Compositional Text-to-Image Generation with Dense Blob Representations

Weili Nie [1] **Sifei Liu** [1] **Morteza Mardani** [1] **Chao Liu** [1] **Benjamin Eckart** [1] **Arash Vahdat** [1]

Abstract

Existing text-to-image models struggle to follow complex text prompts, raising the need for extra grounding inputs for better controllability. In this work, we propose to decompose a scene into visual primitives – denoted as dense blob representations – that contain fine-grained details of the scene while being modular, human-interpretable, and easy-to-construct. Based on blob representations, we develop a blob-grounded text-to-image diffusion model, termed BlobGEN, for compositional generation. Particularly, we introduce a new masked cross-attention module to disentangle the fusion between blob representations and visual features. To leverage the compositionality of large language models (LLMs), we introduce a new in-context learning approach to generate blob representations from text prompts. Our extensive experiments show that BlobGEN achieves superior zero-shot generation quality and better layout-guided controllability on MS-COCO. When augmented by LLMs, our method exhibits superior numerical and spatial correctness on compositional image generation benchmarks. Project page: https://blobgen-2d.github.io.

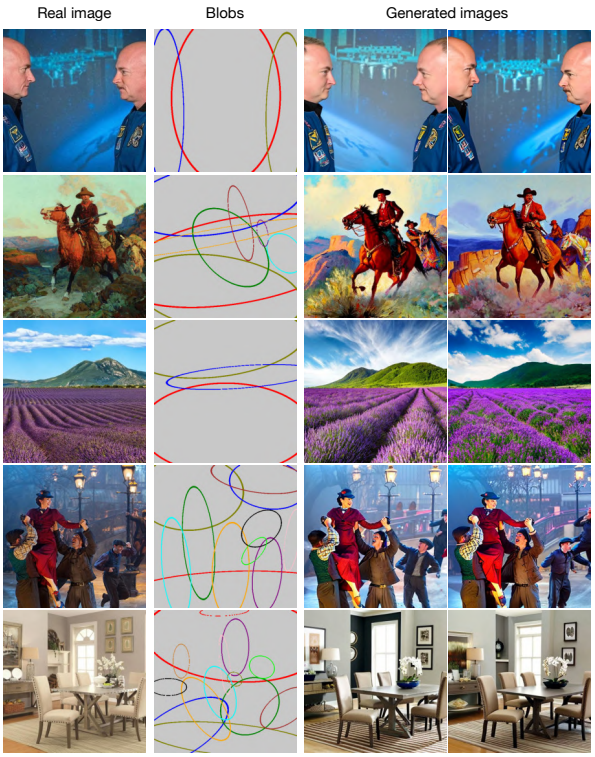


Figure 1. Generated images from blob representations can reconstruct fine-grained details of real images. Each row shows the real image (Left), blobs (Middle), and two randomly generated samples (Right). We do not show blob descriptions for simplicity.

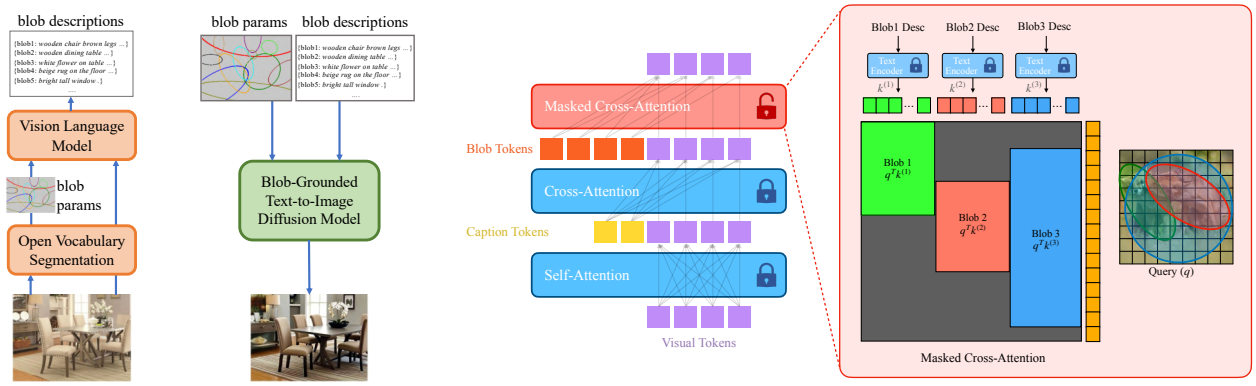
1. Introduction

Recent advances in text-to-image models enable us to generate realistic high-quality images (Ramesh et al., 2022; Saharia et al., 2022; Podell et al., 2023; Balaji et al., 2022). This rapid rise in quality has been driven by new training and sampling strategies (Ho et al., 2020; Song et al., 2020), new network architectures (Dhariwal & Nichol, 2021; Rombach et al., 2022), and internet-scale image-text paired data (Schuhmann et al., 2022). Despite the progress, current large-scale text-to-image models struggle to follow complex prompts, where they tend to misunderstand the context and ignore keywords (Betker et al., 2023; Huang et al., 2023a). Thus, fine-grained controllability is an open problem.

To cope with these challenges, recent works have attempted to condition text-to-image models on visual layouts. Since a text prompt can be vague in describing visual concepts (*i.e.*, the precise location of an object), image generation models may face difficulty striking a balance between expressing the given information and hallucinating missing information. Additional grounding inputs can guide the generation process for better controllability. These layouts can be represented by bounding boxes (Li et al., 2023), semantic maps (Zhang et al., 2023), depths (Rombach et al., 2022), and other modalities (Li et al., 2023; Zhang et al., 2023). Among them, semantic and depth maps provide fine-grained information but are not easy for users to construct and manipulate. In contrast, bounding boxes are user-friendly but

[1]NVIDIA Corporation. Correspondence to: Weili Nie <wnie@nvidia.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



(a) Blob representations (b) Masked cross-attention

Figure 2. (a) We extract blob representations (parameters and descriptions) using existing tools to guide the text-to-image diffusion model. (b) Our model leverages a novel masked cross-attention module that allows visual features to attend to only corresponding blobs.

contain more coarse-grained information than semantic and depth maps (Wang et al., 2016; Li et al., 2023).

In this work, we introduce a new type of visual layout, termed *dense blob representation*, to serve as grounding inputs to guide text-to-image generation. The blob representations correspond to visual primitives (*i.e.*, objects in a scene) and can be automatically extracted from a scene. Specifically, a blob representation has two components: 1) the blob parameter, which formulates a tilted ellipse to specify the object's position, size and orientation; and 2) the blob description, which is a rich text sentence that describes the object's appearance, style, and visual attributes. With this definition, our proposed blob representation can largely preserve the fine-grained layout and semantic information of a scene (see Figure 1). Furthermore, since blob parameters and descriptions are both represented with structured texts, they can be easily constructed and manipulated by users.

We then develop a blob-grounded text-to-image diffusion model, termed BlobGEN, that is built upon existing diffusion models, with blob representations as grounding input. To disentangle the fusion between blob representations and visual features, we devise a masked cross-attention module that relates each blob to the corresponding visual feature solely in its local region. Furthermore, inspired by Feng et al. (2023a); Lian et al. (2023) who prompt LLMs to plan box layouts, we design a new in-context learning approach for LLMs to generate dense blob representations from text prompts. By augmenting our model with LLMs, we can leverage the visual understanding and compositional reasoning capabilities of LLMs to solve complex compositional image generation tasks. Our model paves the way for a modular framework where images can be easily generated or manipulated by users and LLMs.

Our extensive experiments indicate that BlobGEN achieves superior zero-shot generation quality on MS-COCO (Lin et al., 2014). For instance, it improves the zero-shot FID of base model from 10.40 to 8.61, and offers much better layout-guided controllability than GLIGEN (Li et al., 2023) as demonstrated by region-level CLIP (Radford et al., 2021) scores. By solely modifying a single blob representation while holding other blobs static, BlobGEN exhibits a strong local editing and object repositioning capability. With LLM augmentation, our method excels in compositional generation tasks. For instance, our method outperforms LayoutGPT (Feng et al., 2023a) by 5.7% and 1.4% for spatial and numerical accuracy on NSR-1K (Feng et al., 2023a).

Overall, our main contributions are summarized as follows:

- We propose to decompose a scene into dense blob representations, each of which represents fine-grained details of a visual primitive (*i.e.*, an object) in the scene.
- We propose BlobGEN, a blob-grounded modular text-to-image model with a new masked cross-attention module that takes blob representations as grounding input.
- We augment our model with LLMs for compositional generation, by designing a new in-context learning approach for LLMs to infer blob representations from text prompts.
- We show our method achieves better zero-shot generation performance on MS-COCO, and has better numerical and spatial correctness in compositional benchmarks.

2. Method

We first introduce our image decomposition into blob representations, and then describe the new generative framework that conditions on blob representations to generate images. Finally, we present the customized in-context learning procedure that prompts LLMs to generate blobs.

2.1. Image Decomposition into Blob Representations

Given an image, we aim to extract visual primitives or object-level representations that satisfy two properties: 1) They contain fine-grained details of the scene such that

the original image can be *semantically* reconstructed in the maximum degree from them, and 2) they are modular, human-interpretable and easy to construct or manipulate, which means users can create and edit an image efficiently. To this end, we propose a new type of visual layouts, termed *dense blob representations*, each of which describes a single object in a scene. A blob representation consists of two components: *blob parameter* and *blob description*.

Formally, a blob parameter specifies the size, location, and orientation of the blob using a vector of five variables $[c_x, c_y, a, b, \theta]$, where (c_x, c_y) is the center point of the ellipse, a and b are the radii of its semi-major and semi-minor axes, and $\theta \in (-\pi, \pi]$ is the orientation angle of the ellipse. Intuitively, similar to the functionality of bounding boxes (Li et al., 2023; Yang et al., 2023), the blob parameter can represent the location and size of an object. On the other hand, due to the existence of the orientation angle θ, the visual layout depicted by a blob parameter is more fine-grained than a bounding box: 1) it can additionally describe the orientation or pose of an object, and 2) it can more precisely describe the shape and size of an object, particularly those with an elongated shape and a large inclined angle.

A blob description is a text sentence that describes the visual appearance of an object, complementing the spatial layout information depicted by the blob parameter. In this work, we use a region-level synthetic caption extracted by a pre-trained image captioner as the blob description. As shown in Figure 2a, it not only provides the category name but also captures the detailed visual features of an object, including its appearance (*e.g.*, color, texture, and material, etc.) and the spatial relationship of sub-parts within the object region (*e.g.*, "a wooden chair with brown legs and soft seat").

Since our blob representations retain the fine-grained visual layouts and other detailed visual features of the original image, it can faithfully recover the image with a diffusion model (see Figure 1). Note that blob representations can also capture irregular, large objects and background (see Figure 9). Moreover, both blob parameters and descriptions are in the form of simple text inputs, and thus they can be easily constructed and manipulated by human users and even generated by LLMs as we will show next.

2.2. Blob-grounded Text-to-Image Generation

Existing text-to-image diffusion models often consist of convolutional and self-attention layers that operate on image features directly, and cross-attention layers that inject text conditioning into the network (see Figure 2b). We build BlobGEN upon the pre-trained text-to-image Stable Diffusion model, where we introduce new cross-attention layers to incorporate blob grounding into the diffusion model. To retain the prior knowledge of pre-trained models for synthesizing high-quality images, we freeze their weights and only train the newly added layers. In the following, we highlight the key design choices for blob-grounded generation.

Blob Embedding. Denote the blob parameter as $\boldsymbol{\tau} := [c_x, c_y, a, b, \theta]$ and blob description as $\boldsymbol{s} := [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_L]$, where L is the text sentence length. For blob parameter $\boldsymbol{\tau}$, we first encode its orientation angle θ to the sine and cosine representation $(\sin\theta, \cos\theta)$, and then obtain the blob parameter embedding $\boldsymbol{e}_\tau = \text{Fourier}(\tilde{\boldsymbol{\tau}}) \in \mathbb{R}^{d_\tau}$, where $\tilde{\boldsymbol{\tau}} := [c_x, c_y, a, b, \sin\theta, \cos\theta]$ and $\text{Fourier}(\cdot)$ denotes the Fourier feature encoding (Tancik et al., 2020). For blob description, we use the CLIP text encoder f to obtain the sentence embedding $\boldsymbol{e}_s = f(\boldsymbol{s}) := [\boldsymbol{e}_{s_1}, \cdots, \boldsymbol{e}_{s_L}] \in \mathbb{R}^{L \times d_s}$. Before we pass the blob sentence embedding to the network, we first concatenate the two embeddings $\boldsymbol{e}_\tau$ and $\boldsymbol{e}_s$. Thus, the final blob embedding is given by

$$\boldsymbol{e}_b = \text{MLP}([\tilde{\boldsymbol{e}}_{s_1}, \cdots, \tilde{\boldsymbol{e}}_{s_L}]) \in \mathbb{R}^{L \times d_b}$$

where $\tilde{\boldsymbol{e}}_{s_l} := [\boldsymbol{e}_{s_l}; \boldsymbol{e}_\tau] \in \mathbb{R}^{d_s + d_\tau}$ for all $l \in \{1, \cdots, L\}$ with $[\cdot;\cdot]$ denoting a concatenation along the feature dimension, and $\text{MLP}(\cdot)$ represents an MLP layer.

Masked Cross-Attention. Given N blob embeddings, denoted as $\{\boldsymbol{e}_b^{(n)}\}_{n=1}^N$, we represent $\boldsymbol{g} \in \mathbb{R}^{hw \times d_g}$ as the visual features of an image, where h and w represent the spatial size of the feature maps. If the query, key and value are denoted by $\boldsymbol{q} := \boldsymbol{g}\boldsymbol{W}_q \in \mathbb{R}^{hw \times d_g}$, $\boldsymbol{k}^{(n)} := \boldsymbol{e}_b^{(n)}\boldsymbol{W}_k^{(n)} \in \mathbb{R}^{L \times d_g}$, and $\boldsymbol{v}^{(n)} := \boldsymbol{e}_b^{(n)}\boldsymbol{W}_v^{(n)} \in \mathbb{R}^{L \times d_g}$, respectively, a standard cross-attention between $\boldsymbol{g}$ and $\{\boldsymbol{e}_b^{(n)}\}_{n=1}^N$ is

$$\text{CA}(\boldsymbol{g}, \{\boldsymbol{e}_b^{(i)}\}) = \sigma\left(\frac{\boldsymbol{q}[\boldsymbol{k}^{(1)}; \cdots; \boldsymbol{k}^{(N)}]^T}{\sqrt{d_g}}\right)[\boldsymbol{v}^{(1)}; \cdots; \boldsymbol{v}^{(N)}]$$

where $[\cdot;\cdot]$ is a concatenation along the sequence dimension and $\sigma(\cdot)$ is the softmax function. We can see that, in the standard cross-attention, every blob embedding attends to every feature "pixel" (in the $h \times w$ plane) of the feature maps. This is undesirable since blob embedding only conveys information about its corresponding local region, and its interaction with other regions may confuse the model, leading to more text leakage and entanglement in generation.

To solve this issue, we propose to mask the feature maps $\boldsymbol{g}$ such that each blob embedding only attends to its local region, as shown in Figure 2b. Denote the attention mask for the i-th blob as $\boldsymbol{m}^{(i)} \in \mathbb{R}^{hw}$. It is obtained by downsampling the i-th blob's binary ellipse mask where a pixel value is 1 if it is within the blob ellipse, and 0 otherwise. Accordingly, we define the *masked cross-attention* as

$$\text{CA}_m(\boldsymbol{g}, \{\boldsymbol{e}_b^{(i)}, \boldsymbol{m}^{(i)}\}) = \sigma\left(\frac{[\boldsymbol{a}^{(1)}; \cdots; \boldsymbol{a}^{(N)}]}{\sqrt{d_g}}\right)[\boldsymbol{v}^{(1)}; \cdots; \boldsymbol{v}^{(N)}]$$

where the i^{th} attention weight for the j^{th} location is:

$$\boldsymbol{a}_j^{(i)} = \begin{cases} \boldsymbol{q}_j \boldsymbol{k}^{(i)T} & \text{if } \boldsymbol{m}_j^{(i)} = 1 \\ -\infty & \text{otherwise} \end{cases} \quad \text{for } j \in \{1, 2, \ldots, hw\}.$$

With this masking design, blob representations and local visual features are well aligned in an explicit manner. Therefore, the blob grounding process can be more modular and independent across different object regions, and the model can be more disentangled in generation.

Other Design Choices. Similar to (Alayrac et al., 2022; Li et al., 2023), we also add the masked cross-attention module in a gated way, where a learnable scalar controls the information flow from the cross-attention branch for more stable training. We optionally add the gated self-attention module proposed by (Li et al., 2023), which shows a slight improvement in generation quality. We do not make any changes to self-attention and convolutional layers, allowing information to propagate across different image regions for overall long-range correlations. For the image-level global captions, we find synthetic ones work better than original real captions (Betker et al., 2023), so we only use synthetic global captions to train our model (see Appendix A.1). Finally, we use the original denoising score matching loss (Ho et al., 2020) to train only the new parameters.

2.3. LLMs for Blob Generation

Here, we aim to show that our blob representations can be generated by LLMs. Specifically, we design two separate in-context learning processes: one for generating blob parameters and another for generating blob descriptions.

Blob Parameter Generation. Inspired by Feng et al. (2023a), we adopt the CSS format to represent blob parameters such that LLMs better understand their spatial meaning. Each generated layout in an in-context example starts with the category name, followed by a declaration section in the CSS style, which is "`object {major-radius: ?px; minor-radius: ?px; cx: ?px; cy: ?px; angle: ?}`". The first four values are measured in pixel length, whereas the last value for angle is expressed in degree and normalized to be within $[0, 180]$. All values are rounded to integers. Next, we follow the procedure of Feng et al. (2023a) to select top-k similar demonstration examples[1]. The final prompt for LLMs consists of a system prompt that instructs the blob parameter generation, k demonstration examples, and the test prompt (usually a global caption). See Appendix A.3.1 for details.

Blob Description Generation. Blob descriptions are less structured as they are essentially a list of text sentences. Thus, we do not use the CSS format to generate them but we still use the category name as a separator between blobs for the ease of LLM generation. Thus, each generated blob description in an in-context example is formatted as "`object {text sentence}`". We utilize the same method to select top-k demonstration examples and construct the final prompt, which includes a system prompt that instructs the blob description generation, k demonstration examples, and the test prompt. See Appendix A.3.2 for details.

[1] In fact, retrieval from a large blob dataset to obtain in-context demonstration examples is *not necessary* for our method. See Appendix B.5 for more details.

3. Related Work

Text-to-Image Generation. Large text-to-image generative models have attracted much attention in the past few years, due to their unprecedented photorealism in generation. Among them, many methods are based on diffusion models (Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022; Balaji et al., 2022) or auto-regressive models (Ramesh et al., 2021; Yu et al., 2022; Chang et al., 2023). Instead of purely improving visual quality, recent models aim at improving their prompt following capabilities (Podell et al., 2023; Betker et al., 2023), where training on synthetic image captions becomes a promising direction (Betker et al., 2023; Chen et al., 2023a; Wu et al., 2023b). Different from them, we use the synthetic caption as an object-level text description for each blob. By conditioning existing text-to-image models on blob representations, our generation can follow more fine-grained, object-level user instructions while maintaining high visual quality.

Compositional Image Generation. Early works have proposed to learn concept distributions defined by energy functions, which can be explicitly combined (Du et al., 2020; Nie et al., 2021; Liu et al., 2022). Blob parameters have also been used for spatially disentangled generation with GANs (Epstein et al., 2022). Based on text-to-image models, recent methods have focused on learning special text tokens to represent concepts and injecting them into the text prompt for concept compositions (Ruiz et al., 2023; Kumari et al., 2023; Xiao et al., 2023). Other methods have explored guiding internal representations through cross-attention maps to steer sampling (Feng et al., 2022; Chefer et al., 2023; Epstein et al., 2023; Chen et al., 2023b; Phung et al., 2023). Another line of research aims at conditioning text-to-image models on extra visual layouts for better controllability (Yang et al., 2023; Li et al., 2023; Zheng et al., 2023; Huang et al., 2023b; Feng et al., 2023b). However, none of them uses blob representations as grounding inputs for compositional generation.

LLM-augmented Image Generation. With their generalization abilities, LLMs have also been used in text-to-image generation. Wu et al. (2023a) introduce a prompt manager that links LLMs with various text-to-image models to exe-

Table 1. Evaluation of zero-shot generation quality and layout-guided controllability on MS-COCO validation set. †These models need to use an extra database for retrieval-augmented generation.

Method	#Parameters	Zero-shot Generation	Controllability		
		FID ↓	mIOU ↑	rCLIP$_t$ ↑	rCLIP$_i$ ↑
CogView (Ding et al., 2021)	4B	27.10	-	-	-
KNN-Diffusion (Sheynin et al., 2022)	470M†	16.66	-	-	-
GLIDE (Sheynin et al., 2022)	6B	12.24	-	-	-
Make-a-Scene (Gafni et al., 2022)	4B	11.84	-	-	-
DALL-E 2 (Ramesh et al., 2022)	5.5B	10.39	-	-	-
LAFITE2 (Zhou et al., 2022)	1.45B†	8.42	-	-	-
Muse (Chang et al., 2023)	3B	7.88	-	-	-
Imagen (Saharia et al., 2022)	8B	7.27	-	-	-
Parti (Yu et al., 2022)	20B	7.23	-	-	-
Re-Imagen (Chen et al., 2022)	8B†	6.88	-	-	-
w/ SD decoder (Rombach et al., 2022)					
Base model (SD-1.4, Rombach et al. 2022)	1B	11.14	0.1338	0.2443	0.7558
GLIGEN (Li et al., 2023)	1.2B	11.63	0.4154	0.2688	0.7941
GLIGEN w/ synthetic captions	1.2B	10.80	0.4143	0.2724	0.7964
Ours	1.4B	8.94	0.5000	**0.2906**	0.8241
w/ consistency decoder (Betker et al., 2023)					
Base model (SD-1.4, Rombach et al. 2022)	1.5B	10.40	0.1316	0.2381	0.7520
GLIGEN (Li et al., 2023)	1.7B	11.15	0.4238	0.2591	0.7976
GLIGEN w/ synthetic captions	1.7B	10.54	0.4244	0.2609	0.7994
Ours	1.9B	8.61	**0.5103**	0.2794	**0.8288**

cute complex image synthesis tasks. Several works propose to fuse LLMs with text-to-image models for various multi-modal generation tasks by mapping between their embedding spaces (Koh et al., 2023; Sun et al., 2023). More similarly, other works use LLMs to infer visual layouts from text prompts as grounding inputs for text-to-image models (Feng et al., 2023a; Lian et al., 2023; Feng et al., 2023b). They demonstrate that LLMs can generate bounding boxes and well-structured attributes with carefully designed prompts, but it remains unclear whether blob representations can be generated by LLMs and how robust our blob-grounded generative model is to LLM-planned layouts.

4. Experiments

We first evaluate the generation performance of our blob-grounded text-to-image generative model; following that, we evaluate its compositional reasoning performance when augmented with in-context LLMs for blob generation.

4.1. Blob-grounded Text-to-Image Generation

Here we compare BlobGEN with previous methods on zero-shot image generation, and perform ablation studies to highlight the impact of each design choice in our method.

4.1.1. Experiment Setup

Data Preparation. We use a dataset of random 1M image-text pairs from the Common Crawl web index (filtered with the CLIP score) and resize all images to a resolution of 512×512. To extract blob representations for each image, we first apply ODISE (Xu et al., 2023) to get instance segmentation maps, followed by an ellipse fitting optimization to determine blob parameters for each map, aiming to maximize the Intersection Over Union (IOU) between the blob ellipse and segmentation mask. With segmentation maps, we crop out local regions for all objects in an image and use LLaVA-1.5 (Liu et al., 2023a) to caption each blob. On average, each image contains 12 blobs.

Training Details. Our model adopts the LDM framework (Rombach et al., 2022) and is built upon the SD-1.4 checkpoint. An image of resolution 512×512 is mapped to a latent space of $64 \times 64 \times 4$ by an image encoder. By default, our model is trained on 1M samples for 400K steps using a batch size of 512, requiring 9 days on 64 NVIDIA A100 GPUs. We use the AdamW optimizer (Loshchilov & Hutter, 2018) and the learning rate of 5×10^{-5} with a linear warm-up for the first 10K steps. We set the maximum number of blobs per image to 15. To encourage the model to rely more strongly on the blob representations, we randomly drop the global caption with 50% probability.

Evaluation Metrics. We use FID (Heusel et al., 2017) to compare the visual quality of generated images from different models. Unless stated otherwise, all FIDs are computed with 30K generated and real images. To the measure the controllability, i.e., how well the generation follows the layout guidance, we propose three metrics: mIOU, rCLIP$_i$ and rCLIP$_t$. The mIOU is defined as the mean IOU between the segmentation maps by applying LangSAM[2] to the generated image and the region ellipse masks depicted by input blob parameters. The rCLIP$_i$ is defined as the region-level CLIP score between the generated image and the ground-truth

[2]https://github.com/luca-medeiros/lang-segment-anything.

Figure 3. Zero-shot layout-grounded generation results of GLIGEN and our method on MS-COCO validation set. In each row, we visualize the reference real image (Left), bounding boxes and GLIGEN generated image (Middle), blobs and our generated image (Right). All images are in resolution of 512×512.

real image (paired with the input global caption). Here, "region-level" means that we pass a cropped image specified by the blob parameter to the CLIP image encoder for an image embedding. Similarly, the rCLIP$_t$ is defined as the region-level CLIP score between the generated image and the corresponding blob descriptions.

4.1.2. Zero-shot Generation on MS-COCO

Quantitative Results. In Table 1, we compare our method with the state-of-the-art models in terms of zero-shot generation quality and controllability on the MS-COCO validation set. For our method and two closely related baselines (base model and GLIGEN), we report the results with different image decoders (*i.e.*, SD decoder and consistency decoder). First, we observe that our method achieves lower FID than both GLIGEN, which uses bounding boxes as grounding input, and the base model SD-1.4. The consistency decoder always improves FID but it also increases the overall model size. When compared with other text-to-image models that have a much larger model size, our FID also remains competitive. It implies that adding blob representations largely improves the image synthesis quality.

Furthermore, our method outperforms both GLIGEN and base model by a large margin regarding all three metrics for controllability: mIOU, rCLIP$_i$ and rCLIP$_t$. This demonstrates that our generated images also have better region-level consistency with the grounding inputs, besides better image-level visual quality. Also, the consistency decoder in general achieves better mIOU and rCLIP$_i$ than the SD decoder, but it deteriorates the rCLIP$_t$ score. We hypothesize that this discrepancy arises from a misalignment between the consistency decoder and the CLIP text encoder. GLIGEN trained with synthetic captions can also slightly improve the controllability scores over its counterpart with original real captions. This further supports the recent finding that training on higher-quality synthetic captions can improve the model's prompt-following ability (Betker et al., 2023).

Qualitative Results. We first visualize the zero-shot generation results on the MS-COCO validation set. In Figure 3, we compare the generated images of our method that takes blobs as input and GLIGEN that takes bounding boxes as input, where the "ground-truth" real images are also shown as a reference. We can see that our generated images are much more aligned with the reference images from both two perspectives: 1) visual appearance of each object, where the object color, shape and style have been better captured by the blob descriptions; and 2) its spatial arrangement, where the object pose and orientation have been better captured by blob parameters. These results demonstrate that with blob representations, our method achieves much more fine-grained control over the generation.

We then visualize various image editing results by manually changing a blob representation (*e.g.*, either blob description or blob parameter) while keeping other blobs the same. See Appendix A.6 for more details. In Figure 4, we show that our method can enable different local editing capabilities by solely changing the corresponding blob descriptions. In particular, we can change the object color, category, and texture while keeping the unedited regions mostly the same. Furthermore, our method can also make different object repositioning tasks easily achievable by solely manipulating the corresponding blob parameters. For instance, we can move an object to different locations, change an object's orientation, and add/remove an object while also keeping other regions nearly unchanged. Note that we have not applied any attention map guidance or sample blending trick (Avrahami et al., 2022; Ge et al., 2023) during sampling, to preserve localization and disentanglement in the editing process. Thus, these results demonstrate that a well-disentangled and modular property naturally emerges in our blob-grounded generation framework.

4.1.3. Ablation Studies

In Table 2, we remove each component in our method separately and compare the resulting generation performance with our original design to highlight its impact.

Blob Embedding Concatenation. When blob text embeddings are directly passed to our Masked CA module, without

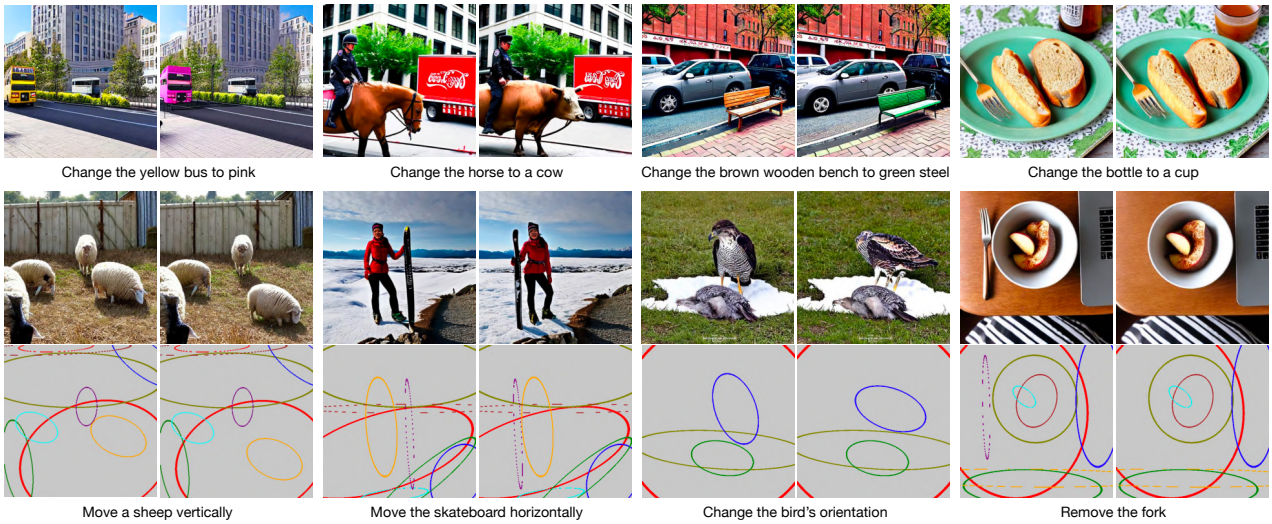


Figure 4. Various image editing results of our method on the MS-COCO validation set, where each example contains two generated images: (Left) original setting and (Right) after editing. The top row shows the local editing results where we only change the blob description and since the blob parameters stay the same after editing, we do not show blob visualizations. The bottom two rows show the object reposition results where we only change the blob parameter. All images are in resolution of 512×512.

Table 2. Ablation studies on each component of our method separately. Here, to save computational cost, we train on 140K random samples and evaluate using 10K samples. All the models are evaluated after training for 150K steps with a batch size of 256.

	FID ↓	mIOU ↑	rCLIP_t ↑	rCLIP_i ↑
Ours	9.14	0.4805	0.2859	0.8453
w/o blob emb. cat.	9.21	0.4706	0.2849	0.8435
w/o masking in CA	9.25	0.4401	0.2839	0.8410
w/o Masked CA	9.47	0.4471	0.2840	0.8412
w/o Gated SA	9.79	0.4898	0.2856	0.8422
w/o prompt tuning	9.65	0.4430	0.2795	0.8248

concatenating blob parameter embeddings, we see a small performance reduction consistently across all four metrics. It implies adding blob embedding concatenation can slightly improve both generation quality and controllability.

Masking in Masked CA. When we remove masking in our Masked CA module, *i.e.*, each blob text embedding does not attend only to visual features in the corresponding local region any more, we see a large performance drop on all three metrics: mIOU, rCLIP_t and rCLIP_i, along with a slightly higher FID. It implies incorporating masking in Masked CA mainly improves the controllability.

Masked CA. When we remove the whole Masked CA module (where we only use the Gated SA module from GLIGEN to enable the blob control), we see a large performance drop consistently across all four metrics. It implies the importance of Masked CA in achieving both good generation quality and controllability.

Gated SA. Removing Gated SA results in a much worse FID but has a slightly mixed impact on other three metrics. On average, its impact on controllability is small. We hypothesize that it improves generation quality because it increases the expressive power of the base model.

Prompt Tuning. When we use a different prompt for LLaVA-1.5 to generate blob descriptions, which does not specifically ask the image captioning model to focus on the object itself (see Appendix A.2 for details), we see a significant performance drop across all four metrics. It demonstrates the necessity of high data quality, in particular, a good blob description for each object.

4.2. LLMs for Blob Generation

In this section, we compare our method with previous approaches on compositional reasoning, by prompting LLMs to generate blob representations. We closely follow the evaluation protocol proposed by LayoutGPT (Feng et al., 2023a) for the comparison. We defer a comparison with other LLM-grounded generation methods (*i.e.,* LMD (Lian et al., 2023)) in a different setting to Appendix B.5.

4.2.1. Experiment Setup

Data Preparation. The original NSR-1K benchmark proposed by (Feng et al., 2023a) is mainly targeted for models that use bounding boxes as grounding input. To evaluate our method on NSR-1K, we need to replace bounding boxes with blob representations. Specifically, since almost all images in NSR-1K come from MS-COCO, we can convert their ground-truth segmentation maps to the blob parame-

Table 3. Evaluation of generation compositionality in terms of counting and spatial correctness on NSR-1K (Feng et al., 2023a). Given an input prompt, our method uses LLMs to generate blob representations for blob-grounded image generation. For image accuracy, we use Grounding DINO (Liu et al., 2023b) to detect bounding boxes from generated images.

Method	Numerical Reasoning				Spatial Reasoning	
	Layout Prec	Layout Rec	Layout Acc	Image Acc	Layout Acc	Image Acc
Text → Image						
SD-1.4 (Rombach et al., 2022)	-	-	-	44.82	-	32.58
SD-2.1 (Rombach et al., 2022)	-	-	-	48.49	-	32.20
SDXL (Podell et al., 2023)	-	-	-	46.49	-	46.59
Attend-and-Excite (SD-1.4) (Chefer et al., 2023)	-	-	-	47.91	-	35.98
Attend-and-Excite (SD-2.1) (Chefer et al., 2023)	-	-	-	50.33	-	36.74
Text → Layout → Image						
LayoutGPT (GPT3.5-chat) (Feng et al., 2023a)	75.40	86.23	74.62	61.54	81.98	72.01
LayoutGPT (GPT4) (Feng et al., 2023a)	**81.02**	85.63	78.11	60.25	86.23	74.35
Ours (GPT3.5-chat)	76.08	86.49	75.75	60.46	83.27	75.83
Ours (GPT4)	75.73	**86.77**	**78.67**	**62.96**	**90.23**	**80.16**

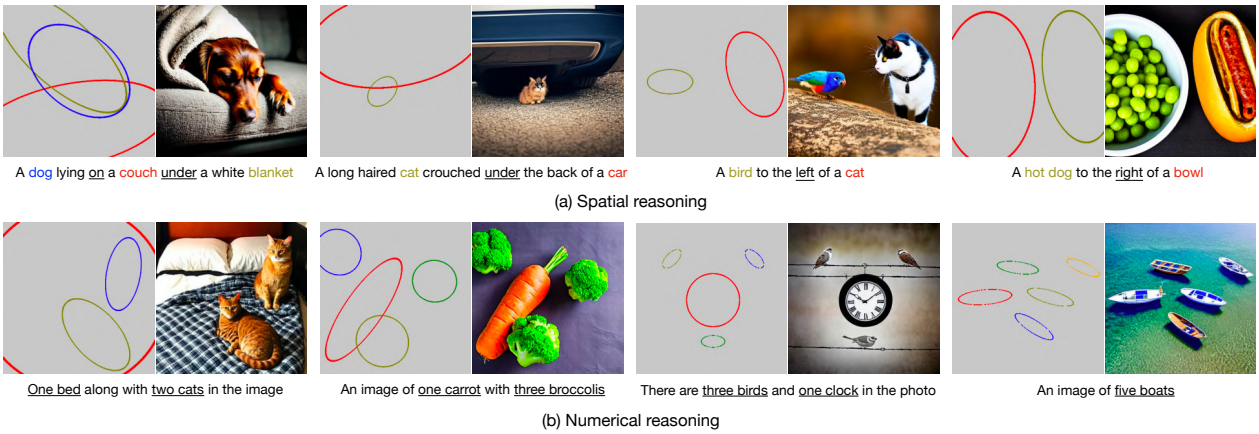


Figure 5. Qualitative results of our method on two compositional generation tasks of NSR-1K (Feng et al., 2023a): (a) spatial reasoning and (b) numerical reasoning. Given a caption, we prompt GPT4 to generate blob parameters (Left) and LLAMA-13B to generate blob descriptions (not shown in the figure), which are passed to our blob-grounded text-to-image model to synthesize an image (Right).

ters using the same ellipse fitting optimization algorithm, and we use LLaVA-1.5 to generate blob descriptions from cropped images. This results in 738 training and 264 testing examples[3] for spatial reasoning, and 38,698 training and 762 testing examples for numerical reasoning.

Evaluation Metrics. Similar to LayoutGPT (Feng et al., 2023a) that evaluates the layout planning performance of LLMs, we report precision (Prec), recall (Rec), and accuracy (Acc) based on generated layout counts and their spatial positions. To evaluate layout consistency from generated images, we observed that the detection model GLIP (Li et al., 2022), which was employed by LayoutGPT, frequently misses salient objects (see Appendix A.5 for details). Thus, we use the more recently proposed detector Grounding DINO (Liu et al., 2023b) to obtain more accurate bounding boxes. We then compute the average accuracy based on the detected bounding boxes and the ground-truth ones by following the same evaluation pipeline.

[3]Note that 19 images in the original test set are not from MS-COCO so we do not include them for evaluation.

4.2.2. Numerical and Spatial Reasoning

Quantitative Results. In Table 3, we show the performance of different models on both numerical reasoning and spatial reasoning. For the layout planning evaluation, we observe that both GPT3.5-chat and GPT4 can generate good blob parameters with our customized in-context learning procedure. For instance, the layout accuracies of GPT4 are 78.67% versus 78.11% on counting correctness and 90.23% versus 86.23% on spatial correctness for our method and LayoutGPT, respectively. For image evaluation, our method also achieves consistently better accuracies than LayoutGPT in both two tasks (counting: 62.96% versus 61.54%, spatial: 80.16% versus 74.35%). Compared with text-to-image models that do not incorporate layout planning, notably SDXL, our results distinctly highlight the significance of our blob-grounded framework in facilitating more reliable generation with better prompt following capabilities.

Qualitative Results. In Figure 5, we visualize the blobs generated by GPT4 and images generated by our blob-grounded model for various prompts from both spatial and

numerical reasoning tasks. We defer more visualization results of a side-by-side comparison among different models to Appendix B.4. We can see that our method can not only synthesize images that align well with the layouts generated by GPT4, which supports our quantitative results, but can also achieve high photo-realism. In particular, we observe that the out-of-context "crop-and-paste" effect less frequently appears in our generated images. Instead, objects have been rendered in a coherent and natural way, along with an appropriate background, to follow physical laws and visual commonsenses. For instance, three birds are standing on electric wires (b, third example) and the cat is curiously looking at a colorful bird on the left (a, third example).

5. Conclusions

In this work, we proposed to ground existing text-to-image generative models on blob representations for compositional generation. In particular, we applied the open-vocabulary segmentation and vision-language models to extract blob parameters and blob descriptions, which contain rich spatial and semantic information of images. We then introduced a blob-grounded generative model, termed BlobGEN, where a new masked cross-attention module that takes blobs as grounding inputs is injected into pre-trained text-to-image models. Furthermore, to leverage the compositional ability of LLMs for image generation, we designed a new in-context learning approach for LLMs to infer blob representations from text prompts. Finally, we performed extensive experiments to show the superior performance of our method in various generative tasks, including layout-guided generation, image editing and compositional reasoning.

Limitations. Our work has several limitations that we leave for the future work. First, even though blob representations can preserve fine-grained details of the image, we cannot solely rely on them to perfectly recover the original image, where a combination with inversion methods (Mokady et al., 2023) is still needed. Second, we see some failure cases for image editing (see Figure 16), which we believe advanced editing techniques (Avrahami et al., 2022) can be applied to alleviate. Third, we also see some failure cases in the numerical and spatial reasoning tasks (see Figure 17). It is an interesting challenge to further improve the integration between LLMs and blob-grounded generation.

Impact Statement

Our blob-grounded text-to-image model is based on a pre-trained text-to-image model, so it may inherit the potential biases and malicious information from the base model. Because our approach improves both the generation quality and the user controllability over the base model, on the positive side, it will improve the efficiency of human users in using generative models for creative work; on the negative side, similar to any generative AI tool, it can be used to generate malicious content. Furthermore, when augmented by LLMs for layout planning, our method will simplify the layout designing process, resulting in less burden on human designers for content creation. But we also note that the biases and misinformation from LLMs could also harm the use of our approach without proper regulation.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Avrahami, O., Lischinski, D., and Fried, O. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18208–18218, 2022.

Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y., and Ramesh, A. Improving image generation with better captions. *https://cdn.openai.com/papers/dall-e-3.pdf*, 2023.

Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.-H., Murphy, K., Freeman, W. T., Rubinstein, M., et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., and Cohen-Or, D. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023.

Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023a.

Chen, M., Laina, I., and Vedaldi, A. Training-free layout control with cross-attention guidance. *arXiv preprint arXiv:2304.03373*, 2023b.

Chen, W., Hu, H., Saharia, C., and Cohen, W. W. Re-imagen: Retrieval-augmented text-to-image generator. In *The Eleventh International Conference on Learning Representations*, 2022.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34: 19822–19835, 2021.

Du, Y., Li, S., and Mordatch, I. Compositional visual generation with energy based models. In *Advances in Neural Information Processing Systems*, 2020.

Epstein, D., Park, T., Zhang, R., Shechtman, E., and Efros, A. A. Blobgan: Spatially disentangled scene representations. *European Conference on Computer Vision (ECCV)*, 2022.

Epstein, D., Jabri, A., Poole, B., Efros, A. A., and Holynski, A. Diffusion self-guidance for controllable image generation. *arXiv preprint arXiv:2306.00986*, 2023.

Feng, W., He, X., Fu, T.-J., Jampani, V., Akula, A. R., Narayana, P., Basu, S., Wang, X. E., and Wang, W. Y. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *The Eleventh International Conference on Learning Representations*, 2022.

Feng, W., Zhu, W., Fu, T.-j., Jampani, V., Akula, A., He, X., Basu, S., Wang, X. E., and Wang, W. Y. Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393*, 2023a.

Feng, Y., Gong, B., Chen, D., Shen, Y., Liu, Y., and Zhou, J. Ranni: Taming text-to-image diffusion for accurate instruction following. *arXiv preprint arXiv:2311.17002*, 2023b.

Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pp. 89–106. Springer, 2022.

Ge, S., Park, T., Zhu, J.-Y., and Huang, J.-B. Expressive text-to-image generation with rich text. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7545–7556, 2023.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Huang, K., Sun, K., Xie, E., Li, Z., and Liu, X. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *arXiv preprint arXiv:2307.06350*, 2023a.

Huang, L., Chen, D., Liu, Y., Shen, Y., Zhao, D., and Zhou, J. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023b.

Koh, J. Y., Fried, D., and Salakhutdinov, R. Generating images with multimodal language models. *arXiv preprint arXiv:2305.17216*, 2023.

Kumari, N., Zhang, B., Zhang, R., Shechtman, E., and Zhu, J.-Y. Multi-concept customization of text-to-image diffusion. In *CVPR*, 2023.

Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965–10975, 2022.

Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., and Lee, Y. J. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023.

Lian, L., Li, B., Yala, A., and Darrell, T. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*, 2023.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023a.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023b.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Mokady, R., Hertz, A., Aberman, K., Pritch, Y., and Cohen-Or, D. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.

Nie, W., Vahdat, A., and Anandkumar, A. Controllable and compositional generation with latent-space energy-based models. *Advances in Neural Information Processing Systems*, 34:13497–13510, 2021.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Phung, Q., Ge, S., and Huang, J.-B. Grounded text-to-image synthesis with attention refocusing. *arXiv preprint arXiv:2306.05427*, 2023.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and 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.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

Sheynin, S., Ashual, O., Polyak, A., Singer, U., Gafni, O., Nachmani, E., and Taigman, Y. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849*, 2022.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Sun, Q., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, Y., Gao, H., Liu, J., Huang, T., and Wang, X. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33: 7537–7547, 2020.

Wang, X. et al. Deep learning in object recognition, detection, and segmentation. *Foundations and Trends® in Signal Processing*, 8(4):217–382, 2016.

Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., and Duan, N. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023a.

Wu, W., Li, Z., He, Y., Shou, M. Z., Shen, C., Cheng, L., Li, Y., Gao, T., Zhang, D., and Wang, Z. Paragraph-to-image generation with information-enriched diffusion model. *arXiv preprint arXiv:2311.14284*, 2023b.

Xiao, G., Yin, T., Freeman, W. T., Durand, F., and Han, S. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *arXiv preprint arXiv:2305.10431*, 2023.

Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., and De Mello, S. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2955–2966, 2023.

Yang, Z., Wang, J., Gan, Z., Li, L., Lin, K., Wu, C., Duan, N., Liu, Z., Liu, C., Zeng, M., et al. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14246–14255, 2023.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. *IEEE International Conference on Computer Vision (ICCV)*, 2023.

Zheng, G., Zhou, X., Li, X., Qi, Z., Shan, Y., and Li, X. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22490–22499, 2023.

Zhou, Y., Li, C., Chen, C., Gao, J., and Xu, J. Lafite2: Few-shot text-to-image generation. *arXiv preprint arXiv:2210.14124*, 2022.

A. Implementation Details

A.1. Synthetic Global Captions

To train our model, we use synthetic global captions instead of the noisy real captions from the original image-text dataset, as evidenced by recent findings that high-quality synthetic captions can improve the model's prompt following performance (Betker et al., 2023). To generate synthetic captions, we use LLaVA-1.5 as the image captioner with the prompt: "Given the caption of this image '<real caption>', describe this image concisely.". Compared with the standard prompt: "Describe this image concisely.", it can capture the regional information that LLaVA-1.5 cannot provide to incorporate the real caption into the prompt, as shown in Figure 6.

Prompt: Describe this image concisely

The image features a woman with long hair, possibly Princess Leia from Star Wars, posing for a picture. She is wearing a white shirt and has her hand under her chin. The woman appears to be looking at the camera, possibly with a hint of sadness in her expression. The scene is set in a room with a couch in the background, and a tie can be seen hanging nearby.

Prompt: Given the caption of this image '*Carrie Fisher: October 21st 1956 - December 27th 2016*', describe this image concisely

The image is a black and white photo of Carrie Fisher, who played Princess Leia in the Star Wars franchise. She is wearing a white shirt and has her hand under her chin, looking at the camera. The photo captures a moment from her life, and it serves as a tribute to the iconic actress who passed away in 2016.

Prompt: Describe this image concisely

The image features a large building situated next to a body of water, possibly a river or a lake. The building appears to be a castle or a church, with a clock tower visible on its side. The scene is captured in black and white, giving it a classic and timeless feel. The building is surrounded by trees, adding to the serene atmosphere of the location.

Prompt: Given the caption of this image '*Cathedral Notre Dame of Paris*', describe this image concisely

The image is a black and white photograph of the Cathedral Notre Dame of Paris, a historic and iconic building located next to a body of water. The cathedral's clock tower is prominently visible, and the overall scene evokes a sense of history and architectural beauty.

Figure 6. Comparison between two prompts for LLaVA-1.5 to generate synthetic global captions, where we show the response of LLaVA-1.5 for each prompt on an image. We can see that incorporating the original real caption into the prompt can capture the specialized information that LLaVA-1.5 cannot provide.

A.2. Prompt Tuning for Blob Descriptions

When we use the image captioner (*e.g.*, LLaVA-1.5) to describe the cropped image as depicted by the blob parameter, we need to make the blob description to capture fine-grained visual features of the local region. To this end, we consider two types of prompts: one is to explicitly ask for visual features, and the other one is to provide the context of what the full image is about. Specifically, we compare the following two prompts:

- "Can you briefly describe this <category name> in the close-up and focus on its color, appearance, size, and style, etc.?"
- "While in the original full image, '<global caption>', can you briefly describe this <category name> in the close-up?"

Note that we assume the category name for the object in the local region is given, which can come as an output of an off-the-shelf image segmentation model. As evidenced by the matrix scores in Table 2, the first prompt works better than the second one. Therefore, we use the first prompt to generate blob descriptions for all our experiments.

A.3. In-Context Learning for LLMs

We introduce a new in-context learning approach that contains two separate procedures to generate blob parameters and blob descriptions, respectively.

A.3.1. BLOB PARAMETER GENERATION

We mainly use GPT3.5-chat (Ouyang et al., 2022) and GPT4 (Achiam et al., 2023) to generate blob parameters. To make GPT3.5-chat/GPT4 better understand and infer the numerical values, we follow (Feng et al., 2023a) to use the CSS format to represent blob parameters. The final prompt for GPT3.5-chat/GPT4 consists of a system prompt that instructs the blob parameter generation, k demonstration examples, and the test prompt (usually a global caption). In Table 4, we use `"a teddy bear to the left of a bed"` as the text prompt of a real example to show the system prompt.

Table 4. The system prompt for GPT3.5-chat/GPT4 to generate blob parameters for the example `"a teddy bear to the left of a bed"`. Note that we have ignored the wrapped prompts specifically for GPT3.5-chat/GPT4, including `"role: system, content: "`, `"role: user, content: "` and `"role: assistant, content: "` for the sake of readability.

Instruction: Given a sentence prompt that will be used to generate an image, plan the layout of the image. The generated layout should follow the CSS style, where each line starts with the object name and is followed by its absolute position depicted as an ellipse. Formally, each line should be like "object {major-radius: ?px; minor-radius: ?px; cx: ?px; cy: ?px; angle: ?}". The image is 512px wide and 512px high. Therefore, all properties of the positions (including major-radius, minor-radius, cx and cy) should not exceed 512px, and the value of angle is in degree and it should be within [0, 180]. Finally, we prefer all objects to be large (i.e., each ellipse better has a large major-radius), if possible. Prompt: a teddy bear to the right of a cat Layout: teddy-bear {major-radius: 162px; minor-radius: 76px; cx: 444px; cy: 258px; angle: 96} cat {major-radius: 137px; minor-radius: 116px; cx: 149px; 236cy: ?px; angle: 3} [ADDITIONAL $k-1$ DEMONSTRATION EXAMPLES REMOVED FOR SIMPLICITY] Prompt: a teddy bear to the left of a bed Layout:

A.3.2. BLOB DESCRIPTION GENERATION

We mainly use LLaMA-13B to generate blob descriptions. Because the blob descriptions are just a list of text sentences, we use the simple text format for its generation, where the category name of an object is still used as a blob separator. The final prompt for LLaMA-13B includes a system prompt that instructs the blob description generation, k demonstration examples, and the test prompt (usually a global caption). In Table 5, we also use `"a teddy bear to the left of a bed"` as a real example to show the system prompt.

Table 5. The system prompt for LLaMA-13B to generate blob descriptions for the example `"a teddy bear to the left of a bed"`.

Instruction: Given a sentence prompt that will be used to generate an image, plan the region descriptions of the image, where each line starts with the object name. For example, each line should be like "cat {The cat in the close-up is a large, gray and white cat with a fluffy appearance. The cat's size and style suggest that it is a domesticated cat, likely a house cat, and it is comfortable in its environment. The cat's gray and white coloration adds to its unique and visually appealing appearance.}". The generated region description should describe the object in the close-up and focus on its color, appearance, size, and style, etc. Prompt: a teddy bear to the right of a cat Region Desc: teddy-bear {The teddy bear in the close-up is white and has a large size. It is sitting next to a pink stuffed animal, which appears to be a dragon or a panda. The teddy bear is positioned on a bed, and it is surrounded by other stuffed animals, creating a cozy and playful scene.} cat {The cat in the close-up is a large, striped tabby cat. It has a distinctive black and brown striped pattern on its fur, which is quite noticeable. The cat appears to be sitting or standing on top of a stuffed animal, possibly a teddy bear, which adds a playful and curious element to the scene. The cat's size and style give it a unique and eye-catching appearance, making it an interesting subject for a close-up photo.} [ADDITIONAL $k-1$ DEMONSTRATION EXAMPLES REMOVED FOR SIMPLICITY] Prompt: a teddy bear to the left of a bed Region Desc:

A.4. Examples of Blob Representations

```
"Caption":
        "The image features a woman wearing a white dress, which appears to be a wedding
        gown designed by Caroline Castigliano. She is standing in front of a wall,
        possibly in a studio setting. The dress is elegant and has a flowing skirt,
        giving it a sophisticated and stylish appearance. The woman's pose suggests that
        she is either posing for a photoshoot or showcasing the dress for potential
        buyers.",
"Blob parameters":
[
        [0.4695, 0.3939, 0.4875, 0.4631, 0.994, 0.5772],
        [0.4826, 0.646, 0.3555, 0.2395, 0.5352, 0.9988],
        [0.5407, 0.9478, 0.4909, 0.0579, 1.0, 0.4941],
        [0.8141, 0.078, 0.1054, 0.1033, 0.9787, 0.3558]
]
"Blob descriptions":
[
        "In the close-up, the wall appears to be white and has a smooth, clean surface.
        It is likely a plain or minimalist wall, which serves as a neutral backdrop for
        the woman wearing a white dress. The wall does not have any visible patterns,
        textures, or decorations, allowing the focus to remain on the woman and her
        attire.",
        "The person in the close-up is a woman wearing a white dress. The dress appears
        to be a wedding dress, as it is a white ball gown with a long skirt. The woman
        is standing in front of a wall, and she is wearing a necklace, which adds to the
        elegance of her outfit. The close-up view of the image highlights the details of
        the dress and the woman's overall appearance, emphasizing the beauty and grace
        of the scene.",
        "In the close-up, the floor appears to be a white, wooden surface. It has a
        clean and polished look, and it seems to be part of a room with a white
        tablecloth. The floor's size is not clearly visible, but it is described as a
        \"floor\" which suggests it is a significant part of the room. The style of the
        floor is simple and elegant, likely complementing the overall design and
        aesthetics of the room.",
        "In the close-up, the flower is a beautiful pink cherry blossom. It has a
        delicate and elegant appearance, with a soft pink petal color. The flower is
        small in size, and its style is reminiscent of a traditional Japanese cherry
        blossom. The flower is surrounded by other blossoms, creating a visually
        appealing and harmonious arrangement."
]
```

```
"Caption":
        "The image depicts a snowy scene with three wolves standing together in a
        forest. They are positioned close to each other, with one wolf on the left side,
        another in the center, and the third on the right side of the scene. The wolves
        are surrounded by trees, and the snow-covered ground adds to the wintry
        atmosphere. The image is a painting of the wolves, capturing their natural
        habitat and behavior.",
"Blob parameters":
[
        [0.4884, 0.75, 0.4302, 0.335, 0.9999, 0.5119],
        [0.5161, 0.1659, 0.4058, 0.2367, 0.9997, 0.5165],
        [0.1986, 0.3995, 0.1566, 0.1382, 0.9918, 0.5902],
        [0.8894, 0.4035, 0.1194, 0.132, 0.0291, 0.6682],
        [0.4845, 0.5254, 0.1479, 0.1051, 0.5465, 0.9978]
]
"Blob descriptions":
[
        "The snow in the close-up is white and appears to be freshly fallen, creating a
        pristine and serene environment. The snow covers the ground, creating a blanket-
        like appearance, and it is quite thick, indicating a significant amount of
        snowfall. The snow is also covering the trees, which adds to the overall beauty
        of the scene. The style of the snow is smooth and even, without any visible
        tracks or disturbances, giving it a uniform and untouched appearance.",
        "In the close-up, the tree is covered in snow, giving it a white appearance. It
        is a tall tree, with a slender trunk and branches that extend outwards. The
        tree's branches are thin and delicate, creating a graceful and elegant
        silhouette against the snowy background. The tree's color is predominantly white
        due to the snow covering its branches and trunk, creating a striking contrast
        against the surrounding snowy landscape.",
        "The dog in the close-up is a large, furry creature with a wolf-like appearance.
        It has a brown and white color pattern, giving it a distinctive and majestic
        look. The dog is standing in the snow, which adds to its natural and wild
        appearance. Its size is quite impressive, as it is a big and powerful animal,
        likely capable of handling various challenges and environments. The style of the
        dog is reminiscent of a wolf, which suggests that it might be a breed or a dog
        with a strong resemblance to wolves.",
        "The dog in the close-up is a large, furry creature with a wolf-like appearance.
        It has a brown and white color pattern, which gives it a distinctive look. The
        dog's fur is long, and it appears to be standing on a snowy surface, possibly a
        mountain. The image is a painting, which adds to the artistic and visually
        striking nature of the scene. The dog's size is quite impressive, as it
        dominates the frame of the painting, making it the focal point of the artwork.",
        "The image features a large, brown dog with a wolf-like appearance. It has a
        prominent snout and a furry body, giving it a wild and majestic look. The dog is
        standing in a snowy environment, which adds to its natural and rugged
        appearance. The close-up view of the dog allows for a detailed examination of
        its features, such as its eyes, ears, and facial expressions, highlighting its
        unique and striking appearance."
]
```

```
"Caption":
        "The image features a scenic mountain road with two red sports cars parked on
        the side of the road. The cars are Porsche 718 Boxster T and 718 Cayman T
        models, both with their tops down, giving the impression of a sunny day. The
        cars are parked in front of a beautiful mountain range, creating a picturesque
        scene.",
"Blob parameters":
[
        [0.559, 0.2318, 0.4101, 0.2957, 0.9894, 0.6025],
        [0.4498, 0.8637, 0.4206, 0.174, 0.9984, 0.5399],
        [0.4141, 0.5367, 0.452, 0.1057, 0.9923, 0.5875],
        [0.4463, 0.6857, 0.1217, 0.0603, 0.9999, 0.4914],
        [0.8207, 0.733, 0.1665, 0.0322, 0.9878, 0.61],
        [0.1787, 0.6386, 0.1004, 0.0459, 0.9996, 0.4798],
        [0.7849, 0.6843, 0.1994, 0.0139, 0.9915, 0.592]
]
"Blob descriptions":
[
        "The sky in the close-up is a pale blue color, which gives it a serene and calm
        appearance. It is a large, open sky that spans across the entire image, creating
        a sense of vastness and freedom. The sky's style is minimalistic, with no
        visible clouds or other elements that could distract from the main focus of the
        scene, which is the majestic mountain range. The close-up perspective emphasizes
        the beauty and grandeur of the mountains, making them the central point of
        interest in the image.",
        "The road in the close-up is a wet, black asphalt surface. It appears to be a
        small, curvy road, which is suitable for the red sports car parked on it. The
        road's style suggests that it might be a scenic route or a street in a
        residential area, given the presence of the sports car. The wet surface
        indicates that it has recently rained or there might be some moisture on the
        road, which could affect the vehicle's traction and handling.",
        "The mountain in the close-up is green and has a grassy appearance. It is quite
        large and has a distinctive shape, making it visually appealing. The mountain is
        situated in a valley, and the red sports car is parked nearby, adding a
        contrasting element to the scene. The combination of the red sports car and the
        green mountain creates a striking and vibrant image.",
        "The car in the close-up is a red sports car with a convertible top. It appears
        to be a luxury vehicle, possibly a Porsche, and has a sleek and stylish design.
        The car is parked on the street, and the focus is on its color, appearance, and
        size. The red color of the car stands out, and its convertible top adds a sense
        of openness and freedom to the vehicle. The car's overall appearance is sporty
        and elegant, making it an attractive and eye-catching sight on the street.",
        "In the close-up image, the grass appears to be green and has a somewhat blurry
        appearance. It is situated near the water, possibly on the shore or the edge of
        a lake. The grass is relatively small and sparse, with a few patches of it
        visible in the image. The style of the grass can be described as natural and
        untamed, adding to the overall scenic and serene atmosphere of the scene.",
        "The car in the close-up is a red sports car, likely a convertible, with a sleek
        and aerodynamic design. Its color is predominantly red, and it appears to be a
        luxury vehicle. The car is parked on a street, and the focus of the image is on
        the back end of the car, showcasing its rear tire and the overall shape.",
        "The fence in the close-up is black and appears to be made of metal. It is quite
        large and spans across the image, covering a significant portion of the field.
        The fence is situated near a hill. The fence's size and style suggest that it
        serves as a boundary or a barrier for the field, possibly to keep animals or
        people from entering or leaving the area."

]
```

```
"Caption":
        "The image captures a thrilling moment of a female bicyclist in action,
        performing a trick on her bike. She is wearing a red outfit and is skillfully
        riding her bike on a ramp. The scene suggests that she is participating in a
        competition, as she is the series leader. The image showcases her athleticism
        and talent in the sport of bicycle racing.",
"Blob parameters":
[
        [0.4683, 0.5329, 0.4722, 0.5994, 0.707, 0.9552],
        [0.7641, 0.3703, 0.2273, 0.1983, 0.916, 0.2226],
        [0.3426, 0.3848, 0.2302, 0.1385, 0.9999, 0.5095]
]
"Blob descriptions":
[
        "In the close-up, the fence is black and white, and it appears to be a mesh or
        woven design. The fence is quite large, covering a significant portion of the
        image. The style of the fence suggests that it may be a part of a wall or a
        barrier, possibly in an urban or industrial setting.",
        "The person in the close-up is wearing a red and orange outfit, which appears to
        be a racing suit. The outfit is designed to provide flexibility and comfort
        while riding a motorcycle. The person is wearing a helmet, which is an essential
        safety gear for motorcycle riders. The close-up view of the person highlights
        their attire and the motorcycle they are riding, showcasing the rider's skill
        and dedication to the sport.",
        "The bicycle in the close-up is a mountain bike with a black frame and a number
        1 on the front. It appears to be a racing bike, as it is being ridden by a
        person wearing a red outfit. The bike has a large tire, which is typical for
        mountain bikes, and it is designed for off-road cycling. The close-up view of
        the bike allows us to see its details, such as the gears, brakes, and other
        components that make up the bicycle."
]
```

Figure 7. Four examples of decomposing a scene into blob representations, where we visualize the blob ellipses by the side of each image, and also include the synthetic global captions that are used to train our model. All blob parameters are normalized to the range of [0, 1].

A.5. Unreliability of GLIP

LayoutGPT has used GLIP (Li et al., 2022) to detect objects from generated images for evaluating the spatial and numerical correctness of the generation. However, we found that GLIP could consistently miss objects in a generated image, as shown in Figure 8. As a result, we use a more recently developed detection method, called Grounding DINO (Liu et al., 2023b), as a better proxy for our evaluation.

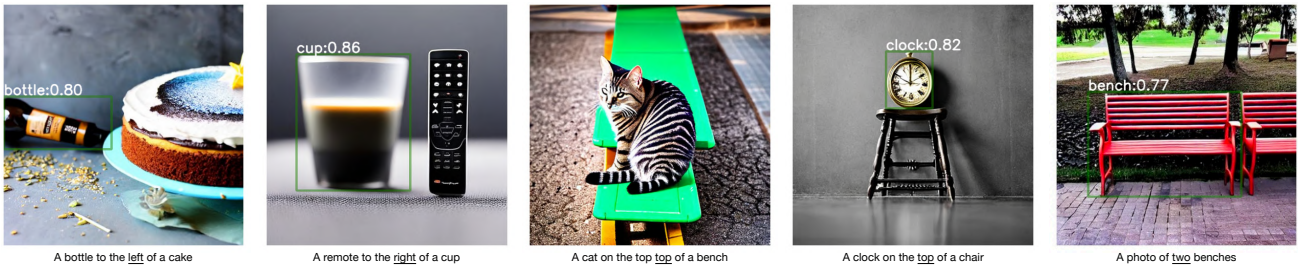


Figure 8. A few examples that GLIP fails to detect all correctly generated objects from the input prompts. All detected objects have been marked a bounding box with a prediction score.

A.6. Detailed Process of Image Editing

Given an image generated from BlobGEN and its conditioning information (including blob representations, global text, and initial Gaussian noise), we only change its blob representations and then pass the new blob representations along with the original global text and the original initial Gaussian noise to BlobGEN to get the edited image. Note that we follow the standard denoising sampling to generate the edited image without relying on any advanced editing technique (i.e., attention guidance or image blending), implying good disentanglement of BlobGEN.

When we change a blob representation (blob parameter and blob description), we can either change the blob parameter (defined as $\tau = [c_x, c_y, a, b, \theta]$) for repositioning, rotating and moving the object, or change the blob description (defined as an object-level text caption) for editing the object's appearance and other visual features. Moreover, we can completely remove/add a blob representation to remove/add the object.

B. Additional Qualitative Results

B.1. Blob Representations Capture Irregular, Large Objects and Background

To show blobs can capture irregular, large objects and background, we provide extra qualitative results in Figure 9.

Irregular Objects. Examining Figure 9(a), the first three rows show that blob representations can capture "a person with waving hands outside the blob". Note that "person" blobs allow the person's hands to be outside their ellipse region and still capture their pose accurately. The last two rows show that blob representations can capture "a river with irregular shapes". Moreover, the fourth row in Figure 9(b) shows that blob representations can capture "the great wall with a zigzag shape".

Two factors allow our blob representation to capture irregular shapes: 1) Training data contains many irregular objects where our blobs are designed to allow some parts of the irregular object to be outside the blob ellipse. Thus, the trained model can quickly learn this design from the training data. 2) More fine-grained shape details of the irregular objects can be captured by the text sentences of blob descriptions (e.g., a blob description may contain "a zigzag river" or "an upside-down person in the air with two arms widely open"), which complement the spatial depiction of blob parameters. 3) In some cases with very large irregular objects, such as "the great wall with a zigzag shape", multiple blobs can be used to capture each individual part of the object, or neighboring objects (e.g. sand, rock, etc) can help with creating a particular irregular shape.

Large Objects and Background. Examining Figure 9(b), the first two rows show that blob representations can capture the large "sky" of a similar color and mood to that in the reference real image. The second row shows that blob representations can capture the large "pier" of a similar color, pattern, and shape to that in the reference real image. The same applies to the "foggy grass" and its reflection in the water in the third row and the large mountains in the last two rows.

The rationale behind this capability is that: 1) We can always use as large blob ellipses as possible to fit the large objects,

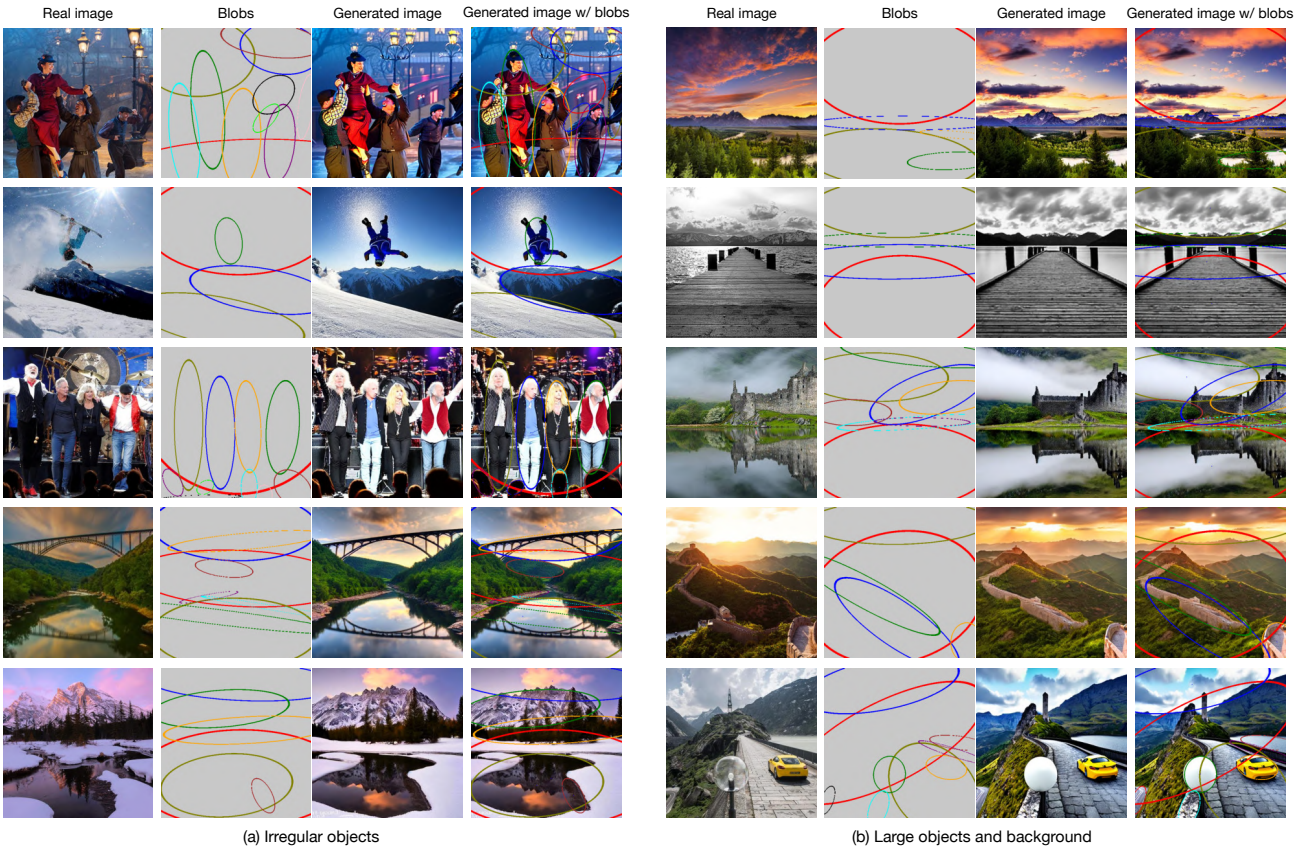


Figure 9. Examples of blob-grounded generation capturing (a) irregular objects (e.g. "person with waving hands" and "river"), and (b) large objects and background (e.g., "sky", "mountain" and "grass"). From left to right in each row, we show the reference real image, blobs, generated image and generated image superposed on blobs. Better zoom in for better visualization.

and we do not necessarily restrict the whole blob ellipse area to be within the pixel range. This provides more flexibility for blobs to capture extremely large objects and backgrounds. 2) More importantly, the text sentence description from the blob representation can effectively help capture the fine-grained details (i.e., color, appearance, texture, etc.) of the large object.

B.2. Zero-Shot Generation on MS-COCO

We show more zero-shot layout-grounded generation results of GLIGEN and our method in Figure 10. In general, our method can capture more fine-grained details of the original model, using dense blob representations.

B.3. Image Editing on MS-COCO

We show more image editing results of our method in Figure 11. Note that we perform various image editing tasks by merely modifying the blob parameter for object repositioning, or modifying the blob description for local object/attribute manipulation. We do not use any advanced image editing technique, such as attention mask guidance (Ge et al., 2023) or sample blending (Avrahami et al., 2022), to maintain localization and disentanglement for editing. As we can see, our method can enable various image editing capabilities, including changing the fine-grained orientation of an object that previous box layouts can hardly work out. With these promising editing results, we believe our blob-grounded generative model in general has a well-disentangled property, with a modular control over generation for each local region depicted by blob representations.

B.4. Numerical and Spatial Reasoning

We show more numerical and spatial reasoning results of our method and previous approaches in Figures 12 and 13, respectively. In addition, we visualize the layouts inferred by GPT4 for both GLIGEN and our method. We can see that

all methods without layout planning in the middle always fail in the task, including SDXL (Podell et al., 2023) that has a large model size and more advanced training procedures, and Attention-and-Excite (Chefer et al., 2023) that has an explicit attention guidance. Compared with LayoutGPT based on GLIGEN, our method can not only generate images with better spatial and numerical correctness, but also in general has better visual quality with less "copy-and-paste" effect.

B.5. Using Fixed In-context Examples without Retrieval

B.5.1. Comparison between with and without Retrieval

Here, we show retrieval from a large blob dataset for getting in-context demo examples is *not necessary* for our method. The main reason why we use retrieval as a demonstration is that we want to follow the exact evaluation protocol of LayoutGPT to make a fair comparison with LayoutGPT on the NSR-1K benchmark. To make a direct comparison between using fixed in-context examples ("fixed") vs. using retrieved in-context examples ("retrieval"), we summarize the results in Table 6. We can see that although using the fixed in-context examples underperforms using retrieved in-context examples in both spatial and numerical reasoning tasks, the performance gap is not large. It implies the effectiveness of our method with the use of fixed in-context examples.

Table 6. Comparison between using fixed in-context examples ("fixed") vs. using retrieved in-context examples ("retrieval") for our method in terms of counting and spatial correctness on NSR-1K (Feng et al., 2023a).

Method	Numerical Reasoning				Spatial Reasoning	
	Layout Prec	Layout Rec	Layout Acc	Image Acc	Layout Acc	Image Acc
retrieval	**75.73**	**86.77**	**78.67**	**62.96**	90.23	**80.16**
fixed	71.91	86.07	77.95	60.74	**90.80**	77.84

B.5.2. Comparison between Our Method and LMD (Lian et al., 2023)

In Table 7, we show when we use the same 8 fixed in-context demo examples (without retrieval) to prompt GPT-4 for blob generation, our method outperforms LMD (Lian et al., 2023), a strong baseline that has used fixed in-context examples for prompting LLMs to generate bounding boxes.

Table 7. Comparison between our method and LMD (Lian et al., 2023) in terms of counting and spatial correctness on NSR-1K (Feng et al., 2023a), where both two methods use the same 8 fixed in-context demo examples to prompt GPT4 for layout generation.

Method	Numerical Reasoning				Spatial Reasoning	
	Layout Prec	Layout Rec	Layout Acc	Image Acc	Layout Acc	Image Acc
LMD (Lian et al., 2023)	71.76	85.96	**78.02**	57.61	83.86	73.50
Ours	**71.91**	**86.07**	77.95	**60.74**	**90.80**	**77.84**

We also show the qualitative results of comparing our method and LMD in Figure 14. We observe that: 1) In some complex examples, such as "a boat to the right of a fork" in Figure 14(a) and "there are one car with three motorcycles in the image" in Figure 14(b), LMD fails but our method works. It confirms the quantitative results in Table 7. 2) LMD consistently has the "copy-and-paste" artifacts in its generated images in which objects are put together without matching their context (since it modifies the diffusion sampling process to enforce compositionality, which may largely deviate from the original data denoising trajectory), such as the example of "a teddy bear to the left of a potted plant" in Figure 14(a) and the example of "a photo of four boats" in Figure 14(b). In contrast, our generated images look much more natural. Besides, the more sophisticated sampling process in LMD makes the image generation slower. For instance, we observed that the sampling time of LMD is around 3$\times$ slower than our method.

B.6. Blob Control over Using LLMs

We add more qualitative results of blob control over using LLMs in Figure 15. To further demonstrate the blob control from LLMs, we consider four cases of how LLMs understand compositional prompts for correct visual planning: (a) swapping object name ("cat" $\leftrightarrow$ "car"), (b) changing relative reposition ("left" $\leftrightarrow$ "right"), (c) changing object number ("three" $\leftrightarrow$ "four"), and (d) swapping object number ("one bench & two cats" $\leftrightarrow$ "two benches & one cat"). We can see that LLMs

have the ability of capturing the subtle differences when the prompts are modified in an "adversarial" manner. Besides, we show that LLMs can generate diverse and feasible visual layouts from the same text prompt, which BlobGEN can use for synthesizing correct images of high quality and diversity.

B.7. Failure Cases

We show some failure cases when our method perform image editing and compositional reasoning tasks in Figures 16 and 17, respectively. For image editing, we see a few failure patterns: 1) the background also changes largely when the object to be edited is covered by the "background" blob, 2) the object itself changes when we only change its color or move its position, and 3) the edited object does not quite follow the instruction. We believe a combination with other image editing techniques will greatly reduce the failure rate. For compositional reasoning, we find that, on one hand, our method might not be perfectly robust to the LLM-generated blobs, and thus may have the "copy-and-paste" effect or completely fail when conditioning on some challenging or even wrong blobs. On the other hand, blob guidance does not prevent our method from generating more similar objects in other empty places, so our method may generate more objects than as instructed.

Figure 10. Zero-shot layout-grounded generation results of GLIGEN and our method on MS-COCO validation set. In each row, we visualize the reference real image (Left), bounding boxes and GLIGEN generated image (Middle), blobs and our generated image (Right). All images are in resolution of 512×512.

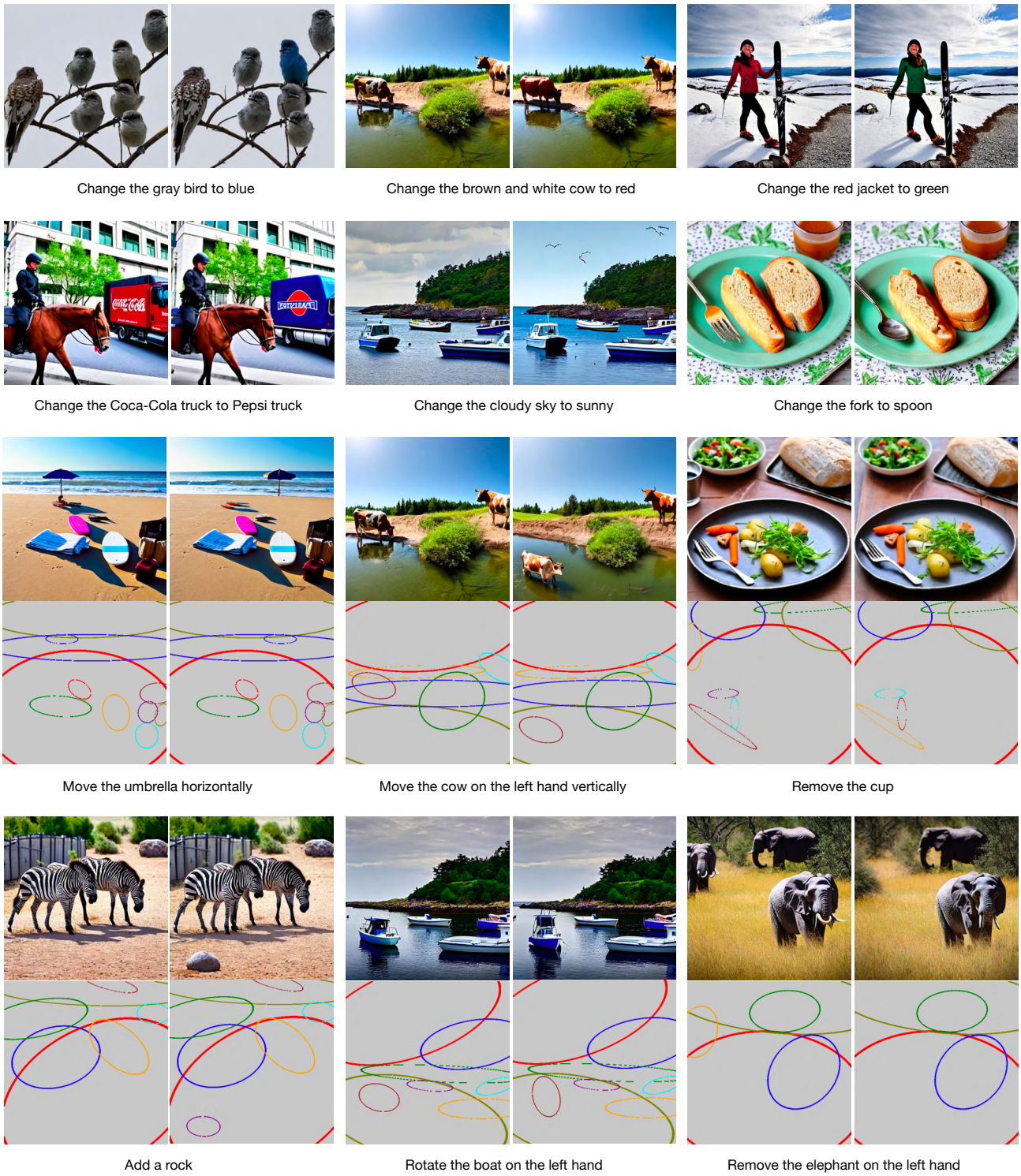


Figure 11. Various image editing results of our method on the MS-COCO validation set, where each example contains two generated images: (Left) original setting and (Right) after editing. The top two rows show the local editing results where we only change the blob description and since the blob parameters stay the same after editing, we do not show blob visualizations. The bottom four rows show the object reposition results where we only change the blob parameter. All images are in resolution of 512×512.

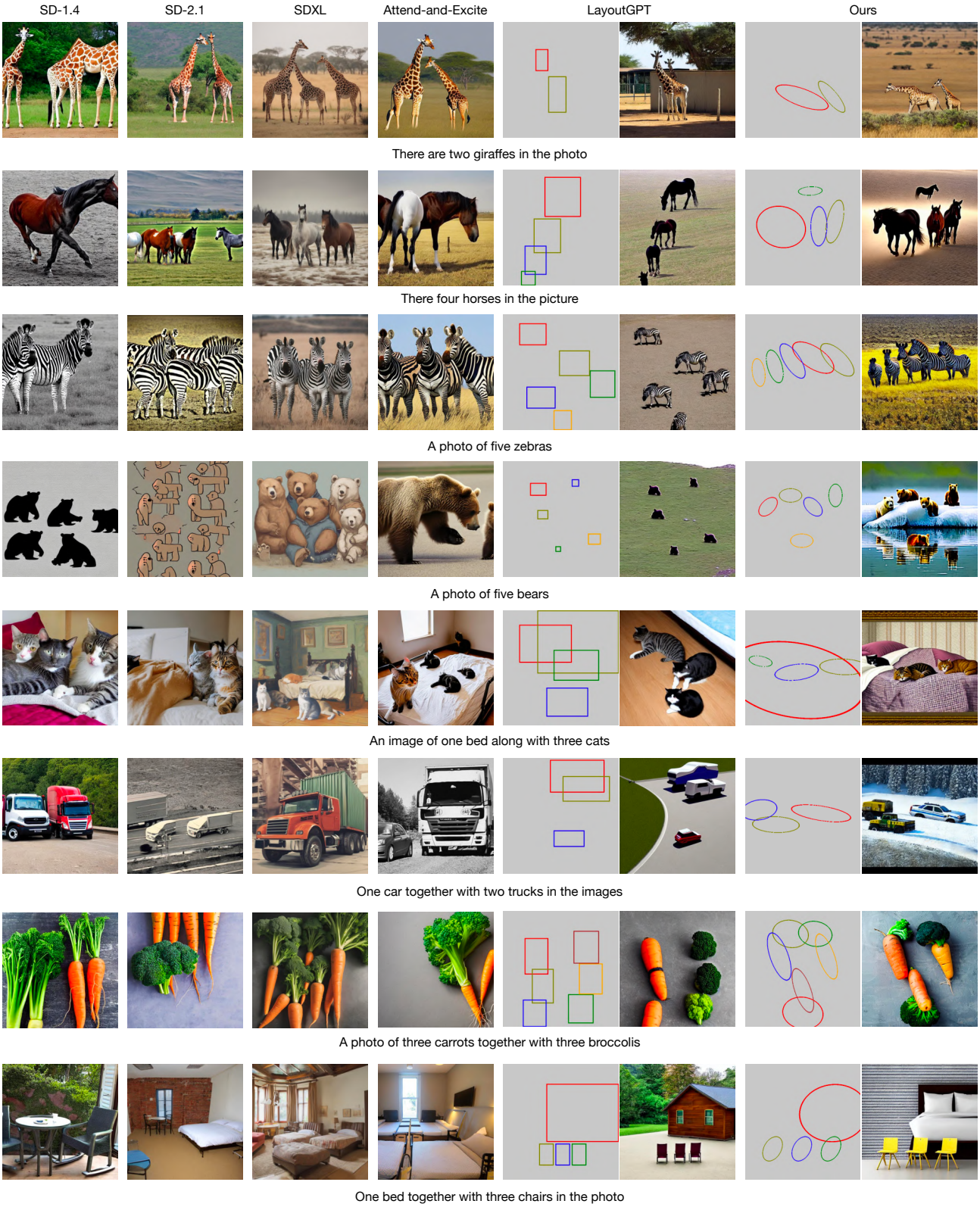


Figure 12. Qualitative results of various methods on numerical reasoning tasks. In our method, given a caption, we prompt GPT4 to generate blob parameters (Left) and LLAMA-13B to generate blob descriptions (not shown in the figure), which are passed to our blob-grounded text-to-image generative model to synthesize an image (Right).

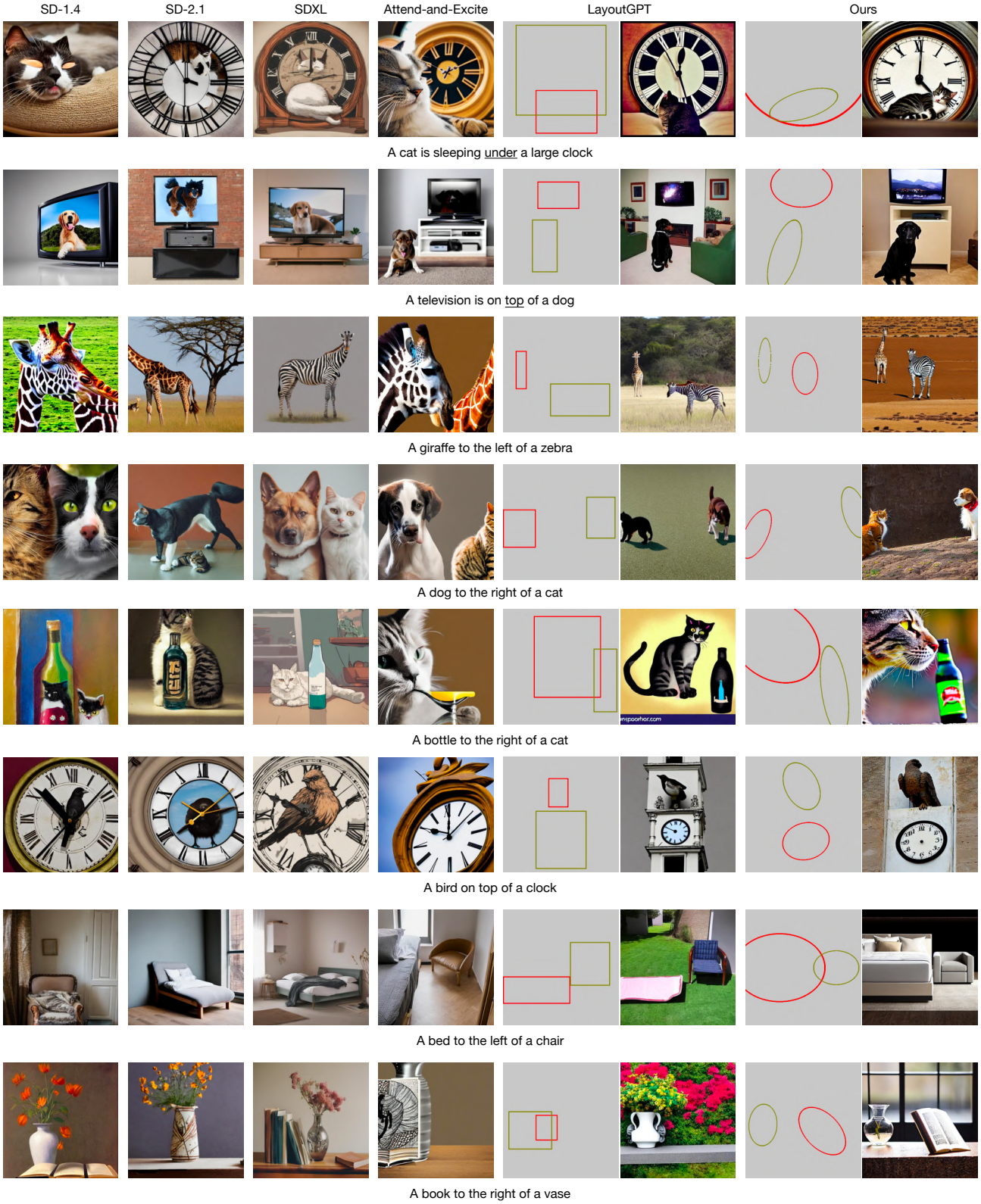


Figure 13. Qualitative results of various methods on spatial reasoning tasks. In our method, given a caption, we prompt GPT4 to generate blob parameters (Left) and LLAMA-13B to generate blob descriptions (not shown in the figure), which are passed to our blob-grounded text-to-image generative model to synthesize an image (Right).

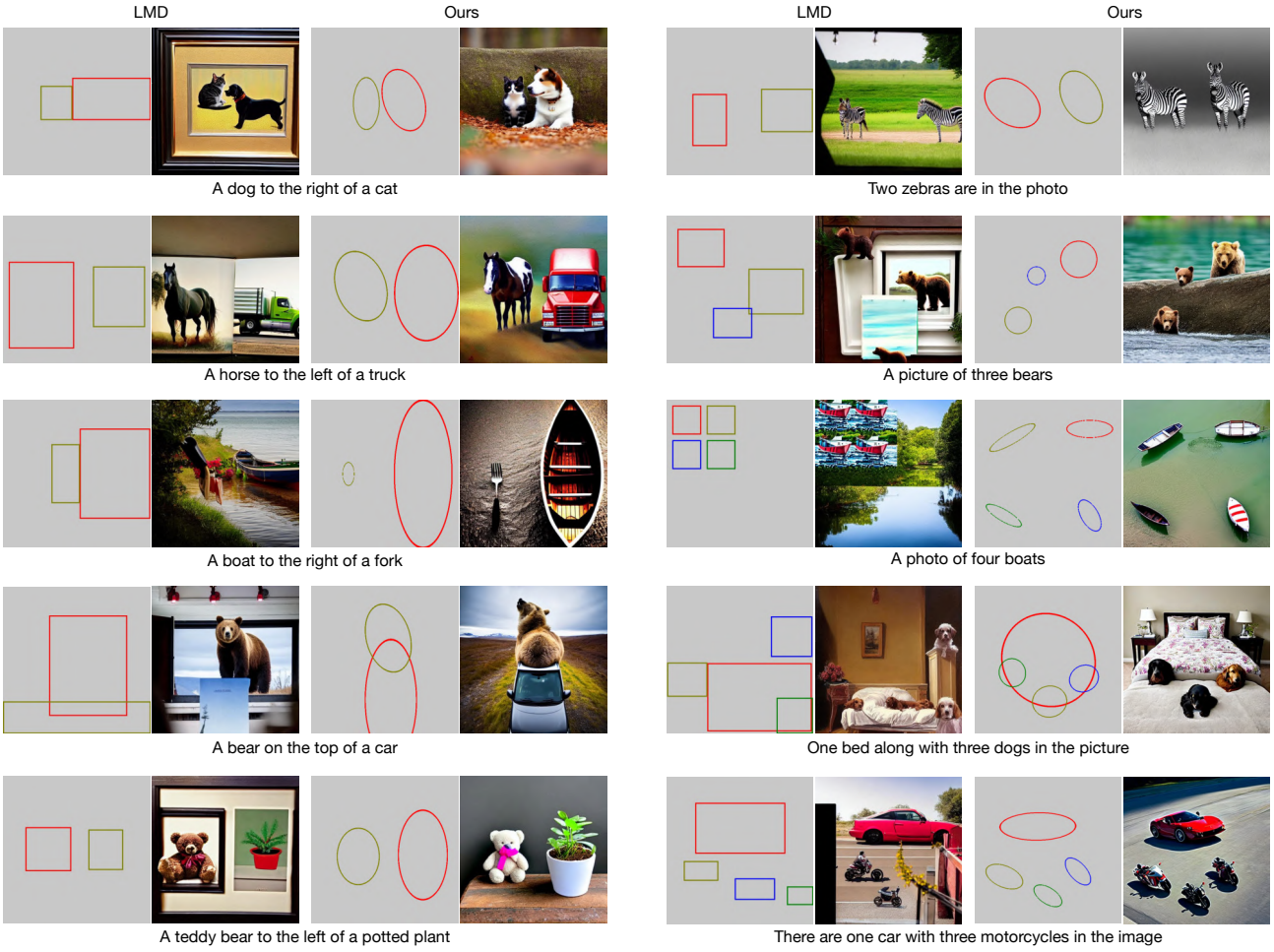
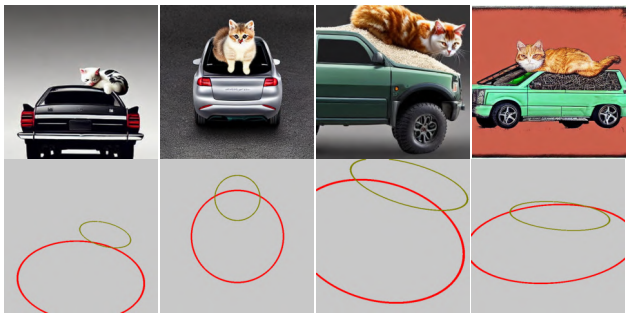
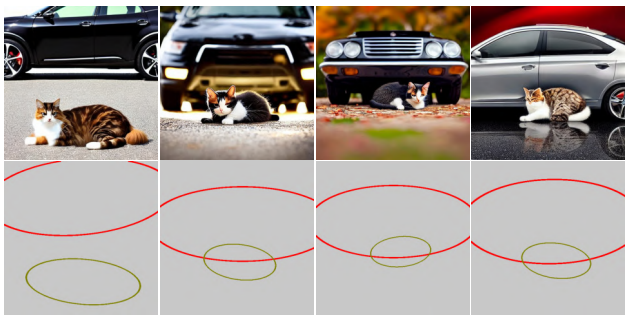


Figure 14. Qualitative results of comparing our method with LMD (Lian et al., 2023) on the NSR-1K benchmark for spatial and numerical reasoning. In each example, we first prompt GPT4 to generate boxes for LMD and blobs for our method, respectively, with the same 8 fixed in-context demo examples. The generated boxes and blobs are passed to LMD and BlobGEN to generate images, respectively. Better zoom in for better visualization.

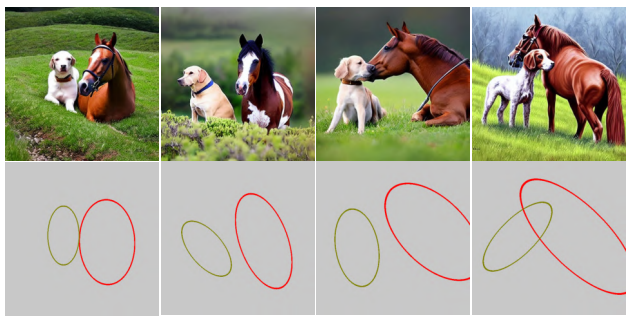


A car under a cat

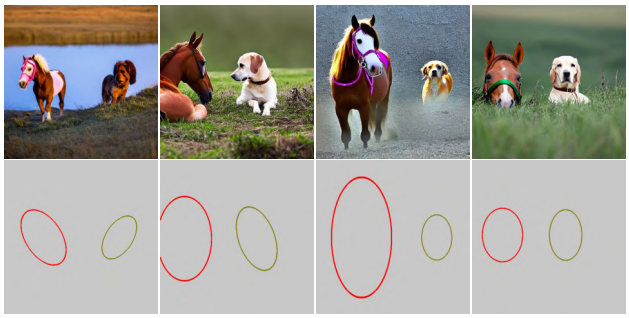


A cat under a car

(a) Swapping object name

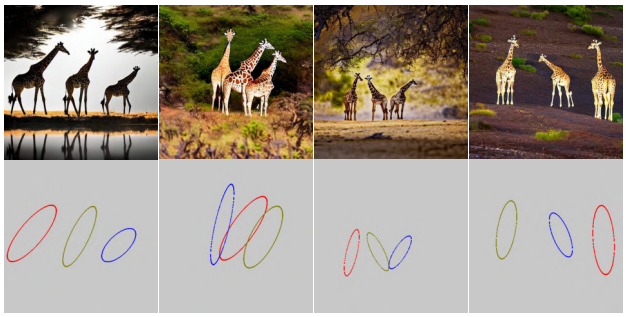


A horse to the right of a dog

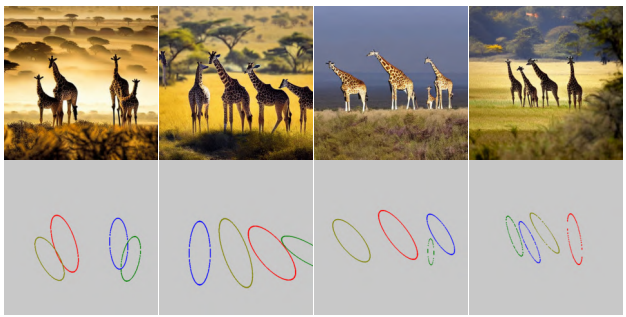


A horse to the left of a dog

(b) Changing relative reposition

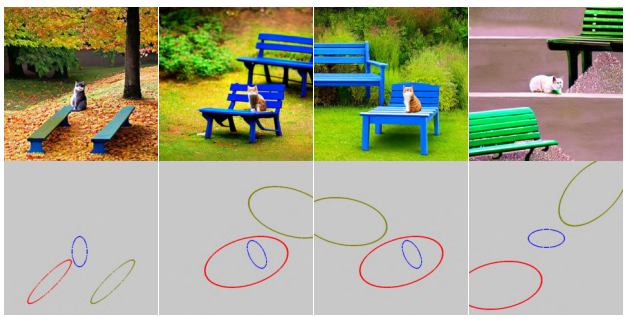


A photo of three giraffes

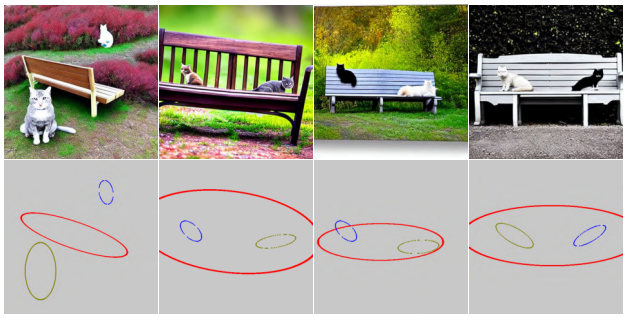


A photo of four giraffes

(c) Changing object number



A picture of two benches along with one cat



A picture of one bench along with two cats

(d) Swapping object number

Figure 15. Qualitative results of blob control over using LLMs, where we consider four cases of how LLMs understand compositional prompts for correct visual planning: (a) swapping object name ("cat" ↔ "car"), (b) changing relative reposition ("left" ↔ "right"), (c) changing object number ("three" ↔ "four"), and (d) swapping object number ("one bench & two cats" ↔ "two benches & one cat"). In each example, we show diverse blobs generated by LLMs (bottom) and the corresponding images generated by BlobGEN (top) from the same text prompt. Better zoom in for better visualization.

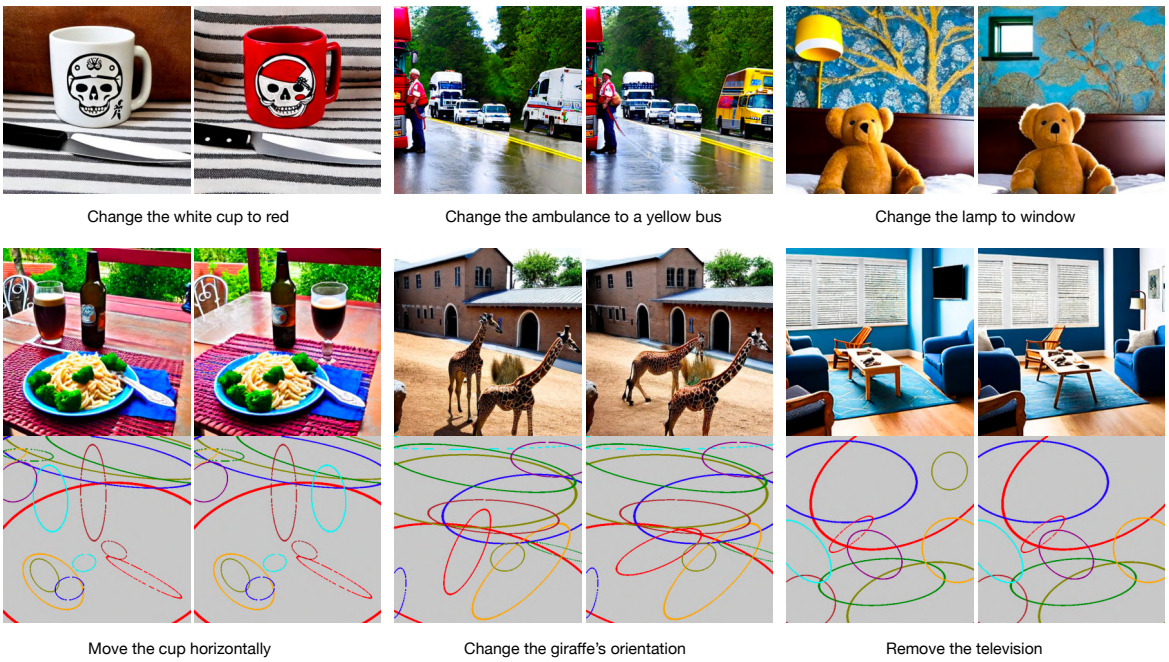


Figure 16. Some failure cases for image editing. Similarly, each example contains two generated images: (Left) original setting and (Right) after editing. The top row shows the local editing results where we only change the blob description and since the blob parameters stay the same after editing, we do not show blob visualizations. The bottom two rows show the object reposition results where we only change the blob parameter. All images are in resolution of 512×512.

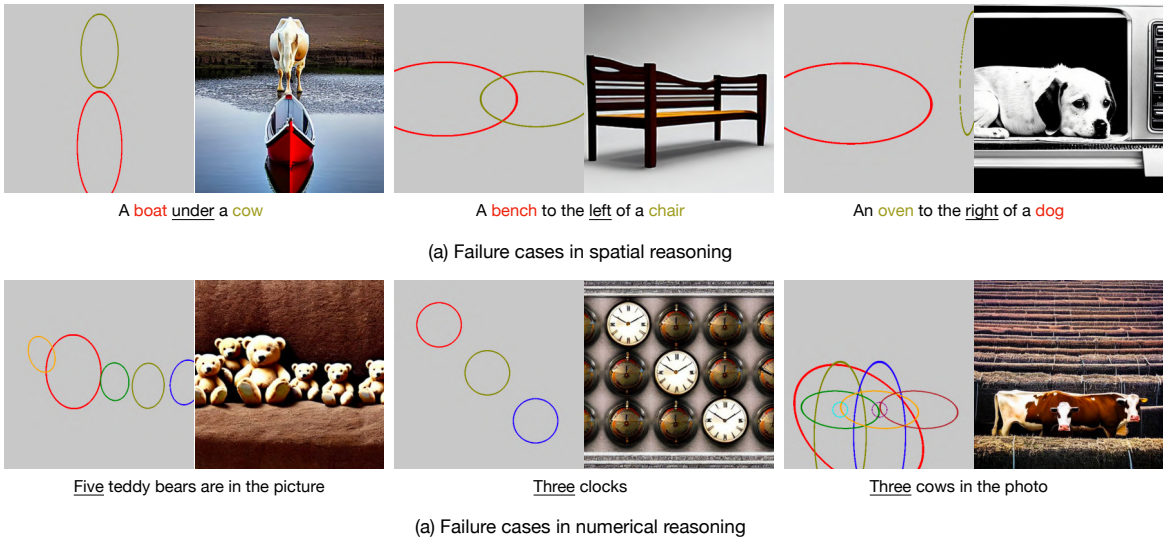


Figure 17. Some failure cases for spatial and numerical reasoning. Given a caption, we prompt GPT4 to generate blob parameters (Left) and LLAMA-13B to generate blob descriptions (not shown in the figure), which are passed to our blob-grounded text-to-image generative model to synthesize an image (Right). All images are in resolution of 512×512.