Laytrol: Preserving Pretrained Knowledge in Layout Control for Multimodal Diffusion Transformers

Sida Huang[1,2], **Siqi Huang**[1,2], **Ping Luo**[3], **Hongyuan Zhang**[2, 3*]
[1]School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University
[2]Institute of Artificial Intelligence (TeleAI), China Telecom
[3]The University of Hong Kong
{sidahuang2001, 4777huang}@gmail.com, pluo@cs.hku.hk, hyzhang98@gmail.com

Abstract

With the development of diffusion models, enhancing spatial controllability in text-to-image generation has become a vital challenge. As a representative task for addressing this challenge, layout-to-image generation aims to generate images that are spatially consistent with the given layout condition. Existing layout-to-image methods typically introduce the layout condition by integrating adapter modules into the base generative model. However, the generated images often exhibit low visual quality and stylistic inconsistency with the base model, indicating a loss of pretrained knowledge. To alleviate this issue, we construct the **Lay**out **Syn**thesis (LaySyn) dataset, which leverages images synthesized by the base model itself to mitigate the distribution shift from the pretraining data. Moreover, we propose the **Lay**out Con**trol** (Laytrol) Network, in which parameters are inherited from MM-DiT to preserve the pretrained knowledge of the base model. To effectively activate the copied parameters and avoid disturbance from unstable control conditions, we adopt a dedicated initialization scheme for Laytrol. In this scheme, the layout encoder is initialized as a pure text encoder to ensure that its output tokens remain within the data domain of MM-DiT. Meanwhile, the outputs of the layout control network are initialized to zero. In addition, we apply Object-level Rotary Position Embedding to the layout tokens to provide coarse positional information. Qualitative and quantitative experiments demonstrate the effectiveness of our method.

Code — https://github.com/HHHHStar/Laytrol

1 Introduction

Diffusion models have significantly promoted the development of text-to-image (T2I) generation. Among them, U-Net-based Stable Diffusion (Rombach et al. 2022) models have been widely used in the T2I community due to the efficiency and effectiveness. Recently, MM-DiT introduced the transformer architecture into diffusion models, leading to the emergence of more advanced T2I models such as Stable Diffusion 3 (Esser et al. 2024) and FLUX (Black-Forest-Labs 2024).

To enhance spatial controllability in T2I models, the task of layout-to-image generation is proposed, which aims to generate different objects within specified regions of the image. For this task, many adapter-based methods (Li et al. 2023; Zhou et al. 2024; Zhang et al. 2024a) have been proposed. These methods insert new adapter modules into the base generative model and train the model on datasets with layout annotations. However, we find that the images generated by adapter-based methods exhibit low visual quality and stylistic inconsistency with the base model, as illustrated in Figure 4. The factors contributing to this issue can be analyzed from two perspectives: the training dataset and the control module. On the one hand, previous training datasets are primarily based on COCO (Lin et al. 2014) or subsets of LAION (Schuhmann et al. 2022). Distribution shift exists between these datasets and the pretraining data of base generative models. On the other hand, control modules in existing methods are trained from scratch, which prevents them from effectively inheriting the pretrained knowledge of the base model. Therefore, ***we focus on effectively utilizing the pretrained knowledge of the base generative model to guide dataset construction and control module parameter initialization***.

Regarding dataset construction, we first generate images using base generative model FLUX (Black-Forest-Labs 2024), and then annotate their layouts using open-source models such as Grounding DINO (Liu et al. 2024). Images generated by the model itself can effectively retain the image style and high-quality details derived from the pretraining knowledge, thereby mitigating the distribution shift from the pretraining data. During this process, we observe that the model tends to produce images with repetitive layout patterns. To mitigate this layout bias, we propose layout prompting, which augments object description by randomly incorporating spatial and size-related phrases.

Regarding initialization of control module parameters, inspired by ControlNet (Zhang, Rao, and Agrawala 2023), we propose to incorporate copied parameters from MM-DiT into the corresponding layout control modules. Leveraging copied parameters requires satisfying two initialization conditions prior to training:

C1 The input to the layout control modules must lie within their own data domain.

C2 The output of the layout control modules must be initialized to zero.

*Corresponding author.

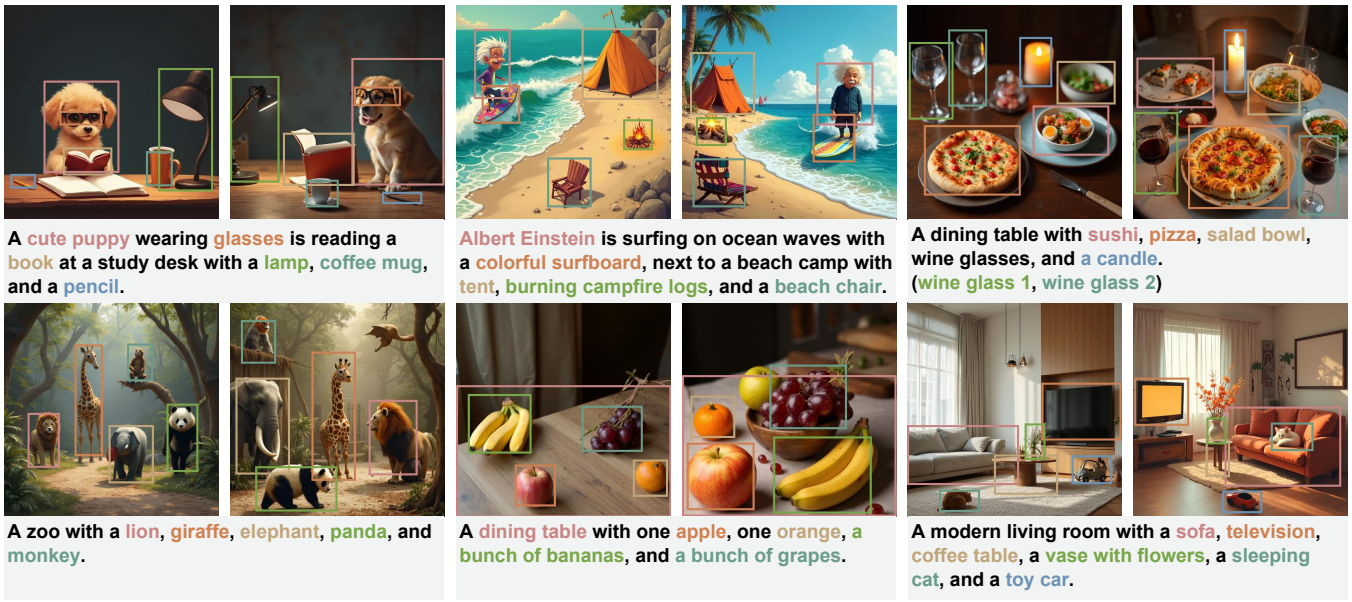

Figure 1: **Layout-to-image results of Laytrol**. The text in the same color as the bounding box corresponds to the local prompt. Laytrol enables layout-conditioned control in MM-DiT while effectively retaining the knowledge learned during the pre-training stage, and simultaneously mitigates the domain shift introduced by fine-tuning.

At the start of training, *C1* ensures that the copied parameters can be properly activated, and *C2* ensures that the base generative network is not disturbed by unstable control conditions. To satisfy *C1*, we initialize the layout encoder to be functionally equivalent to the text encoder, for which its output tokens lie within the domain of MM-DiT. In the following training process, spatial information is gradually injected into the layout encoder. To satisfy *C2*, we add a zero-initialized linear layer, ensuring that the initial output of the layout control modules is zero.

Overall, our contributions can be summarized as follows:

- We propose **Lay**out Con**trol** (Laytrol) Network, a layout-to-image generation method that preserves pretrained knowledge by leveraging parameter copying.
- We introduce **Lay**out **Syn**thesis (LaySyn) dataset, which leverages the base generative model FLUX for image synthesis to alleviate distribution shift. To mitigate layout bias in this process, we employ layout prompting that randomly injects layout phrases into object descriptions.
- We propose object-level Rotary Position Embedding (RoPE) to encoode coarse positional information for layout tokens.

2 Related Works

2.1 Text-to-Image Generation

Text-to-image(T2I) generation is the task of learning a conditional mapping from a natural-language description to a corresponding image. By leveraging noise (Li 2022; Zhang et al. 2025, 2024b; Huang, Zhang, and Li 2025; Huang et al. 2025a; Jiang et al. 2025), diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020) have rapidly advanced and and have been applied across various domains (Wang, Zhang, and Yuan 2025; Fu et al. 2025; Huang et al. 2025b). Open-source Stable Diffusion (Rombach et al. 2022; Podell et al. 2023) have achieved promising performance in T2I synthesis. However, the limited representation capability of CLIP text encoder and U-Net architecture restricts both semantic comprehension and image quality. Recent studies(Saharia et al. 2022; Esser et al. 2024), employ T5 (Raffel et al. 2020) as the text encoder to enhance the understanding ability of textual prompts. In addition, Diffusion Transformer (DiT) (Peebles and Xie 2023) introduces the transformer architecture into the image generation domain, exhibiting superior scalability. Compared to diffusion models, rectified flow model (Liu, Gong et al. 2023) achieves improved performance by learning straight paths between two distribution points. Leveraging these advancements, Stable Diffusion 3 (Esser et al. 2024) and FLUX (Black-Forest-Labs 2024) achieve state-of-the-art performance.

2.2 Layout-to-Image Generation

Since textual descriptions exhibit ambiguity in conveying spatial information, generating images that conform to specific spatial layouts has become increasingly important. Several training-free methods (Bar-Tal et al. 2023; Xie et al. 2023) constrain object positions by optimizing noisy latents guided by attention maps. RPG (Yang et al. 2024) utilizes a large language model (LLM) to design layouts, then generates each object individually and composes them in the latent space. These methods save training resources but typically require more inference steps. Moreover, they suffer from degraded image quality and lower layout fidelity. In contrast, training-based approaches such as GLIGEN (Li et al. 2023) encode bounding box coordinates us-

ing Fourier embeddings (Mildenhall et al. 2021) and control layouts through a newly trained cross-attention layer. Similarly, MIGC (Zhou et al. 2024) also propose dedicated layout control modules within the U-Net architecture. Building on DiT, SiamLayout (Zhang et al. 2024a) trains an MM-DiT-based control module for Stable Diffusion 3 (Esser et al. 2024) and FLUX (Black-Forest-Labs 2024). However, these models train the newly inserted modules from scratch, lacking the utilization of pretrained knowledge.

3 Method

3.1 Preliminaries

Multimodal Diffusion Transformer To introduce the architecture of transformers in image generation, Multimodal Diffusion Transformer (MM-DiT) is proposed to jointly process image tokens $\mathbf{X} \in \mathbb{R}^{n \times d}$ and text tokens $\mathbf{C_T} \in \mathbb{R}^{m \times d}$. Analogous to a standard transformer module, these tokens are projected through linear layers to obtain $\mathbf{Q_I}, \mathbf{K_I}, \mathbf{V_I}$ and $\mathbf{Q_T}, \mathbf{K_T}, \mathbf{V_T}$, respectively. Before computing attention, Rotary Position Embedding (RoPE) (Su et al. 2024) is applied to the query $\mathbf{Q}$ and key $\mathbf{K}$ to encode relative position information. For a visual token $\mathbf{X}^{i,j}$ located at position (i, j) in the 2D latent space, its corresponding query ${\mathbf{Q_I}}^{i,j}$ and key ${\mathbf{K_I}}^{i,j}$ are mapped as

$$\begin{aligned} {\mathbf{Q_I}}^{i,j} &= \mathbf{W_Q} \cdot \mathbf{X}^{i,j} \cdot \mathbf{R}(i,j); \\ {\mathbf{K_I}}^{i,j} &= \mathbf{W_K} \cdot \mathbf{X}^{i,j} \cdot \mathbf{R}(i,j), \end{aligned} \tag{1}$$

where $\mathbf{W_Q}$ and $\mathbf{W_K}$ are linear transformation matrices, $\mathbf{R}(i, j)$ denotes the rotation matrix. For text tokens, FLUX assigns a fixed position index of $(0, 0)$, as their positional information has already been encoded during text embedding. These tokens are then concatenated as $\mathbf{Q} = \mathbf{Q_I} \oplus \mathbf{Q_T}$, $\mathbf{K} = \mathbf{K_I} \oplus \mathbf{K_T}$ and $\mathbf{V} = \mathbf{V_I} \oplus \mathbf{V_T}$. The concatenated tokens from both modalities are subsequently involved in the scaled dot-product attention computation:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}. \tag{2}$$

During computation, the product of two rotation matrices $\mathbf{R}(i, j) \cdot \mathbf{R}(i', j')^\top$ depends solely on their relative position $(i - i', j - j')$.

Layout-to-Image Generation Layout-to-image generation aims to synthesize specified content for different regions of an image, thereby enhancing spatial controllability. The conditional input for this task consists of a global prompt p_g and a layout condition $\mathcal{L}$, which comprises N entities $\mathbf{e}$. Each entity $\mathbf{e}_i$ is associated with a local prompt p_i and a spatial position $\mathbf{b}_i$:

$$\mathcal{L} = \{\mathbf{e}_1, ..., \mathbf{e}_N\}, \quad \text{where } \mathbf{e}_i = (p_i, \mathbf{b}_i). \tag{3}$$

In the equation, $\mathbf{b}$ denotes the bounding box coordinates (x_1, y_1, x_2, y_2). To effectively represent the spatial positions of bounding boxes, GLIGEN (Li et al. 2023) encodes these coordinates using Fourier embeddings (Mildenhall et al. 2021). Finally, the Fourier embeddings and the encoded prompts are transformed by a multi-layer perceptron (MLP) to obtain the final layout tokens ${\mathbf{C_L}}^i \in \mathbb{R}^d$:

$${\mathbf{C_L}}^i = \text{MLP}(\text{CLIP}(p_i) \oplus \text{Fourier}(\mathbf{b}_i)). \tag{4}$$

3.2 Layout Control (Laytrol) Network

For U-Net-based diffusion models, Previous works (Li et al. 2023; Zhou et al. 2024) encode layout condition $\mathcal{L}$ using Eq. (4) and integrate it into the model via inserted cross-attention layers. For MM-DiT-based models, (Zhang et al. 2024a) adopt Eq. (4) to derive $\mathbf{C_L}$, which is then mapped to $\mathbf{Q_L}$, $\mathbf{K_L}$, $\mathbf{V_L}$ using newly introduced linear layers, enabling layout control through self-attention. However, the control modules introduced by existing methods are trained from scratch, limiting their ability to leverage the pretrained knowledge of the base generative model. Inspired by ControlNet (Zhang, Rao, and Agrawala 2023), we propose incorporating copied parameters into the layout control network to mitigate the above limitaition. This approach requires satisfying two initialization conditions *C1* and *C2* as mentioned in Section 1. In the following, we address *C1* in Layout Condition Encoding and *C2* in Layout Condition Integration.

Layout Condition Encoding The input to ControlNet is formulated as $\mathbf{X} + \mathbf{W^0} \times \mathbf{C_I}$, where $\mathbf{X}$ represents the input image tokens, $\mathbf{W^0}$ is a zero-initialized linear layer, and $\mathbf{C_I}$ denotes the condition image tokens, such as those derived from a depth map or Canny edge map. By employing a zero-initialized layer with additive fusion, the initial input remains equivalent to $\mathbf{X}$, thereby satisfying Condition *C1*. In contrast to image conditions, layout conditions consist of multiple local prompts p_i paired with their corresponding spatial positions $\mathbf{b}_i$, whose token structure differs significantly from that of $\mathbf{X}$. This **feature heterogeneity** prohibits direct addition between the input image tokens $\mathbf{X}$ and the layout condition tokens $\mathbf{C_L}$.

In MM-DiT, the input comprises both image tokens and text tokens. The image tokens $\mathbf{X}$ and the local prompt tokens p_i fall within the input domain of MM-DiT, whereas the spatial position tokens $\mathbf{b}_i$ lie outside the input domain. Consequently, initializing position tokens $\mathbf{b}_i$ to zero is a reasonable choice. Moreover, due to the intrinsic correspondence between p_i and $\mathbf{b}_i$, it is essential to perform feature fusion between them.

Based on the above considerations, the layout condition tokens ${\mathbf{C_L}}^i \in \mathbb{R}^{l \times d}$ for each entity $\mathbf{e}_i$ are encoded as

$${\mathbf{C_L}}^i = \text{T5}(p_i) + \mathbf{W^0} \times \text{MLP}(\text{Fourier}(\mathbf{b}_i)), \tag{5}$$

where T5 represents T5 text encoder (Raffel et al. 2020). Prior to the addition, the spatial position tokens are repeated l times to match the length of the local prompt tokens. All layout condition tokens ${\mathbf{C_L}}^i$ are then concatenated as

$$\mathbf{C_L} = \bigoplus_{i=1}^{N} {\mathbf{C_L}}^i \in \mathbb{R}^{(N \cdot l) \times d}. \tag{6}$$

The detailed encoding process is illustrated in Figure 2(b).

According Eq. 5, ${\mathbf{C_L}}^i$ is initially equal to $\text{T5}(p_i)$ at the beginning of training, indicating that ${\mathbf{C_L}}^i$ represents pure text tokens. Feeding $\mathbf{C_L}$ into the text branch of MM-DiT can effectively activates the copied parameters, thereby satisfying condition *C1*. During training, the linear layer $\mathbf{W^0}$

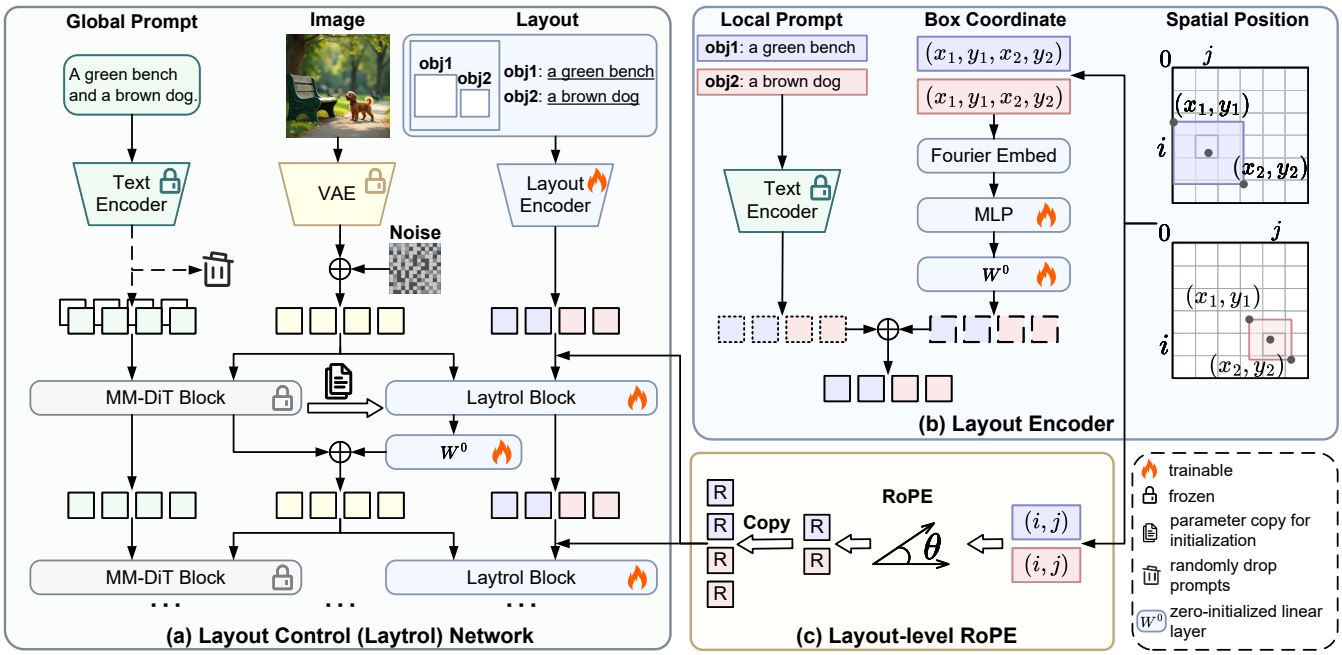

Figure 2: **Overview of Layout Control (Laytrol) pipeline**. **(a)** Laytrol blocks inherit vision-language pre-trained knowledge from DiT via parameter copying, which facilitates learning layout-conditioned control. Setting the global prompt tokens to null tokens encourages the model to focus more on the layout tokens during training. **(b)** The layout encoder is initialized as a pure text encoder with a zero-initialized projection layer. **(c)** The coordinates of the patch containing the bounding box center are used for applying RoPE to the layout tokens.

becomes non-zero and gradually injects spatial position features into $\mathbf{C_L}$. Consequently, $\mathbf{C_L}$ evolves from pure text tokens into layout-aware tokens that integrate both local prompts and spatial positional information.

Layout Condition Integration A standard MM-DiT computes attention over both image and text tokens as follows:

$$\mathbf{X_T}', \mathbf{C_T}' = \mathrm{DiT}(\mathbf{X}, \mathbf{C_T}; \Theta). \tag{7}$$

where Θ denotes the parameters of the DiT block.

As shown in Figure 2(a), Laytrol blocks share the same architecture as MM-DiT blocks. For each Laytrol block, its parameters Θ_c are initialized as a trainable copy of the corresponding MM-DiT parameters Θ. Given image tokens $\mathbf{X}$ and layout condition tokens $\mathbf{C_L}$ as input, Laytrol block outputs the corresponding transformed tokens:

$$\mathbf{X_L}', \mathbf{C_L}' = \mathrm{DiT}(\mathbf{X}, \mathbf{C_L}; \Theta_c). \tag{8}$$

In this process, $\mathbf{C_L}$ guides $\mathbf{X}$ to ensure compliance with the layout condition. During the fusion of $\mathbf{X_T}'$ and $\mathbf{X_L}'$, a zero-initialized linear layer $\mathbf{W}^0$ is added to satisfy condition *C2*:

$$\mathbf{X}' = \mathbf{X_T}' + \mathbf{W}^0 \times \mathbf{X_L}'. \tag{9}$$

This initialization scheme ensures that the Laytrol network has no effect on the base model at the start of training. Therefore, the model can generate a standard image based solely on the global prompt. Finally, $\mathbf{X}'$, $\mathbf{C_T}'$, $\mathbf{C_L}'$ are fed into the next layer of the network.

Layout-Level RoPE Before applying self-attention in MM-DiT, Rotary Position Embedding (RoPE) is added to the tokens as shown in Eq. 1. For an image token at position index (i, j) in the 2D latent space, the corresponding rotation matrix is denoted as $\mathbf{R}(i, j)$, while all text tokens share a common rotation matrix $\mathbf{R}(0, 0)$. To encode coarse spatial information for layout tokens, each layout token associated with a spatial position $\mathbf{b}_i$ is assigned a rotation matrix based on the position index of the patch containing its center coordinate $(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$. This process is illustrated in Figure2(c). This design encourages image tokens located near to the bounding box to attend more to the corresponding layout tokens during attention computation, thereby enhancing layout control in these regions.

3.3 Training Objective and Strategy

During training, the weights of the base generative model are frozen, while only the parameters of the layout encoder and layout control modules are updated. The model is optimized using the standard denoising diffusion loss:

$$\text{loss} = \mathbb{E}_{\mathbf{x}_0, t, p_g, \mathcal{L}, \boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, p_g, \mathcal{L})\|_2^2\right], \tag{10}$$

where $t \in [0, 1]$ denotes the time step, $\mathbf{x}_0$ is the original image and $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$ represents the noisy image. Following (Zhang et al. 2024a), we incorporate a region-aware loss that increases the loss weight within the bounding box regions by a factor of λ.

It is observed that the model tends to predict layout-related noise at higher timestep t, while at lower timestep

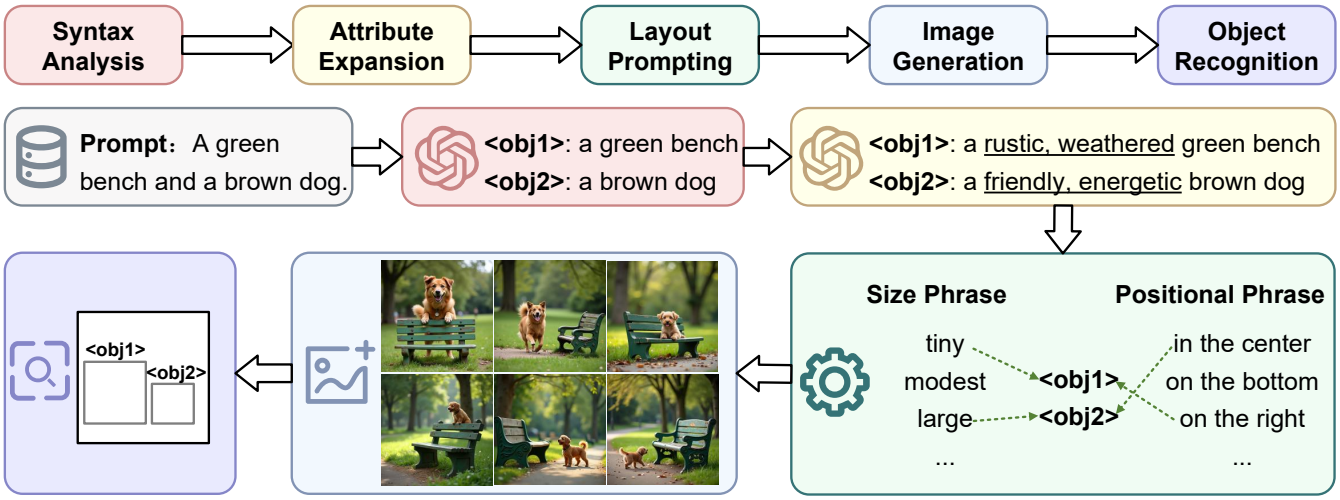

Figure 3: The pipeline for constructing the LaySyn dataset.

t the model focuses on noise associated with fine-grained image details. To emphasize layout-related information, a power-law distribution is employed to assign greater weight to higher t:

$$\pi(t;\alpha) = \alpha \cdot t^{\alpha-1}, t \in [0,1], \tag{11}$$

where we set $\alpha > 1$ in our experiments.

Random Prompt Dropping Modality competition exists among image tokens, global prompt tokens, and layout tokens (Zhang et al. 2024a). Due to extensive pretraining on image-text pairs, the model tends to rely predominantly on global prompt tokens during generation, resulting in relatively low attention from image tokens to layout tokens. To mitigate this issue, the global prompt tokens are replaced with null tokens with a probability p_d, thereby encouraging image tokens to attend more to the layout tokens.

3.4 LaySyn Dataset Construction

A suitable dataset for training layout-to-image models should annotate each image with a global prompt, entity bounding boxes, and corresponding local prompts. Previous works (Li et al. 2023; Zhou et al. 2024; Zhang et al. 2024a) adopt LAION (Schuhmann et al. 2022) or COCO (Lin et al. 2014) as training datasets. However, these datasets exhibit distribution shift relative to the pretraining data of models such as FLUX, leading to a degradation in image quality. To address this issue, we propose Layout Synthesis (LaySyn) dataset. Images are generated using FLUX and annotated with their corresponding layouts, with the goal of preserving the model's original generative capability. During image generation from prompts, we observe that the model often produces images with certain fixed layouts, resulting in limited layout diversity. We refer to this phenomenon as **Layout Bias**. To mitigate layout bias and promote a more uniform distribution of layouts, we propose layout prompting, as illustrated in Figure 3. By randomly incorporating phrases that describe position and size into the object-level prompts, the generated image layouts are enriched.

The LaySyn dataset construction pipeline consists of five stages, as illustrated in Figure 3. First, we select prompts containing multiple objects from the training sets of T2I-CompBench (Huang et al. 2023) and DiffusionDB (Wang et al. 2023). The objects in prompts are localized using syntax analysis performed by GPT-4o (Achiam et al. 2023). To enhance the richness of the prompts, adjectival attributes are added to the object descriptions. During the layout prompting stage, size-related (e.g., tiny, large) and positional (e.g., on the left, in the center) phrases are randomly added to object references. Subsequently, images are generated using FLUX, and object locations are annotated by Grounding DINO (Liu et al. 2024). Bounding boxes with high Intersection-over-Union (IoU) scores are suppressed to avoid redundancy. The final dataset contains approximately 400,000 images.

4 Experiments

4.1 Experimental Details

Training settings We train the model on two datasets: the proposed LaySyn dataset and COCO 2017 (Lin et al. 2014). FLUX.1-dev is employed as the base generative model. The training employs the AdamW optimizer with a cosine annealing learning rate schedule, an initial learning rate of 3e-5, and a weight decay of 0.01. A linear warm-up is applied during the first 1,000 iterations. The model is trained with a batch size of 64 for 500,000 iterations using DeepSpeed ZeRO Stage 3 on 8 NVIDIA H100 GPUs. We adopt bf16-mixed precision training (Wang and Kanwar 2019) to improve computational efficiency, and apply gradient checkpointing to reduce GPU memory consumption. The image resolution is 512×512. The region-aware loss weight λ is set to 2; the exponent α of the sampling distribution is 1.4; and the prompt dropping probability p_d is 0.5.

Evaluation settings We evaluate our model trained on LaySyn using T2I-Compbench (Huang et al. 2023). This benchmark assesses attribute binding, spatial relationships,

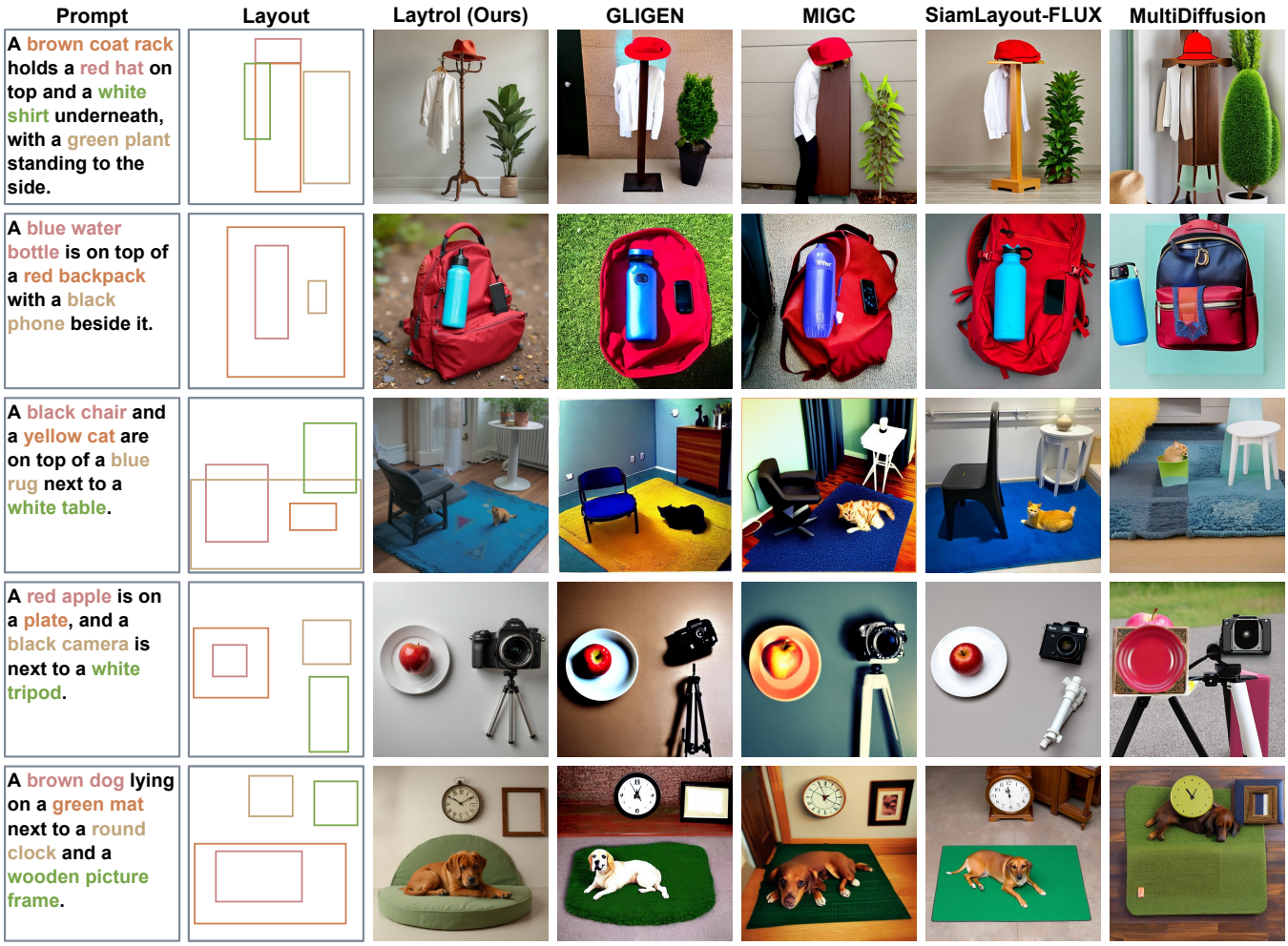

Figure 4: Qualitative comparison with other other methods. From the results, Laytrol shows better performance in terms of stylistic consistency, layout realism and object aesthetics.

and non-spatial relationships. Before image generation, we generate a layout from each prompt using GPT-4o, following a procedure similar to Section 3.4. Our proposed Laytrol is compared with pretrained text-to-image models including Stable Diffusion (Rombach et al. 2022) and FLUX (Black-Forest-Labs 2024), as well as layout-to-image models including GLIGEN (Li et al. 2023), MIGC (Zhou et al. 2024), MultiDiffusion (Bar-Tal et al. 2023) and SiamLayout (Zhang et al. 2024a). For models trained on COCO, we conduct evaluations using FID (Heusel et al. 2017), IS (Salimans et al. 2016), mIoU, AP and SSIM(Wang et al. 2004). Following prior work (Zhou et al. 2024), we employ Grounding-DINO (Liu et al. 2024) to detect each object and compute the maximum IoU between the detected boxes and the corresponding ground truth box. The baseline models used for comparison are the aforementioned layout-to-image methods.

4.2 Qualitative Results

As shown in Figure 4, the images generated by Laytrol exhibit higher stylistic consistency compared to those produced by GLIGEN and MIGC, which often resemble disjointedly objects and lack scene coherence. Moreover, Laytrol achieves more realistic layouts. For instance, in the first row, Laytrol generates a shirt hanging from a coat rack using a hanger, an element that is absent in other models and leads to more plausible interaction between objects. In the second row, the water bottle and phone are realistically positioned on the backpack, reflecting a strong adherence to physical plausibility. In addition, objects generated by Laytrol display superior aesthetic qualities in terms of color, texture and lighting, further validating its ability to preserve pretrained knowledge.

4.3 Quantitative Results

T2I-CompBench In Table 1, we evaluate Laytrol and other methods on T2I-CompBench, a comprehensive benchmark for open-world compositional image generation. For

Model	Attribute Binding			Object Relationship		Complex↑
	Color ↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑	
Stable Diffusion v1.4 (Rombach et al. 2022)	37.65	35.76	41.56	12.46	30.79	30.80
Stable Diffusion v2 (Rombach et al. 2022)	50.65	42.21	49.22	13.42	31.27	33.86
Stable Diffusion XL (Rombach et al. 2022)	58.79	46.87	52.99	21.33	31.19	32.37
FLUX.1 (Black-Forest-Labs 2024)	74.07	57.18	69.22	28.63	31.27	37.03
GLIGEN (Li et al. 2023)	34.00	34.70	49.16	33.22	30.39	27.96
MIGC (Zhou et al. 2024)	65.34	45.99	60.78	36.39	29.80	33.77
MultiDiffusion (Bar-Tal et al. 2023)	65.14	55.37	66.83	26.92	26.92	36.39
SiamLayout-FLUX (Zhang et al. 2024a)	76.63	52.21	65.35	35.84	**31.27**	36.78
Laytrol (Ours)	**80.65**	**57.70**	**70.69**	**47.40**	30.40	**40.36**

Table 1: Quantitative comparison on T2I-CompBench.

Model	FID ↓	IS ↑	mIoU ↑	AP ↑	SSIM ↑
GLIGEN	39.85	23.94	79.71	68.92	16.18
MIGC	39.25	26.58	77.64	65.11	20.43
MultiDiff	56.71	20.88	42.94	25.59	20.46
SiamLayout	36.66	**26.81**	70.09	56.62	26.66
Laytrol	**34.34**	26.39	**80.08**	**70.11**	**27.13**

Table 2: Quantitative comparison on COCO 2017 dataset.

Interval	FID ↓	IS ↑	mIoU ↑	AP ↑	SSIM ↑
1	35.38	26.33	76.75	64.11	27.91
2	37.56	24.73	73.54	59.21	27.60
4	39.22	24.66	72.37	57.63	27.32
6	42.48	23.31	72.16	57.44	27.10

Table 3: Quantitative experiments on the number of Laytrol blocks. Interval refers to the number of MM-DiT blocks controlled by each Laytrol block.

attribute binding, Laytrol achieves moderate improvement in color, texture, and shape. Regarding spatial relationship, Laytrol demonstrates the most significant improvement, indicating that the generated images closely follow the given layouts and accurately reflect the positional relationships among different objects.

COCO 2017 In Table 2, we evaluate Laytrol and other layout-to-image methods on COCO-2017 dataset. Laytrol achieves the best performance in terms of mIoU and AP metrics, demonstrating its superior spatial control capability. Meanwhile, Laytrol also has better results on FID and SSIM, indicating that the images generated by Laytrol exhibit higher overall quality.

4.4 Laytrol Block Scaling Analysis

To improve parameter efficiency and reduce inference cost, we scale down the number of Laytrol blocks by adjusting the interval, which specifies how many MM-DiT blocks are controlled by each Laytrol block. To ease the computational burden, we reduce the number of training iterations to 200,000 in these experiments. Detailed experimental results are presented in Table 3. Although reducing the number of control blocks leads to a slight decline in mIoU and AP, the layout accuracy remains reasonably preserved.

P-Copy	L-RoPE	P-Drop	FID ↓	mIoU ↑	AP ↑
✗	✗	✗	38.22	64.92	51.78
✓	✗	✗	35.65	75.68	63.31
✗	✓	✗	37.20	67.02	52.15
✗	✗	✓	36.70	71.28	56.71
✓	✓	✓	**35.38**	**76.75**	**64.11**

Table 4: Ablation on COCO 2017 dataset. P-Copy, L-RoPE, and P-Drop represent parameter copy, layout-level RoPE, and random prompt dropping, respectively.

4.5 Ablation Study

We present ablations on the components of Laytrol in Table 4. For all experiments, the number of training iterations is set to 200,000. The table demonstrates the effectiveness of the three components: parameter copy, layout-level RoPE, and random prompt dropping. In these components, parameter copy contributes the most significant improvement to the performance.

5 Conclusion

In this work, we propose Laytrol, a layout-to-image generation method that preserves pretrained knowledge through parameter copying. To ensure effective training, we design an initialization scheme that activates the copied parameters and maintains training stability. In addition, we introduce the LaySyn dataset to alleviate the domain shift from the pretraining data of the base generative model. Our work can be further improved by enhancing the diversity of the dataset distribution and integrating multiple types of control conditions.

References

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

Bar-Tal, O.; Yariv, L.; Lipman, Y.; and Dekel, T. 2023. Multidiffusion: Fusing diffusion paths for controlled image generation.

Black-Forest-Labs. 2024. FLUX. https://github.com/black-forest-labs/flux.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Fu, Y.; Si, R.; Wang, H.; Zhou, D.; Sun, J.; Luo, P.; Hu, D.; Zhang, H.; and Li, X. 2025. Object-AVEdit: An Object-level Audio-Visual Editing Model. arXiv:2510.00050.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

Huang, K.; Sun, K.; Xie, E.; Li, Z.; and Liu, X. 2023. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 78723–78747.

Huang, S.; Xu, Y.; Zhang, H.; and Li, X. 2025a. Learn Beneficial Noise as Graph Augmentation. arXiv:2505.19024.

Huang, S.; Zhang, H.; and Li, X. 2025. Enhance Vision-Language Alignment with Noise. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16): 17449–17457.

Huang, Z.; Qiu, X.; Ma, Y.; Zhou, Y.; Chen, J.; Zhang, H.; Zhang, C.; and Li, X. 2025b. NFIG: Multi-Scale Autoregressive Image Generation via Frequency Ordering. arXiv:2503.07076.

Jiang, K.; Shi, Z.; Zhang, D.; Zhang, H.; and Li, X. 2025. Mixture of Noise for Pre-Trained Model-Based Class-Incremental Learning. arXiv:2509.16738.

Li, X. 2022. Positive-Incentive Noise. *IEEE Transactions on Neural Networks and Learning Systems*, 1–7.

Li, Y.; Liu, H.; Wu, Q.; Mu, F.; Yang, J.; Gao, J.; Li, C.; and Lee, Y. J. 2023. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22511–22521.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, 740–755. Springer.

Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, 38–55. Springer.

Liu, X.; Gong, C.; et al. 2023. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *The Eleventh International Conference on Learning Representations*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205.

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

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

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. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X.; and Chen, X. 2016. Improved Techniques for Training GANs. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

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

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Wang, J.; Zhang, H.; and Yuan, Y. 2025. Adv-CPG: A Customized Portrait Generation Framework with Facial Adversarial Attacks. In *Proceedings of the IEEE/CVF Confer-*

ence on Computer Vision and Pattern Recognition (CVPR), 21001–21010.

Wang, S.; and Kanwar, P. 2019. BFloat16: The secret to high performance on Cloud TPUs. *Google Cloud Blog*, 4(1).

Wang, Z.; Bovik, A.; Sheikh, H.; and Simoncelli, E. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.

Wang, Z. J.; Montoya, E.; Munechika, D.; Yang, H.; Hoover, B.; and Chau, D. H. 2023. DiffusionDB: A Large-scale Prompt Gallery Dataset for Text-to-Image Generative Models. arXiv:2210.14896.

Xie, J.; Li, Y.; Huang, Y.; Liu, H.; Zhang, W.; Zheng, Y.; and Shou, M. Z. 2023. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7452–7461.

Yang, L.; Yu, Z.; Meng, C.; Xu, M.; Ermon, S.; and Cui, B. 2024. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *Forty-first International Conference on Machine Learning*.

Zhang, H.; Hong, D.; Gao, T.; Wang, Y.; Shao, J.; Wu, X.; Wu, Z.; and Jiang, Y.-G. 2024a. CreatiLayout: Siamese Multimodal Diffusion Transformer for Creative Layout-to-Image Generation. *arXiv preprint arXiv:2412.03859*.

Zhang, H.; Huang, S.; Guo, Y.; and Li, X. 2025. Variational Positive-Incentive Noise: How Noise Benefits Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(9): 8313–8320.

Zhang, H.; Xu, Y.; Huang, S.; and Li, X. 2024b. Data Augmentation of Contrastive Learning is Estimating Positive-incentive Noise. *arXiv preprint arXiv:2408.09929*.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3836–3847.

Zhou, D.; Li, Y.; Ma, F.; Zhang, X.; and Yang, Y. 2024. Migc: Multi-instance generation controller for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6818–6828.