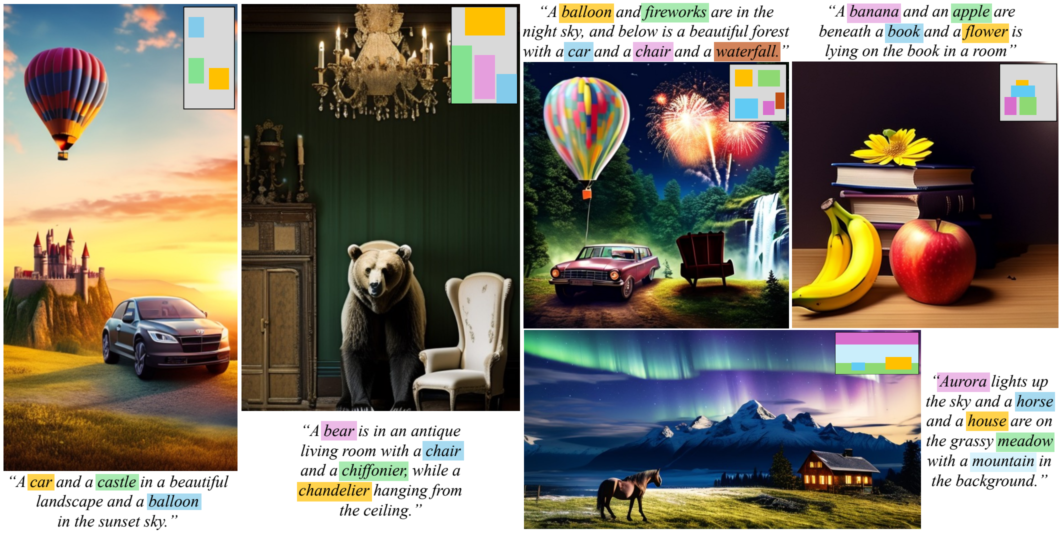
GrounDiT: Grounding Diffusion Transformers via Noisy Patch Transplantation

Phillip Y. Lee[*] **Taehoon Yoon**[*] **Minhyuk Sung**
KAIST
`{phillip0701,taehoon,mhsung}@kaist.ac.kr`



"A car and a castle in a beautiful landscape and a balloon in the sunset sky."

"A bear is in an antique living room with a chair and a chiffonier, while a chandelier hanging from the ceiling."

"A balloon and fireworks are in the night sky, and below is a beautiful forest with a car and a chair and a waterfall."

"A banana and an apple are beneath a book and a flower is lying on the book in a room"

"Aurora lights up the sky and a horse and a house are on the grassy meadow with a mountain in the background."

Figure 1: Spatially grounded images generated by our GrounDiT. Each image is generated based on a text prompt along with bounding boxes, which are displayed in the upper right corner of each image. Compared to existing methods that often struggle to accurately place objects within their designated bounding boxes, our GrounDiT enables more precise spatial control through a novel noisy patch transplantation mechanism.

Abstract

We introduce a novel training-free spatial grounding technique for text-to-image generation using Diffusion Transformers (DiT). Spatial grounding with bounding boxes has gained attention for its simplicity and versatility, allowing for enhanced user control in image generation. However, prior training-free approaches often rely on updating the noisy image during the reverse diffusion process via backpropagation from custom loss functions, which frequently struggle to provide precise control over individual bounding boxes. In this work, we leverage the flexibility of the Transformer architecture, demonstrating that DiT can generate noisy patches corresponding to each bounding box, fully encoding the target object and allowing for fine-grained control over each region. Our approach builds on an intriguing property of DiT, which we refer to as *semantic sharing*. Due to semantic sharing, when a smaller patch is jointly denoised alongside a generatable-size image, the two become "semantic clones". Each patch is denoised in its own branch of the generation process and then transplanted into the corresponding region of the original noisy image at each timestep, resulting in robust spatial grounding for each bounding box. In our experiments on the HRS and DrawBench benchmarks, we achieve state-of-the-art performance compared to previous training-free spatial grounding approaches. Project Page: `https://groundit-visualai.github.io/`.

[*] Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1 Introduction

The Transformer architecture [45] has driven breakthroughs across a wide range of applications, with diffusion models emerging as significant recent beneficiaries. Despite the success of diffusion models with U-Net [42] as the denoising backbone [22, 43, 41, 39], recent Transformer-based diffusion models, namely Diffusion Transformers (DiT) [37], have marked another leap in performance. This is demonstrated by recent state-of-the-art generative models such as Stable Diffusion 3 [13] and Sora [6]. Open-source models like DiT [37] and its text-guided successor PixArt-α [8] have also achieved superior quality compared to prior U-Net-based diffusion models. Given the scalability of Transformers, Diffusion Transformers are expected to become the new standard for image generation, especially when trained on an Internet-scale dataset.

With high quality image generation achieved, the next critical step is to enhance user controllability. Among the various types of user guidance in image generation, one of the most fundamental and significant is *spatial grounding*. For instance, a user may provide not only a text prompt describing the image but also a set of bounding boxes indicating the desired positions of each object, as shown in Fig. 1. Such spatial constraints can be integrated into text-to-image (T2I) diffusion models by adding extra modules that are designed for spatial grounding and fine-tuning the model. GLIGEN [31] is a notable example, which incorporates a gated self-attention module [1] into the U-Net layers of Stable Diffusion [41]. Although effective, such fine-tuning-based approaches incur substantial training costs each time a new T2I model is introduced.

Recent training-free approaches for spatially grounded image generation [9, 47, 11, 36, 38, 48, 12, 26] have led to new advances, removing the high costs for fine-tuning. These methods leverage the fact that cross-attention maps in T2I diffusion models convey rich structural information about where each concept from the text prompt is being generated in the image [7, 19]. Building on this, these approaches aim to align the cross-attention maps of specific objects with the given spatial constraints (*e.g.* bounding boxes), ensuring that the objects are placed within their designated regions. This alignment is typically achieved by updating the noisy image in the reverse diffusion process using backpropagation from custom loss functions. However, such loss-guided update methods often struggle to provide precise spatial control over individual bounding boxes, leading to missing objects (Fig. 4, Row 9, Col. 5) or discrepancies between objects and their bounding boxes (Fig. 4, Row 4, Col. 4). This highlights the need for finer control over each bounding box during image generation.

We aim to provide more precise spatial control over each bounding box, addressing the limitations in previous loss-guided update approaches. A well-known technique for manipulating local regions of the noisy image during the reverse diffusion process is to directly replace or merge the pixels (or latents) in those regions. This simple but effective approach has proven effective in various tasks, including compositional generation [17, 50, 44, 32] and high-resolution generation [5, 29, 24, 25]. One could consider defining an additional branch for each bounding box, denoising with the corresponding text prompt, and then copying the noisy image into its designated area in the main image at each timestep. However, a key challenge lies in creating a noisy image patch–at the same noise level–that *reliably* contains the desired object while fitting within the specified bounding box. This has been impractical with existing T2I diffusion models, as they are trained on a limited set of image resolutions. While recent models such as PixArt-α [8] support a wider range of image resolutions, they remain constrained by specific candidate sizes, particularly for smaller image patches. As a result, when these models are used to create a local image patch, they are often limited to denoising a fixed-size image and cropping the region to fit the bounding box. This approach can critically fail to include the desired object within the cropped region.

In this work, we show that by exploiting the flexibility of the Transformer architecture, Diffusion Transformers (DiT) can generate noisy image patches corresponding to the size of each bounding box, thereby reliably including the desired object. We first introduce an intriguing property of DiT: when simultaneously denoising a smaller noisy patch alongside a generatable-size noisy image, the two images gradually become *semantic clones*. We refer to this phenomenon as *shared sampling*. Building on this observation, we propose a training-free framework that involves *cultivating* a noisy patch for each bounding box in a separate branch and then *transplanting* that patch into the corresponding region of the original noisy image being generated in the main branch. By iteratively transplanting the separately denoised image patches into their respective bounding boxes, we achieved fine-grained spatial control over each bounding box. This approach leads to more robust spatial grounding, particularly in cases where previous methods fail to accurately adhere to spatial constraints.

In our experiments on the HRS [3] and DrawBench [43] datasets, we evaluate our framework, GROUNDIT, using PixArt-α [8] as the base text-to-image DiT model. Our approach demonstrates superior performance in spatial grounding compared to previous training-free methods [38, 9, 47, 48], especially outperforming the state-of-the-art approach [47], highlighting its effectiveness in providing fine-grained spatial control.

2 Related Work

In this section, we review the two primary approaches for incorporating spatial controls into text-to-image (T2I) diffusion models: fine-tuning-based methods (Sec. 2.1) and training-free guidance techniques (Sec. 2.2).

2.1 Spatial Grounding via Fine-Tuning

Fine-tuning with additional modules is a powerful approach for enhancing T2I models with spatial grounding capabilities [51, 31, 2, 53, 46, 16, 10, 54]. SpaText [2] introduces a spatio-textual representation that combines segmentations and CLIP embeddings [40]. ControlNet [51] incorporates a trainable U-Net encoder that processes spatial conditions such as depth maps, sketches, and human keypoints, guiding image generation within the main U-Net branch. GLIGEN [31] enables T2I models to accept bounding boxes by inserting a gated attention module into Stable Diffusion [41]. GLIGEN's strong spatial accuracy has led to its integration into follow-up spatial grounding methods [48, 38, 30] and applications such as compositional generation [15] and video editing [23]. InstanceDiffusion [46] further incorporates conditioning modules to provide finer spatial control through diverse conditions like boxes, scribbles, and points. While these fine-tuning methods are effective, they require task-specific datasets and involve substantial costs, as they must be retrained for each new T2I model, underscoring the need for training-free alternatives.

2.2 Spatial Grounding via Training-Free Guidance

In response to the inefficiencies of fine-tuning, training-free approaches have been introduced to incorporate spatial grounding into T2I diffusion models. One approach involves a region-wise composition of noisy patches, each conditioned on a different text input [5, 50, 32]. These patches, extracted using binary masks, are intended to generate the object they are conditioned on within the generated image. However, since existing T2I diffusion models are limited to a fixed set of image resolutions, each patch cannot be treated as a complete image, making it uncertain whether the extracted patch will contain the desired object. Another approach leverages the distinct roles of attention modules in T2I models—self-attention captures long-range interactions between image features, while cross-attention links image features with text embeddings. By using spatial constraints such as bounding boxes or segmentation masks, spatial grounding can be achieved either by updating the noisy image using backpropagation based on a loss calculated from cross-attention maps [48, 38, 9, 7, 18, 36], or by directly manipulating cross- or self-attention maps to follow the given spatial layouts [26, 4, 14]. While the loss-guided methods enable spatial grounding in a training-free manner, they still lack precise control over individual bounding boxes, often leading to missing objects or misalignmet between objects and their bounding boxes. In this work, we propose a novel training-free framework that offers fine-grained spatial control over each bounding box by harnessing the flexibility of the Transformer architecture in DiT.

3 Background: Diffusion Transformers

Diffusion Transformer (DiT) [37] represents a new class of diffusion models that utilize the Transformer architecture [45] for their denoising network. Previous diffusion models like Stable Diffusion [41] use a U-Net [42] architecture, of which each layer contains a convolutional block and attention modules. In contrast, DiT consists of a sequence of DiT blocks, each containing a pointwise feedforward network and attention modules, removing convolution operations and instead processing image tokens directly through attention mechanisms.

DiT follows the formulation of diffusion models [22], in which the forward process applies noise to a real clean data $\mathbf{x}_0$ by

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, I),\ \alpha_t \in [0,1]. \tag{1}$$

The reverse process denoises the noisy data $\mathbf{x}_t$ through a Gaussian transition

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{2}$$

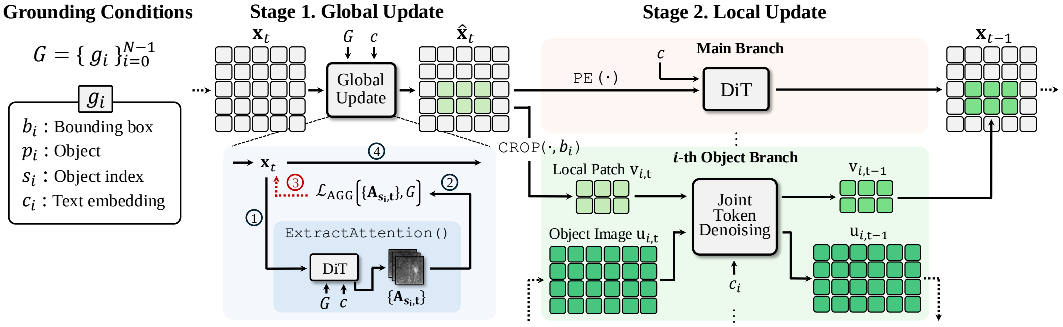


Figure 2: A single denoising step of GROUNDIT consists of two stages. Stage 1 (Sec. 5.1) performs Global Update, which updates the noisy image $\mathbf{x}_t$ using a custom loss function and obtains $\hat{\mathbf{x}}_t$. Stage 2 (Sec. 5.3) performs Local Update, providing fine-grained control over individual bounding boxes through a novel noisy patch cultivation-transplantation technique.

where $\mu_\theta(\mathbf{x}_t, t)$ is calculated by a learned neural network ϵ_θ trained by minimizing the negative ELBO objective [27]. While $\Sigma_\theta(\mathbf{x}_t, t)$ can also be learned, it is usually set as time dependent constants.

Positional Embeddings. As DiT is based on the Transformer architecture, the noisy image is treated as a set of image tokens. Suppose a noisy image $\mathbf{x}_t \in \mathbb{R}^{h \times w \times d}$. In DiT, $\mathbf{x}_t$ is first divided into patches, where each patch is transformed into an image token through a linear embedding. This results in a sequence of $(h/l) \times (w/l)$ image tokens, where l is the patch size. Importantly, before each denoising step, 2D sine-cosine positional embeddings are assigned to each image token, providing spatial information. This differs from U-Net diffusion models, which typically do not utilize positional embeddings for the noisy image.

Let $\texttt{PE}(\cdot)$ represent the application of positional embeddings. The set of image tokens, $\texttt{PE}(\mathbf{x}_t)$, which has been applied positional embeddings, is then passed through the DiT blocks during the denoising step as follows:

$$\mathbf{x}_{t-1} \leftarrow \texttt{Denoise}(\texttt{PE}(\mathbf{x}_t), t, c) \tag{3}$$

where $\texttt{Denoise}(\cdot)$ denotes a single denoising step of DiT at timestep t, and c is the text embedding. Detailed formulations of the positional embeddings are provided in the **Supplementary (Sec. A.1)**.

4 Problem Definition

This work aims to introduce a training-free framework that utilizes a text-to-image Diffusion Transformer [8, 37] to generate spatially accurate images based on bounding boxes. Let P be the input text prompt (*i.e.* list of tokens) for image generation – we refer to P as the *global prompt*. Let c_P be the text embedding of P. Consider a set of N grounding conditions $G = \{g_i\}_{i=0}^{N-1}$, where each condition specifies the coordinates of a bounding box and the desired object to be placed within it. Specifically, each g_i consists of (b_i, p_i, c_i, s_i), where the bounding box $b_i \in \mathbb{R}^4$ represents the x, y coordinates of its upper-left and lower-right corners, $p_i \in P$ is the word describing the desired object within the box, c_i is the corresponding text embedding, and s_i denotes the index of p_i in the global prompt P, such that $p_i = P[s_i]$. The objective is to generate an image that adheres to the global prompt P while ensuring each object is accurately positioned within its corresponding bounding box b_i.

5 GROUNDIT: Grounding Diffusion Transformers

We propose GROUNDIT, a training-free framework for spatially grounded image generation using a text-to-image Diffusion Transformer (DiT). GROUNDIT transforms each denoising step in the reverse diffusion process into a two-stage pipeline. In the first stage, Global Update, the noisy image $\mathbf{x}_t$ is refined via gradient descent, utilizing rich structural information encoded in the cross-attention maps, as commonly used in prior works for U-Net diffusion models [48, 47, 38, 9, 36, 7] (Sec. 5.1). To overcome the lack of precise local control for each grounding condition $g_i \in G$ in the first stage, we introduce the second stage, Local Update. Based on our observations on the semantic sharing property of DiT (Sec. 5.2), Local Update provides fine-grained control for each grounding condition through a novel noisy patch cultivation and transplantation process (Sec. 5.3). An overview of GROUNDIT's two-stage denoising step is shown in Fig. 2.

5.1 Stage 1: Global Update with Cross-Attention Maps

Let $\mathbf{x}_t \in \mathbb{R}^{h \times w \times d}$ be the noisy image at timestep t during the reverse diffusion process in DiT. As suggested by Chefer *et al.* [7], attention maps extracted from the cross-attention modules of a text-to-image diffusion model provide valuable structural information about which regions of $\mathbf{x}_t$ correspond to each text token in the input prompt P. To leverage this, the noisy image $\mathbf{x}_t$ is first passed into DiT, with positional embeddings applied as $\bar{\mathbf{x}}_t = \texttt{PE}(\mathbf{x}_t)$. Consider the DiT as being composed of M sequential blocks. As $\bar{\mathbf{x}}_t$ passes through the m-th block, the cross-attention map $a^m_{s_i,t} \in \mathbb{R}^{h \times w \times 1}$, corresponding to the object p_i from the grounding condition g_i is extracted. Recall that s_i denotes the index of word p_i in P. For each grounding condition g_i, the *mean* cross-attention map $A_{s_i,t}$ is then obtained by averaging the attention maps over all M blocks:

$$A_{s_i,t} = \frac{1}{M} \sum_{m=0}^{M-1} a^m_{s_i,t}. \tag{4}$$

Here we define $\texttt{ExtractAttention}(\cdot)$, which refers to the process of extracting all cross-attention maps corresponding to $g_i \in G$ as $\mathbf{x}_t$ passes through the DiT denoising step. This is formalized as:

$$\{A_{s_i,t}\}_{i=0}^{N-1} \leftarrow \texttt{ExtractAttention}(\mathbf{x}_t, t, c_P, G). \tag{5}$$

Following prior works on U-Net diffusion models [48, 47, 38, 9, 36, 7], we evaluate the spatial alignment between the mean cross-attention map $A_{s_i,t}$ for object p_i and its dedicated bounding box b_i using a predefined grounding loss $\mathcal{L}(A_{s_i,t}, b_i)$. For the grounding loss, we adopt the definition proposed in R&B [47]. The aggregated grounding loss $\mathcal{L}_{\text{AGG}}$ is then computed by summing the grounding loss across all grounding condition $g_i \in G$:

$$\mathcal{L}_{\text{AGG}}(\{A_{s_i,t}\}_{i=0}^{N-1}, G) = \sum_{i=0}^{N-1} \mathcal{L}(A_{s_i,t}, b_i). \tag{6}$$

Finally, based on the backpropagation from $\mathcal{L}_{\text{AGG}}$, the input noisy image $\mathbf{x}_t$ is updated via gradient descent as below:

$$\hat{\mathbf{x}}_t \leftarrow \mathbf{x}_t - \omega_t \nabla_{\mathbf{x}_t} \mathcal{L}_{\text{AGG}} \tag{7}$$

where ω_t is a scalar weight value for gradient descent. We refer to Eq. 7 as the *Global Update*, since the entire image $\mathbf{x}_t$ is updated based on an aggregated loss from all grounding conditions in G.

The Global Update in Eq. 7 achieves reasonable accuracy in spatial grounding with respect to the bounding boxes from G. However, we find that Global Update alone often struggles under more complex grounding conditions. For instance, when G contains multiple bounding boxes (*e.g.* five in Fig. 4, Row 9) or small, thin boxes (Fig. 4, Row 5), the desired objects may either be missing or misaligned with their bounding boxes. As seen in these examples, Global Update lacks fine-grained control over individual bounding boxes, underscoring the need for precise control tailored to each grounding condition g_i.

5.2 Semantic Sharing in Diffusion Transformers

Here we present our observations on an intriguing property of DiT [37, 8]—*semantic sharing*—that can be adopted for fine-grained local control designed for each grounding condition $g_i \in G$.

Convolution-Free Property of DiT. Recall that DiT does not include convolutional operations, in contrast to U-Net diffusion models. During the denoising steps of DiT, the noisy image $\mathbf{x}_t \in \mathbb{R}^{h \times w \times d}$ is treated as a set of image tokens. As a result, the positional information of an image token becomes solely dependent on the positional embedding which is assigned prior to every denoising step (Eq. 3).

Joint Token Denoising. Taking advantage of the convolution-free property of DiT and the role of positional embeddings in assigning spatial information to each image token, we introduce *joint token denoising*. Consider two different noisy images, $\mathbf{x}_t$ and $\mathbf{y}_t$, both at timestep t in the reverse diffusion process. Joint token denoising between $\mathbf{x}_t$ and $\mathbf{y}_t$ is illustrated in Fig. 3-(A). First, $\mathbf{x}_t$ and $\mathbf{y}_t$ are each assigned positional embeddings based on their respective sizes, resulting in $\bar{\mathbf{x}}_t = \texttt{PE}(\mathbf{x}_t)$ and $\bar{\mathbf{y}}_t = \texttt{PE}(\mathbf{y}_t)$. This allows DiT to treat both $\bar{\mathbf{x}}_t$ and $\bar{\mathbf{y}}_t$ as a single complete image. It is important to note that the sizes of the two noisy images do not need to be the same. The key aspect of joint token

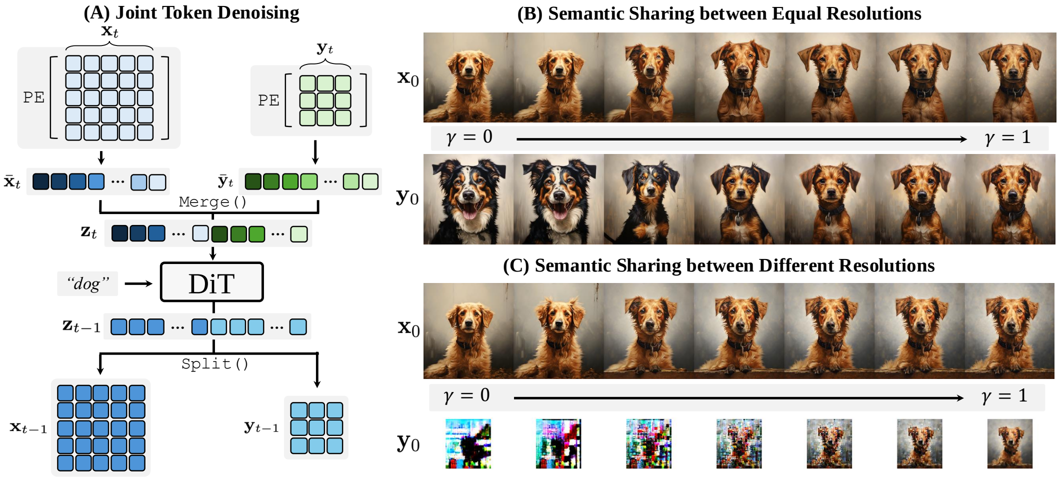


Figure 3: **(A) Joint Token Denoising (Alg. 1).** Two different noisy images, $\mathbf{x}_t$ and $\mathbf{y}_t$, are each assigned positional embeddings based on their respective sizes. The two sets of image tokens are then merged and passed through DiT for a denoising step. Afterward, the denoised tokens are split back into $\mathbf{x}_{t-1}$ and $\mathbf{y}_{t-1}$. **(B), (C) Semantic Sharing.** Denoising two noisy images using joint token denoising results in semantically correlated content between the generated images. Here, γ indicates that joint token denoising is during the initial $100\gamma\%$ of the timesteps, after which the images are denoised for the remaining steps.

denoising is that the two noisy images—or two sets of image tokens—are merged into a single set $\mathbf{z}_t$ via $\texttt{Merge}(\cdot)$ (Alg. 1, line 4). $\mathbf{z}_t$ is then passed through DiT for denoising. After the denoising step, DiT returns the denoised output $\mathbf{z}_{t-1}$, which is then split back into the original sets of image tokens via $\texttt{Split}(\cdot)$, resulting in the denoised versions $\mathbf{x}_{t-1}$ and $\mathbf{y}_{t-1}$. The full algorithm of joint token denoising is shown in Alg. 1.

Semantic Sharing. Surprisingly, we found that joint token denoising of two noisy images leads to semantically correlated content being generated in their corresponding pixels, even when the initial random noise is different. We demonstrate this through a simple experiment. Consider two noisy images, $\mathbf{x}_T \in \mathbb{R}^{h_\mathbf{x} \times w_\mathbf{x} \times d}$ and $\mathbf{y}_T \in \mathbb{R}^{h_\mathbf{y} \times w_\mathbf{y} \times d}$, both initialized from unit Gaussian distribution $\mathcal{N}(0, I)$. We define a reverse diffusion process in which, for the first $100\gamma\%$ ($\gamma \in [0, 1]$) of the denoising steps, $\mathbf{x}_t$ and $\mathbf{y}_t$ are denoised together using joint token denoising. For the remaining $100(1-\gamma)\%$, the two noisy images are denoised independently. The same text embedding c is used as a condition for both cases.

Fig. 3 shows the generated images from $\mathbf{x}_T$ and $\mathbf{y}_T$ across different γ values. Fig. 3-(B) illustrates a case where $\mathbf{x}_T$ and $\mathbf{y}_T$ have the same resolution ($h_\mathbf{x} = h_\mathbf{y}, w_\mathbf{x} = w_\mathbf{y}$), while in Fig. 3-(C) their resolutions differ ($h_\mathbf{x} > h_\mathbf{y}, w_\mathbf{x} > w_\mathbf{y}$). When $\gamma = 0$, the two noisy images are denoised completely independently, leading to clearly distinct generated images (leftmost columns). Notably, in Fig. 3-(C), when the resolution of $\mathbf{y}_T$ is set to be much smaller than DiT's generatable resolution, the output is implausible. However, as γ increases, allowing $\mathbf{x}_T$ and $\mathbf{y}_T$ to be denoised jointly through joint token denoising in the initial steps, the generated images become increasingly similar. When $\gamma = 1$, the images generated from $\mathbf{x}_T$ and $\mathbf{y}_T$ appear almost identical. This pattern holds not only when both noisy images have the same resolution (Fig. 3-(B)), but even when one images is not of DiT's generatable resolution (Fig. 3-(C)).

These results demonstrate that the positional embeddings assigned to each image token play a crucial role in shaping the content generated within the token. Assigning identical or similar positional embeddings to different image tokens promotes strong interactions between them during the self-attention in DiT, which subsequently affects the cross-attention. This correlated behavior during joint token denoising causes the two image tokens to converge toward semantically similar outputs, a phenomenon we refer to as *semantic sharing*. While self-attention sharing techniques have been explored in U-Net diffusion models to enhance style consistency between images [20, 34], they have been limited to images of equal resolution. By taking advantage of the flexibility to assign positional embeddings across different resolutions, our joint token denoising approach extends across

Algorithm 1: Pseudocode of Joint Token Denoising (Sec. 5.2).

```
Inputs: $\mathbf{x}_t \in \mathbb{R}^{h_\mathbf{x} \times w_\mathbf{x} \times d}, \mathbf{y}_t \in \mathbb{R}^{h_\mathbf{y} \times w_\mathbf{y} \times d}, t, c, l$; // Noisy images, timestep, text embedding,
        patch size.
Outputs: $\mathbf{x}_{t-1}, \mathbf{y}_{t-1}$;                         // Noisy images at timestep $t-1$.
Function JointTokenDenoise($\mathbf{x}_t, \mathbf{y}_t, t, c$):
    $n_\mathbf{x} \leftarrow h_\mathbf{x} w_\mathbf{x} / l^2,\ n_\mathbf{y} \leftarrow h_\mathbf{y} w_\mathbf{y} / l^2$;     // Store the number of image tokens.
    $\bar{\mathbf{x}}_t \leftarrow$ PE($\mathbf{x}_t$), $\bar{\mathbf{y}}_t \leftarrow$ PE($\mathbf{y}_t$);          // Apply positional embeddings.
    $\mathbf{z}_t \leftarrow$ Merge($\bar{\mathbf{x}}_t, \bar{\mathbf{y}}_t$);                  // Merge two sets of image tokens.
    $\mathbf{z}_{t-1} \leftarrow$ Denoise($\mathbf{z}_t, t, c$);                       // Denoising step with DiT.
    $\{\mathbf{x}_{t-1}, \mathbf{y}_{t-1}\} \leftarrow$ Split($\mathbf{z}_{t-1}, \{n_\mathbf{x}, n_\mathbf{y}\}$);     // Split back into two sets.
    return $\mathbf{x}_{t-1}, \mathbf{y}_{t-1}$;
```

heterogeneous resolutions, offering greater versatility. We provide further discussions and analysis on the generatable resolution of DiT and semantic sharing in the **Supplemetary (Sec. A.4)**.

5.3 Stage 2: Local Update with Semantic Sharing

Building on the semantic sharing property of DiT discussed in Sec. 5.2, we introduce the second stage of GrounDiT's denoising pipeline. Recall that the Global Update in Sec. 5.1 alone is insufficient for archieving fine-grained control for each specific grounding condition $g_i \in G$. To address this, we propose a novel noisy patch cultivation-transplantation mechanism, illustrated in Fig. 2, to enhance precise spatial control over each bounding box.

Main/Object Branch. Starting from the noisy image $\hat{\mathbf{x}}_t$ obtained after Global Update, we define multiple denoising branches which are designed to perform denoising in parallel. Specifically, a single *main branch* simply denoises $\hat{\mathbf{x}}_t$ with the global prompt P such that $\tilde{\mathbf{x}}_{t-1} \leftarrow \texttt{Denoise}(\hat{\mathbf{x}}_t, t, c)$. $\tilde{\mathbf{x}}_{t-1}$ is later utilized as a template for the patch transplanation to obtain the final denoised output $\mathbf{x}_{t-1}$. In addition, we set N *object branches* where the i-th branch is designed to provide fine-grained local control for the region in corresponding to the grounding condition g_i using joint token denoising. The following section provides details on the denoising process of object branch.

Noisy Patch Cultivation. Consider the i-th object branch at timestep t of the reverse diffusion process. The inputs for this branch are twofold. First, we define a *noisy object image* $\mathbf{u}_{i,t} \in \mathbb{R}^{h_{\mathbf{u}_i} \times w_{\mathbf{u}_i} \times d}$ which is initialized as $\mathbf{u}_{i,T} \sim N(0, I)$. The role of $\mathbf{u}_{i,t}$ is to cultivate noisy image tokens that contain rich semantic information of the desired object p_i from the grounding condition g_i. Second, we take the subset of image tokens from $\hat{\mathbf{x}}_t$ which are within the bounding box region specified in b_i. This operation can be expressed as $\mathbf{v}_{i,t} \leftarrow \texttt{Crop}(\hat{\mathbf{x}}_t, b_i)$, and we call $\mathbf{v}_{i,t} \in \mathbb{R}^{h_{\mathbf{v}_i} \times w_{\mathbf{v}_i} \times d}$ a *noisy local patch*. The main objective of noisy patch cultivation is to convey the semantic information of p_i encoded in $\mathbf{u}_{i,t}$ into the local patch $\mathbf{v}_{i,t}$. This is realized by applying our joint token denoising (Alg. 1) with $\mathbf{u}_{i,t}$ and $\mathbf{v}_{i,t}$ to obtain the denoised versions $\mathbf{u}_{i,t-1}$ and $\mathbf{v}_{i,t-1}$ as follows:

$$\{\mathbf{u}_{i,t-1}, \mathbf{v}_{i,t-1}\} \leftarrow \texttt{JointTokenDenoise}(\mathbf{u}_{i,t}, \mathbf{v}_{i,t}, t, c_i) \tag{8}$$

where c_i is the text embedding of the object p_i.

Benefiting from the semantic sharing with the noisy object image $\mathbf{u}_{i,t}$ during the joint token denoising, the denoised local patch $\mathbf{v}_{i,t-1}$ is expected to have richer semantic features of the object p_i compared to the previous timestep. The significance of this approach is that even if the noisy local patch $\mathbf{v}_{i,t}$ is not of a usual generatable resolution by DiT (which is mostly the case as we crop the small bounding box regions of $\hat{\mathbf{x}}_t$ to obtain $\mathbf{v}_{i,t}$), it offers a simple and effective way for making $\mathbf{v}_{i,t}$ to be richer in terms of the information of object p_i. We refer to this process as *noisy patch cultivation*.

Noisy Patch Transplantation. After obtaining the local patch $\mathbf{v}_{i,t-1}$ as described in Eq. 8, the image tokens are injected back into their original region, specified by b_i, within $\tilde{\mathbf{x}}_t$ from the main branch as:

$$\tilde{\mathbf{x}}_{t-1} \leftarrow \tilde{\mathbf{x}}_{t-1} \odot (1 - m_i) + \texttt{Uncrop}(\mathbf{v}_{i,t-1}, b_i) \odot m_i \tag{9}$$

where $\odot$ denotes the Hadamard product, m_i is a binary mask corresponding to the bounding box b_i, and $\texttt{Uncrop}(\mathbf{v}, b)$ zero-pads $\mathbf{v}$ so that it aligns with the b_i region in the output. This injection provides fine-grained local control for the grounding condition $g_i \in G$. Once the ouptuts from all N

object branches are injected, we obtain $\mathbf{x}_{t-1}$, which represents the final output of the GrounDiT denoising step at timestep t. In $\mathbf{x}_{t-1}$, the image tokens within the region corresponding to b_i are expected to possess richer semantic information about the object p_i when compared to the initial $\tilde{\mathbf{x}}_t$ obtained from the Main Branch, owing to the semantic sharing during joint token denoising in Eq. 8. This process is referred to as *noisy patch transplantation*. We provide implementation details and full pseudocode of GrounDiT single denoising step in the **Supplementary (Sec. A.5).**

6 Results

In this section, we present the experiment results of our method, GrounDiT, and provide comparisons with baselines. For the base text-to-image DiT model, we use PixArt-α [8], which builds on the original DiT architecture [37] by incorporating an additional cross-attention module to condition on text prompts.

6.1 Evaluation Settings

Baselines. We compare our method with state-of-the-art training-free approaches for bounding box-based image generation, including R&B [47], BoxDiff [48], Attention-Refocusing [38], and Layout-Guidance [14]. For a fair comparison, we also implement R&B using PixArt-α, which we refer to as *PixArt-R&B*, and treat it as an internal baseline. Note that this is identical to our method without the Local Guidance (Sec. 5.3).

Evaluation Metrics and Benchmarks.

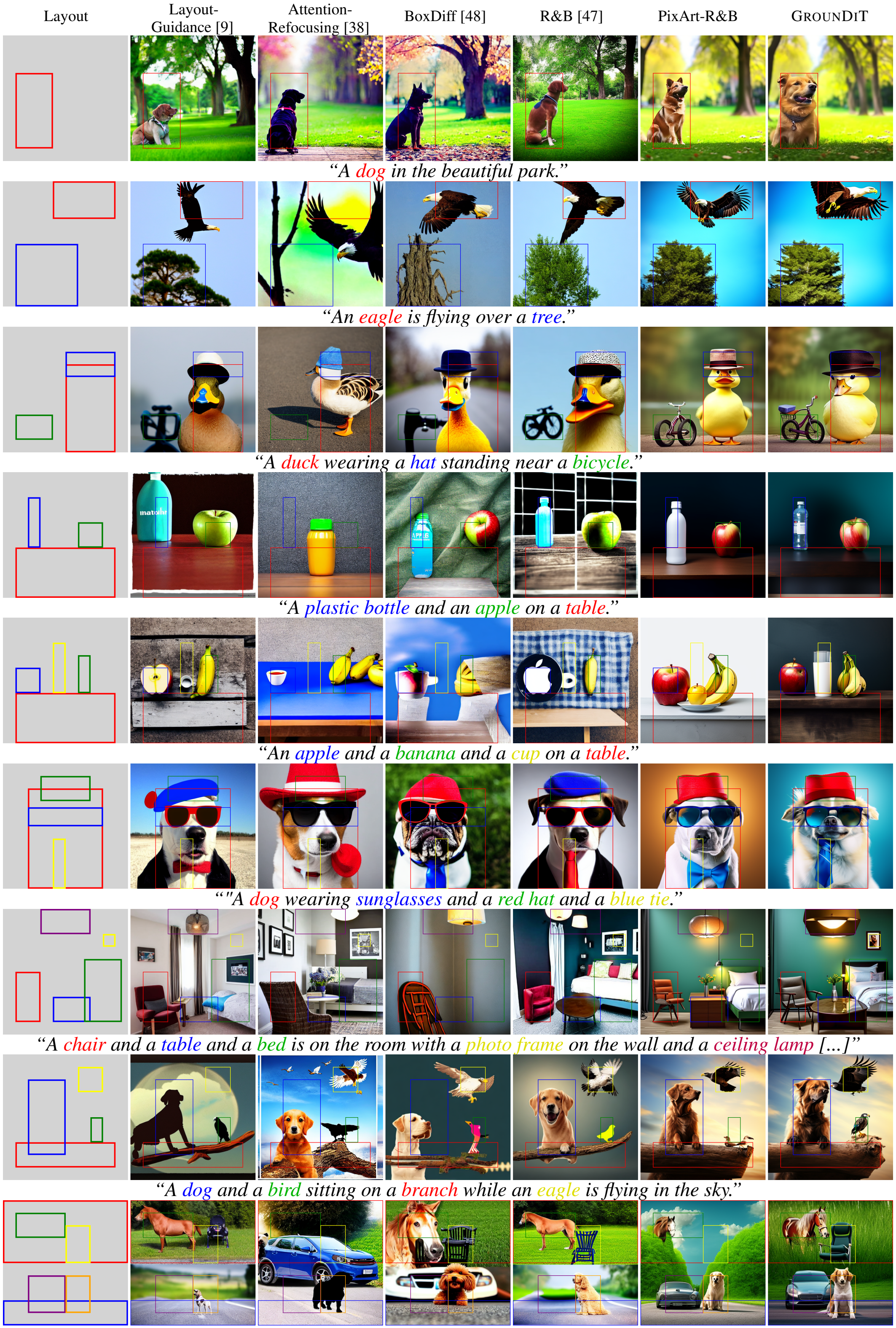
- **(Grounding Accuracy)** We follow the evaluation protocol of R&B [47] to assess spatial grounding on the HRS [3] and DrawBench [43] datasets, using three criteria: spatial, size, and color. The HRS dataset consists of 1002, 501, and 501 images for each respective criterion, with bounding boxes generated using GPT-4 by Phung *et al.* [38]. For DrawBench, we use the same 20 positional prompts as in R&B [47].
- **(Prompt Fidelity)** We use the CLIP score [21] to evaluate how well the generated images adhere to the text prompt. Additionally, we assess our method using PickScore [28] and ImageReward [49], which provide human alignment scores based on the consistency between the text prompt and generated images.

Method	HRS			DrawBench
	Spatial (%)	Size (%)	Color (%)	Spatial (%)
Backbone: Stable Diffusion [41]				
Stable Diffusion [41]	8.48	9.18	12.61	12.50
PixArt-α [8]	17.86	11.82	19.10	20.00
Layout-Guidance [9]	16.47	12.38	14.39	36.50
Attention-Refocusing [38]	24.45	16.97	23.54	43.50
BoxDiff [48]	16.31	11.02	13.23	30.00
R&B [47]	30.14	26.74	32.04	55.00
Backbone: PixArt-α [8]				
PixArt-R&B	37.13	20.76	29.07	**60.00**
GrounDiT (Ours)	**45.01**	**27.75**	**35.67**	**60.00**

Table 1: Quantitative comparisons of grounding accuracy on HRS [3] and DrawBench [43] benchmarks. **Bold** represents the best, and underline represents the second best method.

6.2 Grounding Accuracy

Quantitative Comparison. Tab. 1 presents a quantitative comparison of grounding accuracy between our method, GrounDiT, and baselines. GrounDiT outperforms all baselines across different criteria of grounding accuracy—spatial, size, and color—including the state-of-the-art R&B [47] and our internal baseline PixArt-R&B. Notably, the spatial accuracy on the HRS benchmark [3] (Col. 1) of GrounDiT is significantly higher, with a +14.87% improvement over R&B and +7.88% over PixArt-α. The comparison between PixArt-α [8], PixArt-R&B and GrounDiT highlights the effectiveness of the two-stage pipeline of GrounDiT. First, integrating the loss-based Global Update into PixArt-α results in a substantial improvement in spatial accuracy (from 17.86% to 37.13%).



“A dog in the beautiful park.”

“An eagle is flying over a tree.”

“A duck wearing a hat standing near a bicycle.”

“A plastic bottle and an apple on a table.”

“An apple and a banana and a cup on a table.”

“"A dog wearing sunglasses and a red hat and a blue tie.”

“A chair and a table and a bed is on the room with a photo frame on the wall and a ceiling lamp [...]”

“A dog and a bird sitting on a branch while an eagle is flying in the sky.”

“A car and a dog on the road while horse and a chair is on the grass.”

Figure 4: Qualitative comparisons between our GrounDiT and baselines. Leftmost column shows the input bounding boxes, and columns 2-6 include the baseline results. The rightmost column includes the results of our GrounDiT.

Then, incorporating our key contribution, the Local Update, further boosts accuracy (from 37.13% to 45.01%). For size accuracy (Col. 2), which evaluates how well the size of each generated object matches its corresponding bounding box, GROUNDIT shows a +1.01% improvement over R&B. In terms of color accuracy (Col. 3), our method achieves a +6.60% improvement over PixArt-R&B and outperforms R&B by +3.63%p. This underscores the effectiveness of our patch transplantation technique in accurately assigning color descriptions to the corresponding objects. As DrawBench [43] only contains images with two bounding boxes, which are relatively easy to generate, employing the Global Update is sufficient for grounding. We present additional quantitative comparisons of grounding accuracy in the **Supplementary (Sec. A.2)**.

Qualitative Comparison. Fig. 4 presents the qualitative comparisons. When the grounding condition involves one or two simple bounding boxes (Rows 1, 2), both our method and the baselines successfully generate objects within the designated regions. However, as the number of bounding boxes increases and the grounding conditions become more challenging, the baselines struggle to correctly place each object inside the bounding box (Rows 4, 8), or even fail to generate the object at all (Rows 5, 7, 9). In contrast, GROUNDIT successfully grounds each object within the boxes, even when the number of boxes is relatively high, such as four boxes (Rows 5, 6, 8), five boxes (Row 7) and six boxes (Row 9). This highlights that our proposed patch transplantation technique provides superior control over each bounding box, addressing the limitations of previous loss-based update methods, as discussed in Sec. 5.1. Since our method is based on DiT, it can generate images with **various aspect ratios**, all while incorporating grounding capabilities. The results demonstrating this can be seen in Fig. 1 and Fig. 5 in the Supplementary. For more qualitative comparisons, please refer to the **Supplementary (Sec. A.7)**.

6.3 Prompt Fidelity

Tab. 2 presents a quantitative comparison of prompt fidelity between our method and PixArt-R&B. Each metric is measured using the generated images from the HRS dataset [3]. GROUNDIT achieves higher CLIP score [21] than PixArt-R&B (Col. 1), indicating that our patch transplantation improves the text prompt fidelity of the generated images. Additionally, our method achieves a higer ImageReward [49] score, which measures human preference by considering both prompt fidelity and overall image quality. While GROUNDIT shows a slight underperformance compared to PixArt-R&B in Pickscore [28], it remains comparable overall. We provide further comparisons of prompt fidelity with other baselines in the **Supplementary Sec. A.3.**

Method	CLIP score ↑	ImageReward ↑	PickScore ↑
PixArt-R&B	33.49	0.28	**0.52**
GROUNDIT (Ours)	**33.63**	**0.44**	0.48

Table 2: Quantitative comparisons on prompt fidelity on HRS benchmark [3]. **Bold** represents the best method.

7 Conclusion

In this work, we introduced GROUNDIT, a training-free spatial grounding technique for text-to-image generation, leveraging Diffusion Transformers (DiT). To address the limitation of prior approaches, which lacked fine-grained spatial control over individual bounding boxes, we proposed a novel approach that transplants an image patch generated in a separate denoising branch into the designated area of the main image. By exploiting an intriguing property of DiT, semantic sharing, which arises from the flexibility of the Transformer architecture and the use of positional embeddings, GROUNDIT generates a smaller patch by simultaneously denoising two noisy image: one with a smaller size and the other with a generatable size by DiT. Through semantic sharing, these two noisy images become "semantic clones", enabling fine-grained spatial control for each bounding box. Our experiments on the HRS and DrawBench benchmarks demonstrated that GROUNDIT achieves state-of-the-art performance compared to previous training-free grounding methods.

Limitations and Societal Impacts. A limitation of our method is the increased computation time, as it requires a separate object branch for each bounding box. We provide further analysis on the computation time in the **Supplementary (Sec. A.6)**. Additionally, like other generative AI techniques, our method is susceptible to misuse, such as creating deepfakes, which can raise significant concerns related to privacy, bias, and fairness. It is crucial to develop safeguards to control and mitigate these risks responsibly.

Acknowledgements

We thank Juil Koo and Jaihoon Kim for valuable discussions on Diffusion Transformers. This work was supported by the NRF grant (RS-2023-00209723), IITP grants (RS-2019-II190075, RS-2022-II220594, RS-2023-00227592, RS-2024-00399817), and KEIT grant (RS-2024-00423625), all funded by the Korean government (MSIT and MOTIE), as well as grants from the DRB-KAIST SketchTheFuture Research Center, NAVER-Intel, Hyundai NGV, KT, and Samsung Electronics.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *CVPR*, 2023.

[3] Eslam Mohamed Bakr, Pengzhan Sun, Xiaogian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *ICCV*, 2023.

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

[5] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing diffusion paths for controlled image generation. In *ICML*, 2023.

[6] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. `https://openai.com/research/video-generation-models-as-world-simulators`, 2024.

[7] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics*, 2023.

[8] Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024.

[9] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *WACV*, 2024.

[10] Jiaxin Cheng, Xiao Liang, Xingjian Shi, Tong He, Tianjun Xiao, and Mu Li. Layoutdiffuse: Adapting foundational diffusion models for layout-to-image generation. *arXiv preprint arXiv:2302.08908*, 2023.

[11] Guillaume Couairon, Marlène Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. *arXiv preprint arXiv:2403.13589*, 2023.

[12] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *NeurIPS*, 2023.

[13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

[14] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *ICLR*, 2023.

[15] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393*, 2023.

[16] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022.

[17] Songwei Ge, Taesung Park, Jun-Yan Zhu, and Jia-Bin Huang. Expressive text-to-image generation with rich text. In *ICCV*, 2023.

[18] Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. Initno: Boosting text-to-image diffusion models via initial noise optimization. In *CVPR*, 2024.

[19] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *ICLR*, 2023.

[20] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In *CVPR*, 2024.

[21] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[23] Hyeonho Jeong and Jong Chul Ye. Ground-a-video: Zero-shot grounded video editing using text-to-image diffusion models. In *ICLR*, 2024.

[24] Gwanghyun Kim, Hayeon Kim, Hoigi Seo, Dong Un Kang, and Se Young Chun. Beyondscene: Higher-resolution human-centric scene generation with pretrained diffusion. *arXiv preprint arXiv:2404.04544*, 2024.

[25] Jaihoon Kim, Juil Koo, Kyeongmin Yeo, and Minhyuk Sung. Synctweedies: A general generative framework based on synchronized diffusions. *arXiv preprint arXiv:2403.14370*, 2024.

[26] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *ICCV*, 2023.

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[28] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *NeurIPS*, 2023.

[29] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. SyncDiffusion: Coherent montage via synchronized joint diffusions. In *NeurIPS*, 2023.

[30] Yuseung Lee and Minhyuk Sung. Reground: Improving textual and spatial grounding at no cost. *arXiv preprint arXiv:2403.13589*, 2024.

[31] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023.

[32] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *TMLR*, 2024.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[34] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. *arXiv preprint arXiv:2311.12891*, 2023.

[35] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022.

[36] Wan-Duo Kurt Ma, Avisek Lahiri, JP Lewis, Thomas Leung, and W Bastiaan Kleijn. Directed diffusion: Direct control of object placement through attention guidance. In *AAAI*, 2024.

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[38] Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. *arXiv preprint arXiv:2306.05427*, 2023.

[39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference*, 2015.

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

[44] Takahiro Shirakawa and Seiichi Uchida. Noisecollage: A layout-aware text-to-image diffusion model based on noise cropping and merging. In *CVPR*, 2024.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[46] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *CVPR*, 2024.

[47] Jiayu Xiao, Liang Li, Henglei Lv, Shuhui Wang, and Qingming Huang. R&b: Region and boundary aware zero-shot grounded text-to-image generation. In *ICLR*, 2024.

[48] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *ICCV*, 2023.

[49] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *NeurIPS*, 2023.

[50] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *ICML*, 2024.

[51] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

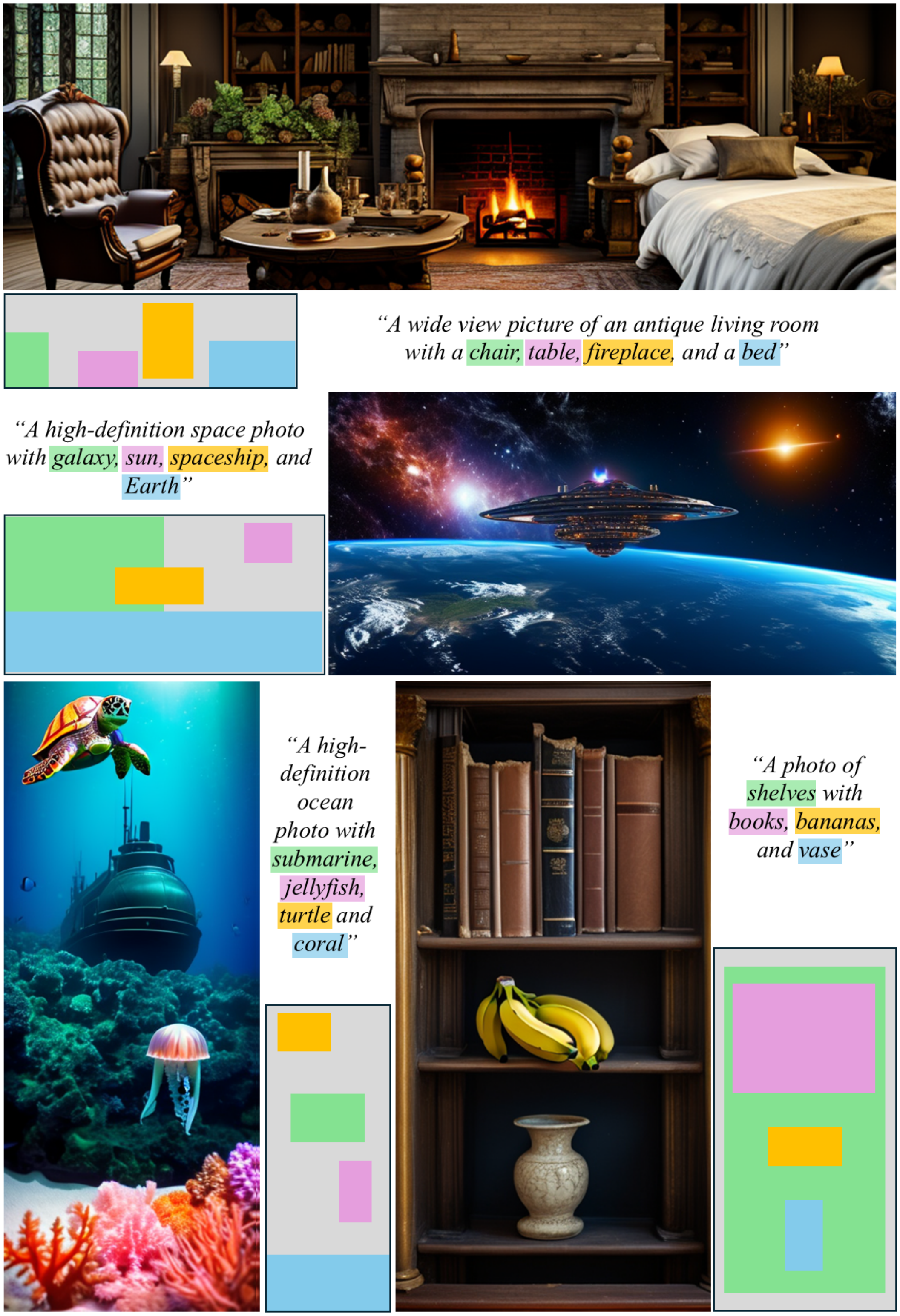
[52] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[53] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *CVPR*, 2023.

[54] Dewei Zhou, You Li, Fan Ma, Xiaoting Zhang, and Yi Yang. Migc: Multi-instance generation controller for text-to-image synthesis. In *CVPR*, 2024.

[55] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *CVPR*, 2022.

A Appendix



"A wide view picture of an antique living room with a chair, table, fireplace, and a bed"

"A high-definition space photo with galaxy, sun, spaceship, and Earth"

"A high-definition ocean photo with submarine, jellyfish, turtle and coral"

"A photo of shelves with books, bananas, and vase"

Figure 5: Spatially grounded images generated by our GrounDiT with varying *aspect ratios* and *sizes*. Each image is generated based on a text prompt along with bounding boxes, which are displayed next to (or below) each image.

A.1 Positional Embeddings in Diffusion Transformers

Diffusion Transformers (DiT) [37] handle noisy images of varying *aspect ratios* and *sizes* by processing them as sequences of tokens. For this, the noisy image is first divided into patches, with each patch subsequently converted into an image token of hidden dimension D through a linear embedding layer. DiT then applies 2D sine-cosine positional embeddings to each image token, based on its coordinates (x, y), defined as follows:

$$p_{x,y} := \text{CONCAT}\left[p_x,\ p_y\right], \quad \text{where} \quad p_x := \left[\cos(w_d \cdot x),\ \sin(w_d \cdot x)\right]_{d=0}^{D/4}$$
$$p_y := \left[\cos(w_d \cdot y),\ \sin(w_d \cdot y)\right]_{d=0}^{D/4}$$

where $w_d = 1/10000^{(4d/D)}$. The positional embedding $p_{x,y}$ is then added to each corresponding image token, denoted as $\mathtt{PE}(\cdot)$.

A.2 Additional Quantitative Comparisons: Grounding Accuracy

In addition to Sec. 6.2, we provide further quantitative comparisons of grounding accuracy between our GrounDiT and the baselines. Specifically, we generated images based on text prompts and bounding boxes using each method, then calculated the mean Intersection over Union (mIoU) between the detected bounding boxes from object detection [55] and the input bounding boxes. Below, we present the quantitative comparisons across three datasets with varying average numbers of bounding boxes: subset of MS-COCO-2014 [33], HRS-Spatial [3], and a custom dataset.

Dataset	Subset of MS-COCO-2014 [33]	HRS-Spatial [3]	Custom Dataset
Avg. # of Bounding Boxes	2.06	3.11	4.48

Table 3: Average number of bounding boxes per dataset.

Subset of MS-COCO-2014. We filtered the validation set of MS-COCO-2014 [33] to exclude image-caption pairs where the target objects were either not mentioned in the captions or duplicate objects were present. From this filtered set, we randomly selected 500 pairs for evaluation.

The results are presented in Tab. 4, column 2. GrounDiT outperforms R&B by 0.021 (a 5.1% improvement) and PixArt-R&B by 0.014 (a 2.2% improvement). The relatively small margin can be attributed to the simplicity of the task, as this dataset has **an average of 2.06 bounding boxes** (Tab. 3), making it less challenging even for the baseline methods.

HRS-Spatial. Column 3 of Tab. 4 presents the results on the *Spatial* subset of the HRS dataset [3]. GrounDiT surpasses R&B [47] by 0.046 (a 14.1% improvement) and PixArt-R&B by 0.038 (an 11.4% improvement). Compared to the results on the MS-COCO-2014 subset, the higher percentage increase in mIoU highlights the robustness of GrounDiT under more complex grounding conditions. Note that HRS-Spatial has **an average of 3.11 bounding boxes** (Tab. 3), which is higher than that of the MS-COCO-2014 subset (2.06).

Custom Dataset. The custom dataset consists of 500 layout-text pairs, generated using the layout generation pipeline from LayoutGPT [15]. As shown in column 4 of Tab. 4, GrounDiT outperforms R&B by 0.052 (a 26.3% improvement) and PixArt-R&B by 0.044 (a 21.4% improvement). This dataset has the **highest average number of bounding boxes at 4.48** (Tab. 3). These results further emphasize the robustness and effectiveness of our approach in handling more complex grounding conditions with a larger number of bounding boxes.

A.3 Additional Quantitative Comparisons: Prompt Fidelity

In addition to Sec. 6.3, we provide further quantitative comparisons of the prompt fidelity of the generated images between our GrounDiT and the baselines. We evaluated the generated images from the HRS dataset [3] using three different metrics: CLIP score [21], ImageReward [49], and PickScore [28]. The results are presented in Tab. 5. Since PickScore evaluates preferences between a pair of images, we report the difference between our GrounDiT and each baseline method in column 4. Our GrounDiT consistently outperforms the baselines in both CLIP score and ImageReward. For PickScore, GrounDiT outperforms all baselines except PixArt-R&B, while remaining comparable.

Method	Subset of MS-COCO-2014 [33]	HRS-Spatial [3]	Custom Dataset
Backbone: Stable Diffusion [41]			
Stable Diffusion [41]	0.176	0.068	0.030
PixArt-α [8]	0.233	0.085	0.036
Layout-Guidance [9]	0.307	0.199	0.122
Attention-Refocusing [38]	0.254	0.145	0.078
BoxDiff [48]	0.324	0.164	0.106
R&B [47]	0.411	0.326	0.198
Backbone: PixArt-α [8]			
PixArt-R&B	0.418	0.334	0.206
GrounDiT (Ours)	**0.432**	**0.372**	**0.250**

Table 4: Quantitative comparisons of mIoU (↑) on a subset of MS-COCO-2014 [33], HRS-Spatial [3], and our custom dataset. **Bold** represents the best, and underline represents the second best method.

Method	CLIP score ↑	ImageReward ↑	PickScore ↑ (Ours − Baseline)
Backbone: Stable Diffusion [41]			
Layout-Guidance [9]	32.48	-0.401	+0.30
Attention-Refocusing [38]	31.36	-0.508	+0.22
BoxDiff [48]	32.57	-0.199	+0.30
R&B [47]	33.16	-0.021	+0.26
Backbone: PixArt-α [8]			
PixArt-R&B	33.49	0.280	-0.04
GrounDiT (Ours)	**33.63**	**0.444**	-

Table 5: Quantitative comparisons of prompt fidelity on the HRS dataset [3]. **Bold** represents the best method.

A.4 Additional Analysis on Semantic Sharing

In this section, we provide further analysis on the semantic sharing property of DiT, initially introduced in Sec. 5.2.

Generatable resolution of DiT. Although recent DiT models can generate images at various resolutions, they still struggle to produce images at *completely arbitrary* resolutions. We speculate that this limitation arises not from the model architecture itself, but from the resolution of the training images, which typically falls within a specific range [8]. Generating images at resolutions far outside this range often results in implausible outputs, suggesting the existence of an acceptable resolution range for DiT, which we refer to as its *generatable resolution*. In Fig. 6, we illustrate this phenomenon. When the noisy image size falls within DiT's generatable resolution range, the model produces plausible images (rightmost two images). However, when the image size is significantly outside this range (leftmost two images), DiT fails to generate a plausible image.

Semantic Sharing. Even though DiT models have a limited range of generatable resolutions, their Transformer architecture offers flexibility in handling varying lengths of image tokens, making it feasible to merge two sets of image tokens and denoise them through a single network evaluation. Leveraging this flexibility of Transformers, we presented our joint token denoising technique (Alg. 1). Our main observation was that the joint token denoising between two noisy images causes the two generated images to become semantically correlated, as illustrated in Fig. 3-(B) and Fig. 3-(C).

In addition to the visualizations in Fig. 3, we further quantify the semantic sharing property by measuring the LPIPS score [52] between two generated images. To explore the effect of joint token denoising, we varied the parameter $\gamma \in [0, 1]$, which controls the proportion of denoising steps where joint token denoising is applied. Specifically, $\gamma = 0$ means no joint token denoising is applied, and each image is denoised independently, while $\gamma = 1$ means full joint token denoising across all steps.

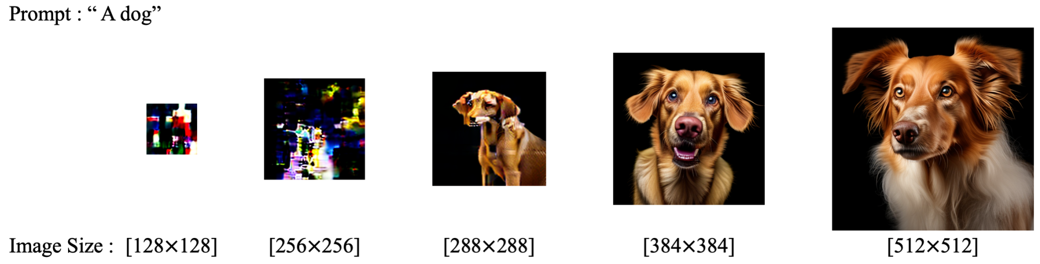
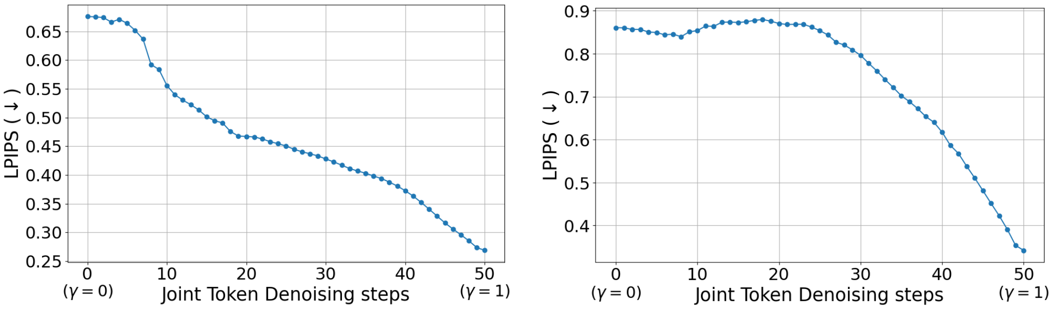


Figure 6: Illustration of the generatable resolution range of DiT. The images are generated using PixArt-α [8] from the text prompt *“A dog”*, with varying resolutions.

As shown in Fig. 7, increasing (*i.e.* applying more joint token denoising steps) results in a decrease in the LPIPS score between the two generated images, indicating that the images become more similar as joint denoising is applied for a larger portion of the denoising process.



(a) Semantic sharing between equal resolutions

(b) Semantic sharing between different resolutions

Figure 7: LPIPS score between two generated images with varying γ value. A gradual decrease in LPIPS [52] indicates that joint token denoising progressively enhances the similarity between the generated images.

A.5 Implementation Details

As the based text-to-image DiT model, we used the 512-resolution version of PixArt-α [8]. For sampling we employed DPM-Solver scheduler [35] with 50 steps. Out of the 50 denoising steps, we applied our GROUNDIT denoising step for the initial 25 steps, and applied the vanilla denoising step for the remaining 25 steps. For the grounding loss in Stage 1 of GROUNDIT, we adopted the definition proposed in R&B [47], and we set the loss scale to 10 and used a gradient descent weight of 5 for the gradient descent update in Eq. 7.

As discussed in Sec. 5.3, for each i-th Object Branch we have a noisy object image $u_{i,t}$ and a noisy local patch $v_{i,t}$, which is extracted from the noisy image $\hat{\mathbf{x}}_t$ in main branch via $\mathbf{v}_{i,t} \leftarrow \texttt{Crop}(\hat{\mathbf{x}}_t, b_i)$. In practice, we pre-define the resolution of the noisy object image $u_{i,t}$ by searching in PixArt-α's generatable resolutions that closely match the aspect ratio of the correponding bounding box b_i. All our experiments were conducted an NVIDIA RTX 3090 GPU. In Algorithm 2, we provide the pseudocode of GROUNDIT single denoising step.

Algorithm 2: Pseudocode of GROUNDIT denoising step.

```
Parameters : $\omega_t$;                                                 // Gradient descent weight.
Inputs: $\mathbf{x}_t, \{\mathbf{u}_{i,t}\}_{i=0}^{N-1}, G, c$;      // Noisy images, grounding conditions, text embedding.
Outputs: $\mathbf{x}_{t-1}, \{\mathbf{u}_{i,t-1}\}_{i=0}^{N-1}$;                   // Noisy images at timestep $t-1$.
Function GlobalUpdate($\mathbf{x}_t, t, c, G$) :
    // $b_i$ holds coordinate information of bounding box, (Sec. 4)
    $\{A_{s_i,t}\}_{i=0}^{N-1} \leftarrow$ ExtractAttention($\mathbf{x}_t, t, c, G$);        // Extract cross-attention maps.
    $\mathcal{L}_{\text{AGG}} \leftarrow \sum_{i=0}^{N-1} \mathcal{L}(A_{s_i t}, b_i)$;                // Compute aggregated grounding loss.
    $\hat{\mathbf{x}}_t \leftarrow \mathbf{x}_t - \omega_t \nabla_{\mathbf{x}_t} \mathcal{L}_{\text{AGG}}$;                          // Gradient descent (Eq. 7)
    return $\hat{\mathbf{x}}_t$;
Function LocalUpdate($\hat{\mathbf{x}}_t, \{\mathbf{u}_{i,t}\}_{i=0}^{N-1}, t, c, G$) :
    $\tilde{\mathbf{x}}_{t-1} \leftarrow$ Denoise($\hat{\mathbf{x}}_t, t, c$);                                  // Main branch
    for $i = 0, \ldots, N-1$ do
        // $i$-th object branch
        $\mathbf{v}_{i,t} \leftarrow$ Crop($\hat{\mathbf{x}}_t, b_i$);                          // Obtain noisy local patch.
        $\{\mathbf{u}_{i,t-1}, \mathbf{v}_{i,t-1}\} \leftarrow$ JointTokenDenoise($\mathbf{u}_{i,t}, \mathbf{v}_{i,t}, t, c_i$);    // Joint token denoising.
    for $i = 0, \ldots, N-1$ do
        // $m_i$ is a binary mask corresponding to $b_i$
        $\tilde{\mathbf{x}}_{t-1} \leftarrow \tilde{\mathbf{x}}_{t-1} \odot (1 - m_i) +$ Uncrop($\mathbf{v}_{i,t-1}, b_i$) $\odot m_i$;       // Patch Transplantation.
    $\mathbf{x}_{t-1} \leftarrow \tilde{\mathbf{x}}_{t-1}$
    return $\mathbf{x}_{t-1}, \{\mathbf{u}_{i,t-1}\}_{i=0}^{N-1}$;
Function GrounDiTStep($\mathbf{x}_t, \{\mathbf{u}_{i,t}\}_{i=0}^{N-1}, t, c, G$):
    $\hat{\mathbf{x}}_t \leftarrow$ GlobalUpdate($\mathbf{x}_t, t, c, G$) ;                           // Global update (Sec. 5.1)
    $\mathbf{x}_{t-1}, \{\mathbf{u}_{i,t-1}\}_{i=0}^{N-1} \leftarrow$ LocalUpdate($\hat{\mathbf{x}}_t, \{\mathbf{u}_{i,t}\}_{i=0}^{N-1}, t, c, G$) ;  // Local update (Sec. 5.3)
    return $\mathbf{x}_{t-1}, \{\mathbf{u}_{i,t-1}\}_{i=0}^{N-1}$;
```

A.6 Analysis on Computation Time

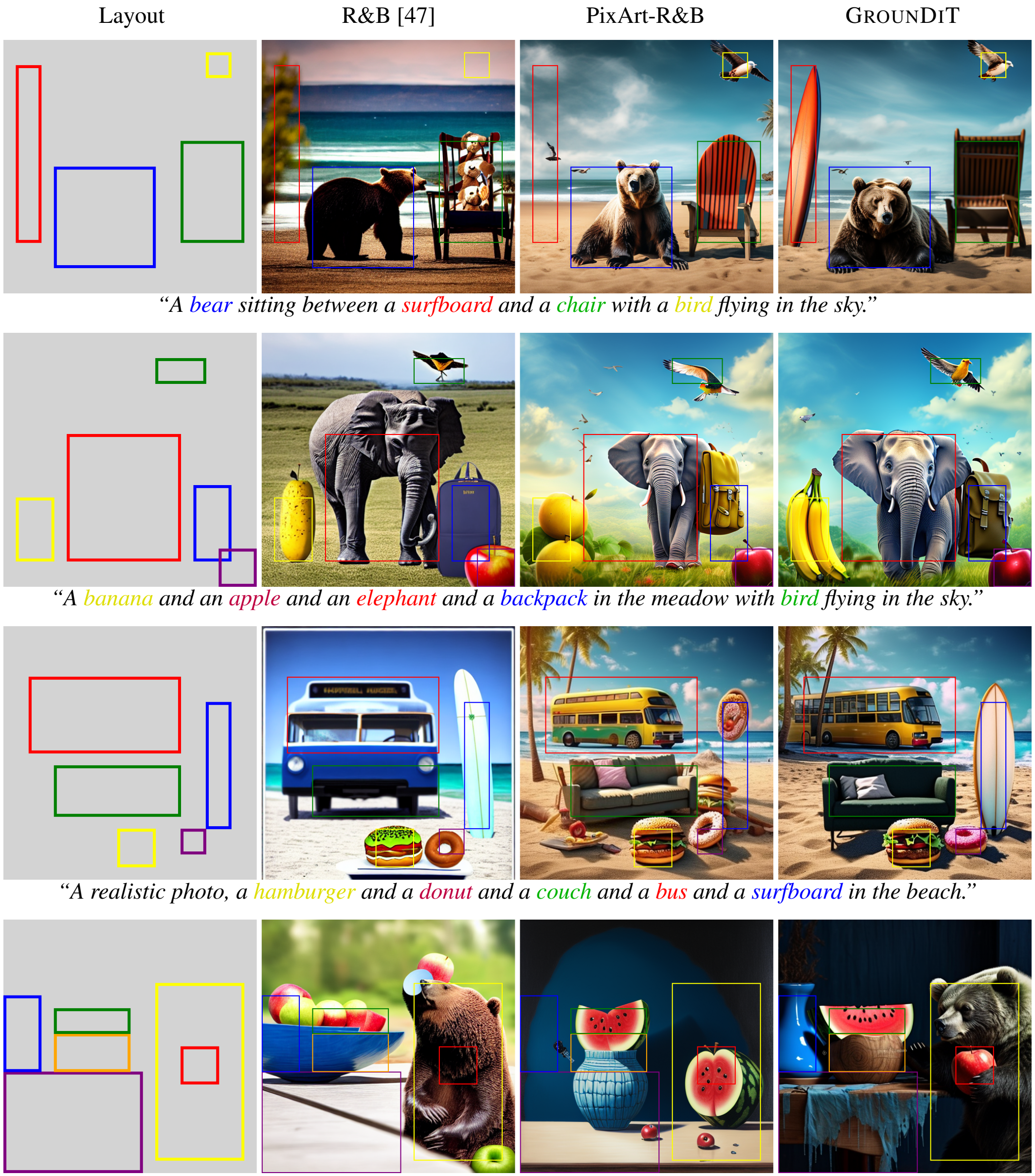
We present the average execution time based on the number of bounding boxes in Tab. 6. While our method shows a slight increase in inference time, the rate of increase remains modest. For three bounding boxes, the inference time is 1.01 times that of R&B and 1.33 times that of PixArt-R&B. Even with six bounding boxes, the inference time is only 1.41 times that of R&B and 1.90 times that of PixArt-R&B.

# of bounding boxes	3	4	5	6
R&B [47]	37.52	38.96	39.03	39.15
PixArt-R&B	28.31	28.67	29.04	29.15
GROUNDIT (Ours)	37.71	41.10	47.83	55.30

Table 6: Comparison of average execution time based on the number of bounding boxes. Values in the table are given in seconds

A.7 Additional Qualitative Results

We provide more qualitative comparisons in Fig. 8. Our method demonstrates greater robustness against issues such as the missing object problem, attribute leakage, or object interruption problem [47], due to its local update mechanism with semantic sharing. For instance, in Row 1, baseline methods struggle to generate certain objects (*i.e.* **missing object problem**). In Row 2, baselines generate a banana that retains features of an apple, illustrating **attribute leakage**. In Row 3, R&B generates a bus that interrupts the generation of a couch, with part of the bus overlapping with the designate region of the couch. Similarly, in PixArt-R&B, a hamburger and a donut interrupt the generation of a surfboard, demonstrating the **object interruption problem**. In more challenging cases, like Row 4, combinations of these issues appear. By contrast, our method consistently generates each object accurately within specified locations, even under complex bounding box configurations, highlighting its robustness and precision. Additional results are shown in Fig. 9, and examples of various aspect ratio images generated with grounding conditions are provided in Fig. 5.



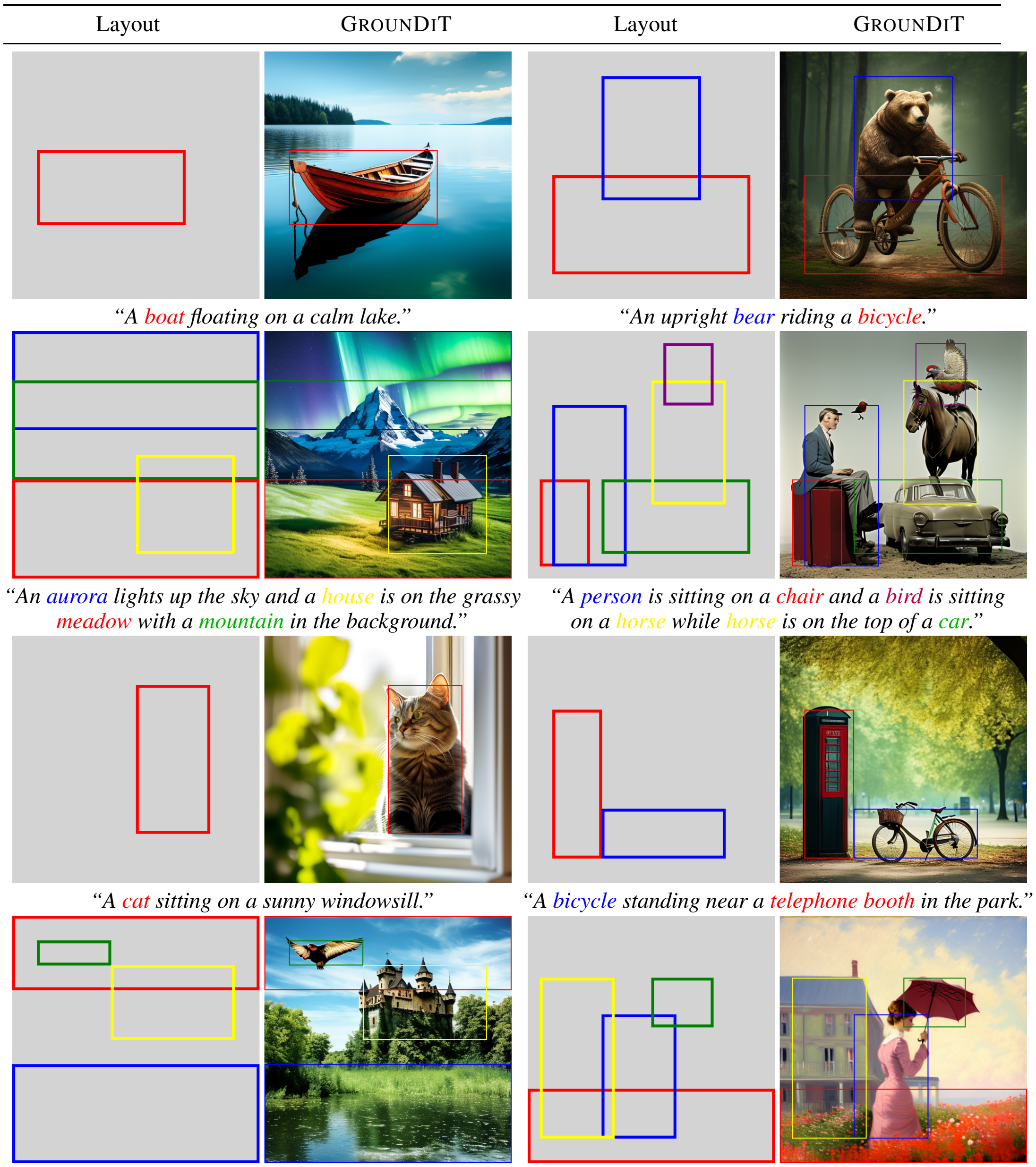
“A bear sitting between a surfboard and a chair with a bird flying in the sky.”

“A banana and an apple and an elephant and a backpack in the meadow with bird flying in the sky.”

“A realistic photo, a hamburger and a donut and a couch and a bus and a surfboard in the beach.”

“A blue vase and a wooden bowl with a watermelon sit on a table, while a bear holding an apple.”

Figure 8: Additional qualitative comparisons between our GrounDiT, the previous state-of-the-art, R&B [47], and our internal baseline PixArt-R&B. Leftmost column shows the input bounding boxes, and columns 2-3 include the baseline results. The rightmost column includes the results of our GrounDiT.



"A boat floating on a calm lake."

"An upright bear riding a bicycle."

"An aurora lights up the sky and a house is on the grassy meadow with a mountain in the background."

"A person is sitting on a chair and a bird is sitting on a horse while horse is on the top of a car."

"A cat sitting on a sunny windowsill."

"A bicycle standing near a telephone booth in the park."

"A castle stands across the lake and the bird flies in the blue sky."

"A Monet painting of a woman standing on a flower field holding an umbrella sideways with a house in the background."

Figure 9: Additional spatially grounded images generated by out GROUNDIT.