_1 as $c_1 = 2, c_2 < 2$; the third level does not have any non-zero masks as all the precision levels of all masks are lower than 3.



Level 3 (8×8) Level 4 (16×16) Level 5 (32×32) Level 6 (64×64)

Figure 4. Illustration of the masks at different levels of a pyramid.

and the embeddings of the text $\boldsymbol{t}$. At the 0-th level, the 1×1 masks contain no shape information, so we simply concatenate the embeddings of all words into a sequence. At the levels where $l > 0$, we spatially spread the embeddings of $\boldsymbol{t}$ over the corresponding masks to jointly represent the shape and semantic, as illustrated in Fig. 2.

Text feature maps. Here we describe in more detail how we construct the text feature map $\boldsymbol{Z}^l$ at level $l > 0$. For simplicity, we drop the superscript l and denote the masks at an arbitrary level as $\boldsymbol{M}$. Each element $M_{i,x,y}$ is a binary value $\in \{0, 1\}$ and we allow overlaps or blank spaces, *i.e.* $n \geq \sum_i^n M_{i,x,y} \geq 0$. Given the initial masks $\boldsymbol{M}$, we introduce normalized masks $\hat{\boldsymbol{M}}$, which are augmented from $\boldsymbol{M}$ by adding an extra mask M_0 to indicate the blank space and normalized by the number of shapes at each location:

$$\hat{M}_{i,x,y} = M_{i,x,y} / \sum_{i=0}^{n} M_{i,x,y}, \quad \text{where } M_{0,x,y} = \mathbb{1}_{\sum_{i=1}^{n} M_{i,x,y}=0}. \tag{2}$$

In the normalized masks $\hat{\boldsymbol{M}}$, each element $\hat{M}_{i,x,y}$ is a continuous value $\in [0, 1]$ and we have $\sum_{i=0}^{n} \hat{M}_{i,x,y} = 1$. We compute each element $\boldsymbol{Z}_{x,y}$ at location x, y as follows,

$$\boldsymbol{Z}_{x,y} = \sum_{i=0}^{n} \boldsymbol{f}(t_i) \cdot \hat{M}_{i,x,y}, \tag{3}$$

where $\boldsymbol{f}(t_i)$ is an embedding of the text t_i. We set t_0 to be a null token $\oslash$ to represent unspecified areas, *i.e.* the blank space indicated by M_0.

The text feature map representation has several advantages. First, it is of the same dimension regardless of the number of masks, and therefore compatible with most deep ConvNet architectures. Second, each element of a text feature map is a convex combination of n text embeddings in the learned embedding space. With a powerful language model, we can achieve a good generalization capability for unseen combinations of concepts. In a text feature map, any overlapping area contains an interpolation of multiple embeddings. Accordingly, users can get creative results derived from hybrid concepts by drawing overlapping shapes.

Data acquisition. During inference, the segmentation masks and text descriptions are both provided by users. During training, we can generate them automatically using text-based object detection [28] and segmentation [16]. We set the regions where no objects are detected as blank space and assign a null token $\oslash$ to these regions. We use CLIP [38] text model to encode the text descriptions. In addition, the 0-th level feature maps can be obtained directly from the image caption embeddings. More details regarding the training data generation step can be found in the supplementary material.

Relation to one-hot label maps. In S2I, a layout containing

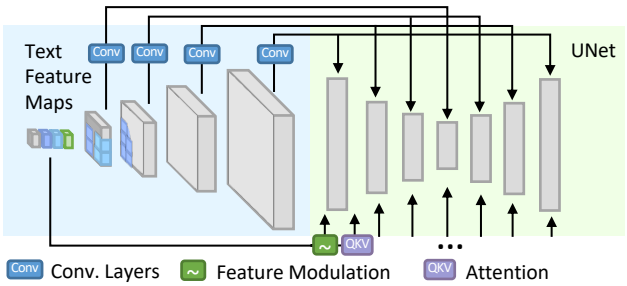


Figure 5. Architecture of the multi-scale guided diffusion model.

at most C classes can be encoded into a C-channel one-hot label map. It can be seen as a special case of text feature maps when the masks are dense and non-overlapping, and the embedding model $\boldsymbol{f}$ is a one-hot encoding function. Specifically, let $\boldsymbol{M}$ be the segmentation masks, $\boldsymbol{t}$ the class labels where $t_i = 1, 2, ..., C$, $f_c(t_i) = \mathbb{1}_{t_i=c}$. Since $\boldsymbol{M}$ are dense and non-overlapping, the normalized masks $\hat{M}_i$ will be the same as M_i for $i > 1$ and $\hat{M}_0 = 0$. Eqn. 3 then becomes

$$Z_{c,x,y} = \sum_{i=1}^{n} \mathbb{1}_{t_i=c} \cdot M_{i,x,y} = \sum_{i|t_i=c} M_{i,x,y}. \quad (4)$$

Therefore, the c-th channel of $\boldsymbol{Z}$ is a mask covering all pixels of class c, *i.e.* a one-hot label map. Through this reformulation, we can see that the one-hot representation has limited capability as $\boldsymbol{f}(t_i)$ is restricted to be binary. The text feature map representation uses a learned language model as $\boldsymbol{f}$ to encode the more informative open-domain layouts.

3.3. Multi-Scale Guided Diffusion

We use a diffusion model to generate images from a text feature pyramid. We use $\boldsymbol{\epsilon}$-prediction and the simplified training objective following [19]:

$$L = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,1),t\sim[1,T]} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}, t)\|_2^2 \right], \quad (5)$$

where T is the number of diffusion steps, $\boldsymbol{x}_t$ is a noisy version of the ground-truth image $\boldsymbol{x}$ at the t-th diffusion step, $\boldsymbol{z}$ is the conditional signal, which in our case is the text feature pyramid $\{\boldsymbol{Z}^l\}_{l=0}^{L}$, and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ represents the network parameterized by $\boldsymbol{\theta}$. We modify the UNet architecture [9] by adding the convolutional layers to combine each text feature map with the UNet feature maps of the corresponding resolution. The 0-th level text feature maps $\boldsymbol{Z}^0$ are passed through all blocks using cross-modal attention [44] and channel-wise feature modulation. Fig. 5 shows the overall architecture of the modified UNet.

Classifier-free guidance [20] has shown to be an effective technique to improve the performance of conditional diffusion models and is widely applied to diffusion models conditioned on text and class tags [9, 35, 43, 44]. To benefit from this technique, we introduce multi-scale spatial guidance for our feature map conditioned diffusion model. As described in Sec. 3.2, in text feature maps from level 1 to L, the unspecified regions are the embedding of a null token $\boldsymbol{f}(\oslash)$. We further apply dropout with probability 0.1 to $\boldsymbol{Z}^0$ by setting $\boldsymbol{Z}^0 = \boldsymbol{f}(\oslash)$. During inference, we estimate two diffusion scores, conditioning on the given text feature pyramid and an empty feature pyramid of repeating $\boldsymbol{f}(\oslash)$, and then perform sampling using their linear combination.

4. Experiments

In this section, we present the experimental results on image synthesis from any-level semantic layouts. We evaluate the proposed method on user-drawn coarse layouts and automatically simulated layout data. We also present results on T2I and S2I, and compare our method with state-of-the-art methods.

4.1. Implementation Details

We train the multi-scale guided diffusion model to generate 64×64 images from a layout of precision up to $\log_2 64$. To generate images at higher resolution, we experiment with two approaches: (1) by training another diffusion model to upsample $64 \rightarrow 256$ following [9]; (2) by generating the latent map at 64×64 resolution then decoding it into a 512×512 image following [43]. Both approaches are effective, demonstrating that the proposed method can generalize well to both pixel-space and latent-space diffusion strategies. We use smaller pixel-space diffusion models for ablation studies and report the main results based on a larger latent-space diffusion model. We encode the text descriptions using the pretrained CLIP ViT-L14 language model [38]. To avoid repeated forward passes, we feed the union of all regional descriptions $\cup_i t_i$ as a sentence into the language model. Then, we average the hidden states corresponding to each word in t_i to use as its embedding $\boldsymbol{f}(t_i)$.

Training details. We use the aesthetic subset of Laion2B-en dataset [46] for training. We generate full text feature pyramids for 5M randomly selected samples and construct only the 0-th level feature maps for the remaining samples. For pixel space diffusion, we train a base model of 300M parameters and a super-resolution model of 300M parameters; for latent-space diffusion, we train a base model of 900M parameters and use the pretrained deocder from [43]. We use batch size 2048 for 64×64 models and batch size 960 for the super-resolution model.

Evaluation benchmark. We use the COCO [32] validation set for evaluation in the T2I setting and the COCO-stuff [7] validation set for S2I. For evaluation in open-domain layout-to-image generation, we construct a new test set OpenLayout containing 260 user-drawn layouts of coarse shapes. The layouts are annotated by 10 users based on text prompts randomly sampled from PartiPrompts [64]. To analyze the effect of precision level control, we also evaluate using the pseudo layouts consisting of accurate shapes, which are extracted from the 5,000 images with caption annotations of COCO validation set (OpenLayout-COCO). Fig. 6 shows

Table 2. Quantitative results on the OpenLayout and OpenLayout-COCO datasets with different precision levels. SS Score: spatial similarity scores.

	OpenLayout		OpenLayout-COCO		
Prec.	SS Score ↑	CLIP Score ↑	SS Score ↑	CLIP Score ↑	FID ↓
0	.436	**.274**	.572	**.260**	**25.3**
3	.533	.268	.685	.256	26.0
4	.543	.266	.716	.256	27.0
5	.558	.267	.729	.256	28.4
6	**.558**	.261	**.736**	.255	30.6

some layouts samples from OpenLayout and Openlayout-COCO.

Figure 6. Top: Pseudo layouts from OpenLayout-COCO dataset. Bottom: real-world user-drawn layouts from OpenLayout dataset.

4.2. Open-Domain Layout-to-Image Generation

Here we discuss the results on the OpenLayout and OpenLayout-COCO datasets. For each sample, we run the model a varying precision level c from 0 to 6 to verify the effectiveness of precision control. Fig. 7 shows the results with different precision levels. For simplicity, we use the same precision level for all regions in an input layout, *i.e.* $\forall i, c_i = c$. We can see that as the precision level increases, the generated images follows the layout more closely. When $c = 0$, the image compositions are not related to the layouts. For $c = 3, 4$ the generated images roughly resemble the shape and location specified in the layout. For $c = 5, 6$, the generated object contours matches the layout more closely. At the lowest precision level, our method can handle very rough layouts, *e.g.*, bounding box layouts, as shown in Fig. 8, despite not being trained on bounding box data.

For quantitative evaluation, we compute the CLIP score using the original captions to measure global semantic alignment. To measure spatial alignment, we define a spatial similarity score (SS score), which is the cosine similarity between the text feature maps of the input layout and the layout reconstructed from the generated image. For OpenLayout-COCO, as the ground-truth images are available, we also compute the FID to measure the visual quality.

Table 2 reports the quantitative evaluation results on images generated with different precision levels. The second and fourth columns show that using a higher precision level generally leads to a higher spatial similarity, which further demonstrates the effectiveness of precision control. On OpenLayout-COCO, the spatial similarity consistently increases as the precision level becomes higher. Whereas, on OpenLayout, it stays the same at higher levels. This is due to the difference in the inherent layout precision exhibited by the two datasets; compared to OpenLayout-COCO, the layouts in OpenLayout are coarser. For a coarse layout, the generated images already match it well at low precision levels and the room for improvement is limited.

From the last column of Table 2, we can see that the results with a lower precision level have a smaller FID, which indicates a similar distribution with respect to the ground-truth images. This is likely because a lower precision level enforces smaller constraints on the generation process, and therefore the generated images can better capture the real image distribution. Similarly, a lower precision level also yields a slightly better CLIP score due to the smaller spatial constraints, as shown in the third and fifth columns of Table 2.

Table 3. Quantitative comparison with T2I methods on COCO.

Method	FID ↓	zero-shot FID↓
SSA-GAN [31]	19.37	-
VQ-Diffusion [15]	13.86	-
DF-GAN [53]	19.32	-
GLIDE [35]	-	12.89
SD [1, 43]	-	9.89
DALLE2 [39]	-	10.87
Ours	**8.55**	**9.47**

4.3. Text-to-Image Generation

As mentioned in Sec. 3, our method can be applied for text-to-image generation by using texts as 0-th layouts. The first four columns of Fig. 9 show the images generated from text prompts by our method and state-of-the-art methods. We can see that the proposed method can generate visually peasant images with reasonable layouts from only text input. Table 3 reports the FID evaluated on the COCO validation set. The classifier-free guidance scale is set to 3. Following [35], we sample 30K images using randomly selected text prompts and compute FID against the entire validation set. It can be seen that the proposed method compares favorably against state-of-the-art text-to-image generation models.

By combining with a named entity recognition (NER) model, we can apply the proposed method to layout controllable text-to-image generation. More specifically, given an input sentence, we parse the noun phrases using NER to generate regional text descriptions, and users can arbitrarily draw the shapes for those noun phrases. The last two columns of Fig. 9 show the text-to-image generation results

Figure 7. Results with different precision levels. For each input layout, we sample the images starting from same noise, so the images at different precision levels can have similar styles.

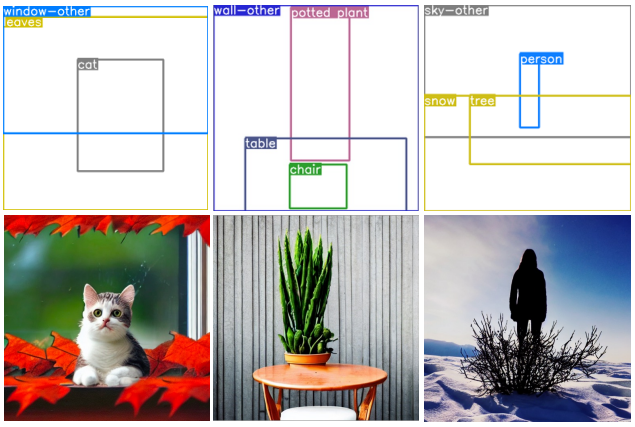


Figure 8. Results with bounding box layout. Precision level = 3.

with layout control. We can see that the generated images match the text well and follow the provided layouts.

4.4. Segmentation-to-Image Generation

Our model can also generate images from a dense segmentation map of closed-set labels, *i.e.* following the original S2I setting. For this application, we treat the class labels as text descriptions and use the highest precision level $c = 5$ for all masks. Fig. 10 shows example images generated by our method and SPADE [36]. The images generated by our model are of significantly better visual quality. Table. 4 reports the quantitative comparison results on the COCO-stuff validation set with the state-of-the-art S2I methods [33, 36, 51]. Since the highest resolution of the input layout to our model is 64, we evaluate the S2I results at 64×64 resolution. From the third and fifth columns of Table. 4 we can see that despite not being designed or trained for this task, the proposed method can achieve lower FID and comparable mIOU. After being fine-tuned on the COCO-stuff training set, our model outperforms previous approaches in terms of both FID and mIOU.

Table 4. Quantitative comparison with S2I methods on COCO-stuff.

Method	FID ↓	zero-shot FID ↓	mIOU↑	zero-shot mIOU↑
SPADE [36]	50.91	-	17.49	-
SAFM [33]	62.28	-	12.28	-
CLADE [51]	55.30	-	17.21	-
Ours	**17.20**	20.17	**23.01**	17.15

4.5. Ablation Studies

We compare the proposed any-level method with two fixed-level baseline models: a text-to-image generation model and a fixed-level layout-to-image generation model. The baseline models are of the same architecture as the any-level models. When training the baseline models, we only use layouts of fixed levels, *i.e.* the 0-th level for the text-to-image baseline and the 4-th level for the fixed-level segmentation-to-image baseline. Table 5 compares the evaluation results of the any-level model and fixed-level baselines at the corresponding levels. The any-level model achieves better results than the baseline models, which further demonstrates the advantage of the unified framework.

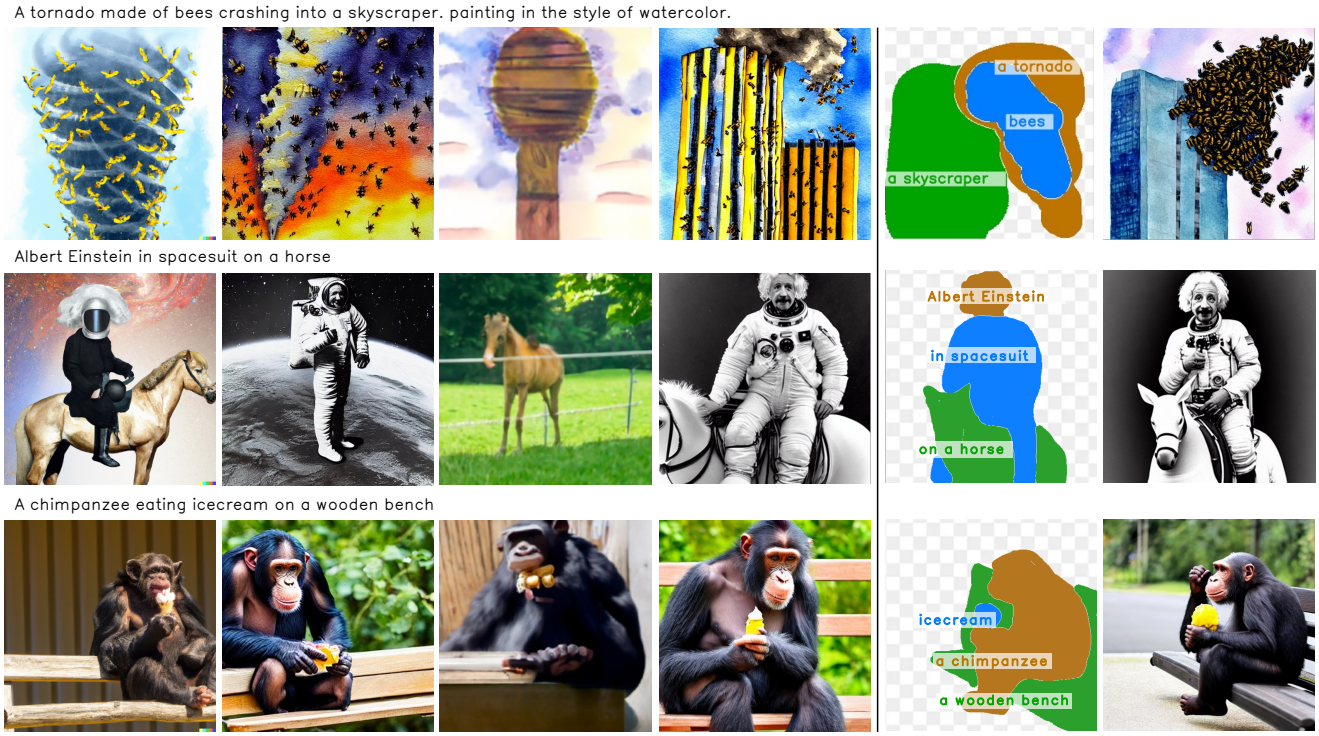


Figure 9. Text-to-image generation results. Ours: our results by using text as a 0-th level layout. Ours*: our results with layout control.

Figure 10. Visual comparison with SPADE [36] for segmentation-to-image generation.

Table 5. Ablation study results on OpenLayout-COCO.

Prec. Level	Model	FID ↓	CLIP Score ↑	SS Score ↑
0	Text-to-Image Baseline	32.83	.2473	.5613
0	Any-Level Model	**32.09**	**.2496**	**.5640**
4	Single-Level Baseline	36.03	.2381	.5729
4	Any-Level Model	**33.70**	**.2475**	**.6978**

5. Conclusion

This paper presents a new conditional image synthesis framework to generate images from any-level open-domain semantic layouts. The input level ranges from pure text to a 2D semantic canvas with precise shapes. Several novel techniques are introduced, including a pipeline for collecting training data; the representations to jointly encode precision level, semantics, and geometry information; and a multi-scale guided diffusion model to synthesize images. A test dataset containing user-drawn layouts is collected to evaluate the proposed method. Experimental results demonstrate the advantage of the unified framework. The proposed method can generate high-quality images following the layout at specified precision levels, and compares favorably against the state-of-the-art methods on public benchmarks.

References

[1] Stable diffusion. https://stability.ai/blog/stable-diffusion-public-release. Accessed: 2022. 2, 3, 6

[2] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984. 2

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

[4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 3

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning and Representation*, 2018. 2

[6] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*, pages 671–679. Elsevier, 1987. 2

[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018. 5

[8] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *International Conference on Computer Vision*, pages 1511–1520, 2017. 3

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Conference on Neural Information Processing Systems*, 34:8780–8794, 2021. 1, 2, 5

[10] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Conference on Neural Information Processing Systems*, 34:19822–19835, 2021. 3

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 2

[12] Stanislav Frolov, Prateek Bansal, Jörn Hees, and Andreas Dengel. Dt2i: Dense text-to-image generation from region descriptions. page 395–406, Berlin, Heidelberg, 2022. Springer-Verlag. 3

[13] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *arXiv preprint arXiv:2203.13131*, 2022. 2, 3

[14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *Conference on Neural Information Processing Systems*, 2014. 2

[15] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 6

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision*, pages 2961–2969, 2017. 4

[17] Sen He, Wentong Liao, Michael Ying Yang, Yongxin Yang, Yi-Zhe Song, Bodo Rosenhahn, and Tao Xiang. Context-aware layout to image generation with enhanced object appearance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 15049–15058, 2021. 3

[18] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Generating multiple objects at spatially distinct locations. In *International Conference on Learning and Representation*, 2018. 3

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Conference on Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 5

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 5

[21] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7986–7994, 2018. 3

[22] Xun Huang, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Multimodal conditional image synthesis with product-of-experts gans. In *European Conference on Computer Vision*, pages 91–109. Springer, 2022. 3

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 2, 3

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning and Representation*, 2014. 2

[27] Bowen Li, Xiaojuan Qi, Philip HS Torr, and Thomas Lukasiewicz. Image-to-image translation with text guidance. *arXiv preprint arXiv:2002.05235*, 2020. 3

[28] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 4

[29] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12174–12182, 2019. 3

[30] Yandong Li, Yu Cheng, Zhe Gan, Licheng Yu, Liqiang Wang, and Jingjing Liu. Bachgan: High-resolution image synthesis from salient object layout. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8365–8374, 2020. 3

[31] Wentong Liao, Kai Hu, Michael Ying Yang, and Bodo Rosenhahn. Text to image generation with semantic-spatial aware gan. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 18187–18196, 2022. 6

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 5

[33] Zhengyao Lv, Xiaoming Li, Zhenxing Niu, Bing Cao, and Wangmeng Zuo. Semantic-shape adaptive feature modulation for semantic image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11214–11223, 2022. 7

[34] Ke Ma, Bo Zhao, and Leonid Sigal. Attribute-guided image generation from layout. In *Brit. Mach. Vis. Conf.*, 2021. 3

[35] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2, 3, 5, 6

[36] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3, 7, 8

[37] Dario Pavllo, Aurelien Lucchi, and Thomas Hofmann. Controlling style and semantics in weakly-supervised image generation. In *European Conference on Computer Vision*, pages 482–499. Springer, 2020. 3

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 4, 5

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2, 3, 6

[40] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3

[41] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016. 2, 3

[42] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. *Conference on Neural Information Processing Systems*, 29, 2016. 3

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 5, 6

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2, 3, 5

[45] Edgar Schönfeld, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *International Conference on Learning and Representation*, 2021. 3

[46] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 5

[47] Assaf Shocher, Yossi Gandelsman, Inbar Mosseri, Michal Yarom, Michal Irani, William T Freeman, and Tali Dekel. Semantic pyramid for image generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning and Representation*, 2020. 1, 2

[50] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *he AAAI Conference on Artificial Intelligence*, 2021. 2

[51] Zhentao Tan, Dongdong Chen, Qi Chu, Menglei Chai, Jing Liao, Mingming He, Lu Yuan, Gang Hua, and Nenghai Yu. Efficient semantic image synthesis via class-adaptive normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 7

[52] Zhentao Tan, Qi Chu, Menglei Chai, Dongdong Chen, Jing Liao, Qiankun Liu, Bin Liu, Gang Hua, and Nenghai Yu. Semantic probability distribution modeling for diverse semantic image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[53] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 16515–16525, 2022. 6

[54] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Conference on Neural Information Processing Systems*, 33:19667–19679, 2020. 2

[55] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Conference on Neural Information Processing Systems*, 29, 2016. 2

[56] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Inter-*

national Conference on Machine Learning, pages 1747–1756. PMLR, 2016. 2

[57] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Conference on Neural Information Processing Systems*, 30, 2017. 2

[58] Bo Wang, Tao Wu, Minfeng Zhu, and Peng Du. Interactive image synthesis with panoptic layout generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7783–7792, 2022. 3

[59] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 3

[60] Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models. *arXiv preprint arXiv:2207.00050*, 2022. 3

[61] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *European Conference on Computer Vision*, pages 720–736. Springer, 2022. 3

[62] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[63] Yuan Xue, Yuan-Chen Guo, Han Zhang, Tao Xu, Song-Hai Zhang, and Xiaolei Huang. Deep image synthesis from intuitive user input: A review and perspectives. *Computational Visual Media*, 8:3–31, 2022. 2

[64] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 3, 5

[65] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *International Conference on Computer Vision*, pages 5907–5915, 2017. 3

[66] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8584–8593, 2019. 3

[67] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2