Directed Diffusion: Direct Control of Object Placement through Attention Guidance

Wan-Duo Kurt Ma[1], **Avisek Lahiri**[2], **J.P. Lewis**[3*], **Thomas Leung**[2], **W. Bastiaan Kleijn**[1,2]
[1]Victoria University of Wellington
[2]Google Research
[3]NVIDIA Research
mawand@ecs.vuw.ac.nz, jpl@nvidia.com, avisek@google.com, leungt@google.com, bastiaan.kleijn@vuw.ac.nz

Abstract

Text-guided diffusion models such as DALL·E 2, Imagen, eDiff-I, and Stable Diffusion are able to generate an effectively endless variety of images given only a short text prompt describing the desired image content. In many cases the images are of very high quality. However, these models often struggle to compose scenes containing several key objects such as characters in specified positional relationships. The missing capability to "direct" the placement of characters and objects both within and across images is crucial in storytelling, as recognized in the literature on film and animation theory. In this work, we take a particularly straightforward approach to provide the needed direction. Drawing on the observation that the cross-attention maps for prompt words reflect the spatial layout of objects denoted by those words, we introduce an optimization objective that produces "activation" at desired positions in these cross-attention maps. The resulting approach is a step toward generalizing the applicability of text-guided diffusion models beyond single images to collections of related images, as in storybooks. Directed Diffusion provides easy high-level positional control over multiple objects, while making use of an existing pre-trained model and maintaining a coherent blend between the positioned objects and the background. Moreover, it requires only a few lines to implement.

Introduction

Text-to-image models such as DALL·E 2 (Ramesh et al. 2022), Imagen (Saharia et al. 2022) and eDiff-I (Balaji et al. 2022) have revolutionized image generation. Platforms such as Stable Diffusion (StabilityAI 2023) and similar systems have democratized this capability, as well as presenting new ethical challenges.

The forementioned systems generate arbitrary images simply by typing a "prompt" or description of the desired image. It is not always highlighted, however, that experimentation and practical experience are often needed if the user has a particular result in mind. Text-to-image diffusion methods often fail to produce desired results, requiring repeated trial-and-error experiments with prompt choices including negative prompts, different random seeds, and hyperparameters including the classifier-free guidance scale, number of denoising steps, and scheduler (Smith 2022). This is particularly true for complex prompts involving descriptions of several objects. For example, in SD a prompt such as *"a bird flying over a house"* fails to generate the house with some seeds, or renders the bird and house but without the "on" relationship. These difficulties have led to the creation of "prompt marketplaces" [1] where expert users share and sell successful settings.

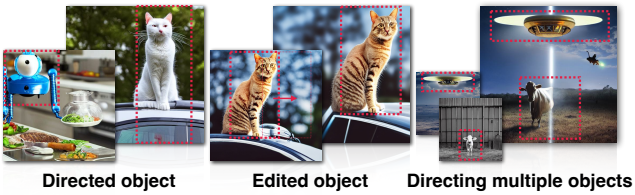

Figure 1: Directed Diffusion[2] (DD) augments denoising diffusion text-to-image generation by allowing the position of specified objects to be controlled with user-specified bounding boxes as highlighted in red. Left: DD generates specified objects (e.g., insect robot, cat) placed according to the given bounding boxes. Middle: We can move a synthesized object, and Right: place multiple objects in desired locations. All the directed objects show the appropriate "contextual" interaction (e.g. shadows) with the background.

The experimentation becomes prohibitive when the goal is to use the images for *storytelling*, which involves established principles for positioning characters relative to each other and the (virtual) camera (Arijon 1976; Thomas and Johnston 1981). Text prompts describing the content of an image do not indicate *where* objects should be placed, and indicating desired positions in the prompt usually fails. As a consequence, extensive and tedious repeated trials are needed to obtain an image where the desired objects exist and their generated positions are acceptable.

Research is addressing some limitations of text-guided diffusion methods, including methods that define new text tokens to denote specific and consistent character or object identities, provide mask-guided inpainting of particular regions, manipulate text guidance representations, and miti-

*Work was done at Google Research.

[1]https://csform.com/best-prompt-marketplaces-for-ai-art
[2]https://hohonu-vicml.github.io/DirectedDiffusion.Page

gate the common failure of guidance using CLIP-like models (Radford et al. 2021) to associate colors with the correct object (Feng et al. 2022). Existing methods still generally struggle to synthesize *several* objects with desired positional relationships. Our preprint (Ma et al. 2023) depicts this issue in the experiments and supplementary material.

Our work is a further step toward guiding text-based diffusion, by introducing *coarse positional control for several objects* as needed for storytelling. In this application only *coarse* positional control is needed – for example, a director might instruct an actor to "start from over there and walk toward the door", rather than specifying the desired positions in floating point precision as is done in animation software packages. We take inspiration from the observation that position is established early in the denoising process (Liew et al. 2022) and from the fact that the cross-attention maps have a clear spatial interpretation (see Fig. 2). Our general approach is to edit the cross-attention maps for particular words during the early denoising steps, so as to concentrate activation at the desired location of the object. We introduce an optimization objective that achieves this without disrupting the learned text-image association in the pretrained model and without requiring extensive code changes. Our method is implemented using the Python DIFFUSERS[3] implementation of stable diffusion (SD) and uses the available pre-trained SD 1.5 model.

Our method makes the following contributions:

- **Storytelling.** Our method is a step towards storytelling by providing consistent control over the positioning of multiple objects.
- **Compositionality.** It provides a direct approach to "compositionality" by providing explicit positional control.
- **Consistency.** The positioned objects seamlessly and consistently fit in the environment, rather than appearing as a splice from another image with inconsistent interaction (shadows, lighting, etc.) This consistency is due to two factors. First, we use a simple bounding box to position and allow the denoising diffusion process to fill in the details, whereas specifying position with a pixel-space mask runs the risk that it may be inconsistent with the shape and position implicit in the early denoising results. Second, subsequent diffusion steps operate on the entire image and seamlessly produce consistent lighting, etc.
- **Simplicity.** Image editing methods for text-to-image models often require detailed masks, depth maps, or other precise guidance information. This information is not available when synthesizing images *ab initio*, and it would be laborious to create. Our method allows the user to control the desired locations of objects simply by specifying approximate bounding boxes.

  From a computational point of view, our method requires no training or fine tuning, and can be added to an existing text-driven diffusion model with cross-attention guidance with only a few lines of code. It requires a simple optimization of a small weight vector $\mathbf{a} \in \mathbb{R}^d, d < 77$. This does not significantly increase the overall synthesis time and generally removes the need for the user to search for a per-word weight parameter.

Storytelling also requires the ability to generate particular objects rather than generic instances (*this* cat rather than "any cat"), and our algorithm is complementary to methods (Gal et al. 2022; Ruiz et al. 2022) that address this. For clarity, ***the examples in the paper do not make use of these complementary algorithms.***

[3]https://github.com/huggingface/diffusers

Related Work

Denoising diffusion models (Sohl-Dickstein et al. 2015; Song and Ermon 2019; Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2021) add Gaussian noise to the data in a number of steps and then train a model to incrementally remove this noise. Aspects of the mathematics are presented in most papers and good tutorials are available (Weng 2021), so we will simply mention several high-level intuitions. The end-result of the forward process is an image effectively consisting of independent normal distributed pixels, which is easy to sample. The backward process denoises random samples from this distribution resulting in novel images, or other data. The mathematical derivation (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) somewhat resembles a hierarchical version of a VAE (Kingma and Welling 2013), although with a fixed encoder and a "latent" space with the same dimensionality as the data. The fixed encoder provides an easy closed-form posterior for each step, allowing the overall loss to split into a sum over uncoupled terms for each denoising step, resulting in faster training. (Song and Ermon 2019) introduced an alternate derivation building on score matching (Hyvärinen 2005). From this perspective adding noise is equivalent to convolving the probability density of the noise with the data, thus blurring the data distribution and providing gradients toward the data manifold from distant random starting points. The denoising process in (Ho, Jain, and Abbeel 2020; Song and Ermon 2019) is stochastic, with an interpretation as Langevin sampling (Song and Ermon 2019). A deterministic variant (Song, Meng, and Ermon 2021) is widely used for image editing applications.

Text-to-image (T2I) models condition the image generation process on the text representation from joint text-image embedding models such as CLIP (Radford et al. 2021), thereby providing the ability to synthesize images simply by typing a phrase that describes the desired image. While T2I models have employed GANs as well as autoregressive models and transformers (Patashnik et al. 2021; Yu et al. 2022; Chang et al. 2023), a number of recent successful approaches use diffusion models (Nichol et al. 2022; Ramesh et al. 2022; Saharia et al. 2022). This choice reflects both the stable training of these models and their ability to learn diverse multi-subject datasets.

Our goal is to help democratize storytelling by enabling high-level open-set, zero-shot placement of several objects in T2I denoising-diffusion synthesis, *while exploiting a pretrained model* to avoid requiring computational resources beyond those available to the typical user. To achieve this

we note the following criteria:

- While a number of methods e.g. (Avrahami, Fried, and Lischinski 2022; Avrahami et al. 2022) use *shape masks* (silhouettes) to guide object placement, we intentionally avoid the use of detailed shape masks, since non-artist users have difficulty producing plausible outlines of objects in perspective as illustrated in (Ma et al. 2023) supplementary. In addition, it has been noted that some shape mask methods produce poor alignment of the synthesized object to the given shape (Park et al. 2022) and may show poor interactions (e.g. shadows) between the object and background. Our approach makes use of a small optimization to guide the cross-attention maps to seamlessly place the directed object. We find that this gives better quality and is more reliable than approaches that directly edit the cross-attention maps see Fig. 9 in our supplementary.
- We desire the ability to control the placement of a "hero" character or object, as well as (possibly) the position of another object that the hero is interacting with, while including both physical interactions (e.g. shadows) and (to the extent possible) storytelling interactions ("the dog *chases* the ball"), along with optional description of the environment ("in the field"). (Liu et al. 2022) provides control over multiple objects but requires training and does not give position control. (Bar-Tal et al. 2023) demonstrates large scene composition with multiple objects but does not demonstrate specified interactions between placed objects.
- (Bar-Tal et al. 2023) note the important distinction between methods that require costly training on curated datasets and those that control the generated content by manipulating the generation process of a pre-trained model. While training custom models or fine-tuning (Avrahami et al. 2022; Li et al. 2023) offers the most power and flexibility, we exclude these approaches since they often require computational resources outside the reach of typical users. For example (Zhang and Agrawala 2023) reports requiring 100s of hours of GPU time for fine-tuning SD while (Li et al. 2023) uses 16 V100 GPUs. We base our approach on SD since it is not proprietary.

Recent papers and preprints (Li et al. 2023; Bar-Tal et al. 2023; Balaji et al. 2022; Xie et al. 2023) provide robust control over the placement of multiple objects. (Balaji et al. 2022) is a powerful trained-from-scratch system that pioneered the idea of guiding object placement by per-word injection of cross-attention activation inside a coarse shape mask, controlled by a user-specified weight. (Park et al. 2022) demonstrate high-quality editing of a single object driven by a combination of text and a detailed mask. They address the issue that the object shape arising from classifier-guided diffusion can conflict with the provided mask. Similar to our work, (Li et al. 2023) argues that high-level box guidance has advantages over shape masks, however their approach requires fine-tuning. (Xie et al. 2023) and (Bar-Tal et al. 2023) are most similar to our aims. Like ours, these systems make use of a pretrained T2I model with no fine-tuning required, and both involve a relatively lightweight placement optimization during synthesis, although the particular approaches differ. While we only recently became aware of these concurrent developments, comparisons are provided in the experiments section and supplementary.

Method

Our objective is to create a controllable synthetic image from a text-guided diffusion model without any training by manipulating the attention from cross-attention layers. We use the following notation: Bold capital letters (e.g., $\mathbf{M}$) denote a matrix or a tensor, vectors are represented with bold lowercase letters (e.g., $\mathbf{m}$), and lowercase letters (e.g., m) denote scalars. Depending on the context, the superscript i on a three-dimensional tensor (e.g., $\mathbf{M}^{(i)}$) denotes a tensor slice as a matrix. This index specifies the slice of the cross-attention map associated with a particular token in the prompt. Similarly, $\mathbf{M}^{(i:j)}$ denotes slices i to j of the tensor.

The DD procedure controls the placement of objects corresponding to several groups of selected words in the prompt; we refer to these as *directed objects* and *directed prompt words*, respectively. Our method is inspired by the intermediate result shown in Fig. 2 (Left). As shown in this figure, the overall position and shape of a synthesized object appears near the beginning of the denoising process, while the final denoising steps do not change this overall position but add details that make it identifiable as a particular object such as cat. This observation has also been exploited in previous work such as (Liew et al. 2022). An additional phenomenon can be seen in Fig. 2 (Right). The cross-attention map has a spatial interpretation, which is the "correlation" between locations in the image and the meaning of a particular word in the prompt. For instance, we can see the cat shape in the cross-attention map associated with the word "cat". DD utilizes this key observation to spatially guide the denoising process to satisfy the user's goal.

Pipeline

Given the prompt $\mathcal{P}$ and the associated region information $\mathcal{R}$ indicating the directed objects, DD synthesizes an image $\mathbf{x}_0$ with appropriate "contextual interaction" between the objects and the remainder of the scene. It uses a pre-trained Latent Diffusion Model (LDM) (Rombach et al. 2022) with no fine-tuning. The region information $\mathcal{R}$ comprises a set of parameters $\mathcal{R} = \{\mathcal{B}, \mathcal{I}\}$, denoting the bounding boxes to position the directed objects, together with the directed prompt word indices in the prompt. We will describe these in the Cross-Attention Map Guidance section below.

Following conventional notation, we define $\mathbf{x}_t$ and $\mathbf{z}_t$ as the synthesized SD image and the predicted latent noise at time step t, respectively, where $t \in \{T, ..., 0\}$ and t=0 is the final denoising step. The images $\mathbf{x}_t$ are reconstructions obtained by feeding the latent $\mathbf{z}_t$ through the VAE decoder $\mathcal{D}(\cdot)$. The images $\mathbf{x}_T$ and $\mathbf{x}_0$ correspond to the Gaussian noise latent $\mathbf{z}_T$ and the final predicted latent $\mathbf{z}_0$, respectively.

The principle of this work is based on the concept *"first position the objects, then refine the results"*. This is reflected in the overall Directed Diffusion pipeline shown in Fig. 3 and also the applications detailed in the scene compositing

Figure 2: (Left): Four evenly selected steps from a SD denoising process. Note that the position of the cat is evident early in the process denoted with red box, however the details that define it as a cat are not yet clear. (Right): The cross-attention maps associated with the word "cat" in the prompt, seen at the beginning and end of the denoising steps shown on the left.

and placement finetuning sections. Reflecting this principle, DD's computation can be divided into two stages:

Attention Editing. This stage focuses on spatially editing the cross-attention map used for conditioning in stable diffusion. It operates during the diffusion steps $t \in [T, T-N)$ that establish the object location, where N is a hyperparameter determining the number of steps in this stage. We chose $N{=}10$ in most of our experiments. As described later, this stage modifies the cross-attention map during the first N denoising steps by amplifying the region inside $\mathcal{B}$ while downweighting the surrounding areas through optimization, denoted in the yellow region in Fig. 3.

Conventional SD Denoising. Following the attention editing stage, this stage runs the standard SD process using classifier-free guidance (Ho and Salimans 2021) over the remainder of the reverse diffusion denoising steps $t \in [T{-}N, 0]$. Note that the only difference between the two stages is the cross-attention editing, denoted in the red region in Fig. 3.

Cross-Attention Map Guidance

To achieve our goal, DD asks the user specific information about the "direction" of the object with $\mathcal{R} = \{\mathcal{B}, \mathcal{I}\}$ to guide the denoising SD process, where $\mathcal{B}$ and $\mathcal{I}$ denote the bounding box specification and the directed prompt words indices in $\mathcal{P}$, respectively. We will use the prompt *"A bear watching a flying bird"* as our specific example throughout this section and the next section.

Specifically, a bounding box $\mathcal{B}{=}\big\{(x,y) \,|\, b_{\text{left}}{\times}w \le x \le b_{\text{right}}{\times}w,\ b_{\text{top}}{\times}h \le y \le b_{\text{bottom}}{\times}h\big\}$ is the set of all pixel coordinates inside $\mathcal{B}$ of resolution $w \times h$. The bounding box with label i guides the location of the subject indicated by the ith prompt word. The exact location and shape of the subject results from the interaction of the Gaussian activation window inside this box with the activation $\mathbf{z}_i$ in the U-net, hence the object is not restricted in shape and may extend somewhat outside the box. In our implementation, $\mathcal{B}$ is generated based on a tuple of four scalars representing the boundary of the bounding box, denoted as $\mathbf{b} = (b_{\text{left}}, b_{\text{right}}, b_{\text{top}}, b_{\text{bottom}})$, $b_*{\in}[0,1]$, describing the bounding box of the directed object position expressed as fractions of the image size. The dimensions of the $\mathcal{B}$ are scaled in proportion to the resolution of the U-net representation.

SD implements text-guided synthesis with classifier-free guidance making use of cross-attention between the projected embedding $\mathbf{Q}$ of the SD latent $\mathbf{z}_t$ and the projected embeddings $\mathbf{K}$ of the $|\mathcal{W}|$ potential words from the prompt. The cross-attention $\mathbf{A}{=}\text{softmax}(\mathbf{Q}\mathbf{K}^T)/\sqrt{d} \in \mathbb{R}^{n^2 \times |\mathcal{W}|}$ is applied with the standard attention mechanism to reweight the text embeddings $\mathbf{V}$, however this matrix can be interpreted as $|\mathcal{W}|$ individual cross-attention maps $\mathbf{A}^{(i)}$. SD uses CLIP (Radford et al. 2021) as its text encoder, which has $|\mathcal{W}| = 77$, while $n = 64$, matching the latent spatial dimension in the current SD implementation. We denote the prompt words for an object to be directed with an indices set $\mathcal{I} \subset \{i | i \in \mathbb{N}, 1 \le i \le |\mathcal{P}|\}$. The cross-attention maps $\mathbf{A}^{(i)}$, where $i \in \mathcal{I}$, are divided into a subset of *prompt maps* corresponding to words in the prompt, and the remainder $i \in \mathcal{T}$ where $\mathcal{T} \coloneqq \{|\mathcal{P}|{+}1, \cdots, |\mathcal{W}|\}$ that do not correspond to any words but are computed to allow constant-size matrix computations. We term the latter as *trailing attention maps*.

We wish to edit the cross attention maps corresponding to the "directed" words in the prompt so as to place the object at the bounding box. In one initial experiment, we simply injected activation in the bounding box region of the cross attention using a 2D Gausssian fall-off. This was not stable (Ma et al. 2023), which might be explained by the fact that it is potentially pushing $\mathbf{z}_t$ outside the range of values encountered during training. A better strategy is to use the network itself to project the solution near the "trained manifold", by expressing an objective at time t in terms of coarser guidance made at time $t{+}1$ that is refined during the $t{+}1{\to}t$ denoising step. Another observation is that we can separate the desired coarse guidance from the directed cross attention maps by making use of the trailing maps to express the coarse guidance, since these are included in the $\mathbf{Q}\mathbf{K}^T$ product.

Given $\mathcal{B}$, we generate a modified cross-attention map using two functions `weaken-mask` and `strengthen-mask`, denoted as $\mathbf{W}(\cdot)$ and $\mathbf{S}(\cdot)$:

$$\mathbf{W}(\mathcal{B}')_{xy} = \begin{cases} c, & (x,y) \in \mathcal{B}' \\ 1., & \text{otherwise,} \end{cases}$$

$$\mathbf{S}(\mathcal{B})_{xy} = \begin{cases} f(x,y), & (x,y) \in \mathcal{B} \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{B}'$ is the complement of $\mathcal{B}$, and $f(\cdot)$ denotes the function that "injects attention" to amplify the region $\mathcal{B}$. In our implementation, we use a Gaussian window of size $\sigma_x = b_w/2, \sigma_y = b_h/2$ to generate the corresponding weight, where $b_w = \text{ceil}((b_{\text{right}}{-}b_{\text{left}}){\times}w), b_h = \text{ceil}((b_{\text{top}}{-}b_{\text{bottom}}) \times h)$ are the width and the height of $\mathcal{B}$.

The $\mathbf{W}(\cdot)$ and $\mathbf{S}(\cdot)$ functions "direct" selected subjects from the prompt toward specific locations of the image in the SD denoising process. $\mathbf{S}(\cdot)$ inserts higher activation into the provided bounding box with a Gaussian window, while $\mathbf{W}(\cdot)$ attenuates the region outside $\mathcal{B}$ by multiplying by $c{<}1$. The resulting *target* maps $\mathbf{D}^{(i)}$ are calculated in Algo. 1 line 5. They are not used directly in the cross-attention denoising guidance. Instead, as described next, $\mathbf{D}^{(i)}\ \forall i \in \mathcal{I}$ provide a target for the loss in Eq. 1, while $\mathbf{D}^{(i)}\ \forall i \in \mathcal{T}$ are unweighted sources for the weighted trailing maps created in Algo. 1 line 7.

SD implements denoising with a U-net architecture, where text guidance $\tau_\theta(y(\mathcal{P}))$ from the prompt is passed

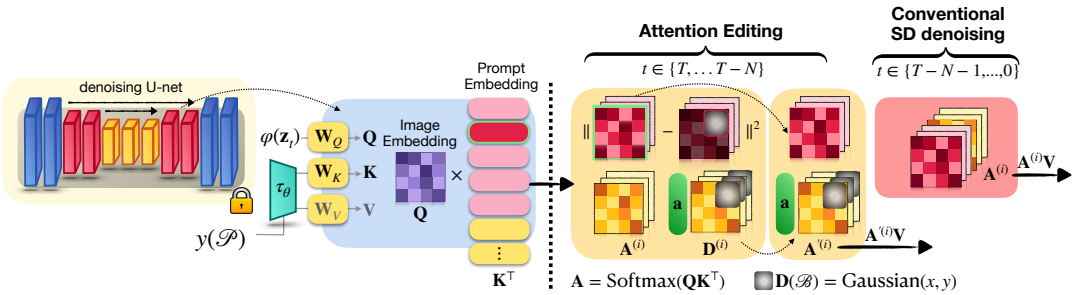

Figure 3: Directed Diffusion (DD) pipeline overview: DD divides the denoising process into initial steps where *Attention Editing* is performed, followed by refinement steps using *Conventional SD Denoising*. The goal of the *Attention Editing* stage is to optimize the cross attention map for a directed word (red, outlined in green) to approximately match a target $\mathbf{D}$ in which neural "activation" has been injected with a Gaussian fall-off inside a bounding box specified by the user. The optimization is performed by adjusting a vector ($\mathbf{a}$, green) that re-weights the trailing attention maps. Note that the optimization objective involves two denoising timesteps. Please see the text for details.

through the tokenizer $y(\cdot)$ and text embedding $\tau_\theta(\cdot)$ to layers of the U-net through cross-attention layers. In each of these cross attention layers, DD clones and modifies selected cross-attention maps $\mathbf{A}^{(i)}, \forall i \in \mathcal{I} \cup \mathcal{T}$ to form the target maps $\mathbf{D}$ in Algo. 1 line 7 for the optimization objective.

Algorithm 1: DD Cross-Attention Editing Algorithm

1: **procedure** DDCrossAttnEdit(DM$(\cdot)$, $\mathcal{P}$, $\mathcal{B}$)
2: **for** $l \in$ layer(DM$(\mathbf{z}_t, \mathcal{P})$) **do**
3: **if** type$(l) \in$ CrossAttn **then**
4: $\mathbf{A} = \text{Softmax}(\mathbf{Q}_l(\mathbf{z}_t) \cdot \mathbf{K}_l(\mathcal{P})^T)$
5: $\mathbf{D}^{(i)} \leftarrow \mathbf{A}^{(i)} \odot \mathbf{W}(\mathcal{B}') + \mathbf{S}(\mathcal{B}) \quad \forall i \in \mathcal{T} \cup \mathcal{I}$
6: $\mathbf{a}^* := \arg\min_{\mathbf{a}} \mathcal{L}_{\mathbf{a}}$
7: $\mathbf{A}'^{(|\mathcal{P}|+1:77)} := \mathbf{D}^{(|\mathcal{P}|+1:77)} \odot \mathbf{a}^*$
8: $\mathbf{z}_t \leftarrow l(\mathbf{z}_t, \mathbf{A}' \cdot \mathbf{V}_l(\mathcal{P}))$
9: **else**
10: $\mathbf{z}_t \leftarrow l(\mathbf{z}_t)$

Our optimization objective seeks to find the best weighed combination of the trailing maps at time t, using a weight vector $\mathbf{a}^* = \arg\min \mathcal{L}_{\mathbf{a}_t} \in \mathbb{R}^{77-|\mathcal{P}|-1}$, such that the "directed" prompt maps $\mathbf{A}^{(i)}, i \in \mathcal{I}$ best match the corresponding target maps $\mathbf{D}^{(i)}$ (with some abuse of notation):

$$\mathcal{L}_{\mathbf{a}_t} = \sum_i \|\mathbf{A}^{(i)}_{t-1}\left(\mathbf{A}^{(|\mathcal{P}|+1:77)}_t \odot \mathbf{a}_t\right) - \mathbf{D}^{(i)}\|^2, \qquad (1)$$

$\forall t \in \{T, ..., T-N\}$, where $\odot$ denotes multiplying each trailing cross-attention map $\mathbf{A}^{(i)}_t$ by the corresponding element of the vector $\mathbf{a}_t$, and the notation $\mathbf{A}^{(i)}_{t-1}(\cdot)$ indicates an (indirect) *functional* dependence of the cross-attention map at time $t-1$ on the weighted sum of the trailing maps edited at time t. After the optimized weights $\mathbf{a}^*$ are obtained, they are used to produce re-weighted trailing maps $\mathbf{A}'$ in Algo. 1 line 7, which then influence the overall cross-attention conditioning at time t in Algo. 1 line 8.

To summarize the dependency flow, the trailing maps at time t are edited, and are reweighted by the optimized $\mathbf{a}^*$. These then influence the denoising at time t, which in turn influences the cross attention map at time $t-1$. This dependency structure expressing the loss at time $t-1$ is necessary because at time t the loss does not involve the weights $\mathbf{a}$, resulting in a zero gradient. In addition we believe that influencing the desired cross-attention through an intermediate denoising step helps keep the solution closer to states seen in training and thus removes the instability we found in early experiments. Please see the supplementary (Ma et al. 2023) for further implementation details.

Application: Scene Compositing As mentioned earlier, prompts involving several objects often fail in T2I models, especially those based on CLIP. While our DD pipeline supports the direction of multiple objects by means of multiple bounding boxes $\mathcal{B}$, in practice, the results are unreliable when the number of bounding boxes is more than two. We resolve this problem by additional editing operations.

Specifically, we first use DD to individually generate multiple objects, and record the latent information $\mathbf{z}^{(r)}_t$ of all steps for the directed objects $r \in \{1, \cdots, R\}$, where R is the total number of directed objects. Then, inspired by (Liu et al. 2022), we linearly interpolate between the $\mathbf{z}^{(r)}_t$ and the latent $\mathbf{z}_t$ conditioned on the original prompt $\mathcal{P}$,

$$\mathbf{z}_t(x, y) := \frac{1}{R}\sum_r w_r\, \mathbf{z}_t(x, y) + (1 - w_r)\, \mathbf{z}^{(r)}_t(x, y) \qquad (2)$$

where $\forall t \in \{T, ..., T-N\}$, $\forall (x, y) \in \mathcal{B}_r$, and $w_r \in \mathbb{R}$ is a given weight, generally set to 0.1 in our experiments. After $N \approx 10$ steps of applying Eq. 2, SD is used to refine the result and produce contextual interactions.

Application: Placement Finetuning In some cases the artist may wish to experiment with different object positions after obtaining a desirable image. However, when the object's bounding box is moved DD typically generates a somewhat different instance of the desired object. Our placement finetuning (PF) method addresses this problem, allow-

ing the artist to immediately experiment with different positions for an object while keeping its identity, and without requiring any model fine-tuning or other optimization (Gal et al. 2022; Ruiz et al. 2022). Note that the PF method is not intended to produce a sequence of images for a video, as that would generally require further control over 3D object pose, camera viewpoint, etc. Please refer to our supplementary (Ma et al. 2023) for the diagram of PF.

The algorithm has three components: First, a mask $\mathbf{M}_o$ for the directed object is obtained by thresholding the final $t{=}0$ cross-attention and clipping by the bounding box $\mathcal{B}$. At a selected time step $T{-}N$ the transformation $X(\cdot)$ is applied to the recorded latent and to the mask $\mathbf{M}_o$, producing a translated mask $X(\mathbf{M}_o)$. A mask $\neg\mathbf{M}_o$ is computed as the complement of $\mathbf{M}_o$ and is used to extract the background region for an initial placement-finetuned latent $\mathbf{z}'_{T-N}$.

The second component inpaints holes in the background caused by moving the forground object. These hole regions are initialized with the transformed latent $X(\mathbf{z}_T)$ masked by $\mathbf{M}_o$. In our experiments the transformation $X(\cdot)$ is translation implemented with `torch.roll`, which causes content that is translated beyond the tensor dimension to wrap around and re-appear on the opposite side. This has the effect of initializing the hole that would otherwise appear on the opposite side with somewhat reasonable values, however more sophisticated inpainting schemes are possible here. After these initializations we perform a single iteration of adding noise followed by denoising, in order to cause the inpainted areas to have distinct detail and blend with the background.

Lastly, there are $t \in \{T{-}N{-}1, \cdots, 0\}$ steps in which the transformed latent from the current step is composited with the background,

$$\mathbf{z}'_t := \mathbf{z}'_t \odot \neg X(\mathbf{M}_o) + X(\mathbf{z}_t) \odot X(\mathbf{M}_o),$$

followed by denoising, resulting in a coherent refined image. N is generally set at 10 in our experiments (Ma et al. 2023). Large N better preserves the original foreground and background, while smaller N encourages more interaction between the foreground and background.

Experiments and Comparisons

We now present several example results of DD. Please see (Ma et al. 2023) for additional results and comparisons. Table 1 gives quantitative CLIP similarities between the embeddings of the prompts and the synthesized images. Our results are similar to or better than the compared methods.

As described earlier, unmodified T2I methods such as stable diffusion are fallible and using such a system is *not* simply a matter of typing a text prompt. A prompt such as *"A dog chasing a ball"* may fail to generate the ball, generate a dog without legs, etc. This poses challenges for reporting qualitative evaluations: while there is no room to show a sufficient range of randomly chosen results, evaluating a *single* randomly-chosen example often results in a failure, and is not representative of how current generative AI tools are used. Users do not simply stop when they receive a bad result, but instead do trial-and-error exploration over a number of random seeds, as well as the prompt and hyperparameters

Eval Type	SD	GLIGEN	CD	DD
SceneComp	0.821	0.802	0.791	**0.824**
OneMask	0.834	**0.810**	-	0.807
TwoMasks	0.842	0.802	-	**0.851**

Table 1: CLIP scores (Hessel et al. 2021; Radford et al. 2021) of the three experimental categories in the Fig. 4 and Fig. 5. See the text for details.

(Smith 2022). To emulate this, we adopt a **SS@k** protocol, in which k images are generated with consecutive seeds following an initial randomly chosen seed, and the best-of-k image is subjectively selected.

Comparison: Scene Composition As mentioned earlier, synthesizing and controlling several objects in an image is a challenge for many T2I methods. Common problems including missing objects, incorrectly colored objects, and incorrectly positioned objects. Fig. 4 (Right) showcases our results alongside results obtained from the recommended settings of the Composable Diffusion (CD) (Liu et al. 2022) and the public implementations of BOXDIFF (Xie et al. 2023) and GLIGEN (Li et al. 2023). The comparison clearly reveals that directed objects, such as the cherry blossoms, church, and trees, exhibit greater realism and detail in our approach compared to CD. Additionally, our results demonstrate better fidelity to the prompt (pond is dark) and less information bleeding between objects (e.g., the church and mountain are not pink).

Figure 4: One object (left) and scene composition (right) comparison: SS@12 results from BLD, CD, GLIGEN, BOXDIFF, and our DD. The bounding box is shown in green (except for CD, which does not support position guidance). Please enlarge to see details including the bounding box. BLD images are reproduced from (Avrahami, Fried, and Lischinski 2022). Prompt used from the left to right: "The [bonfire] next to a man", "A [gravestone] next to a man in red shirt", "A [white horse] in front of the erupting volcano", "The [cherry blossoms] next to the lake and a mountain", "A [white church] under lightning in the pink sky", and "[Mystical trees] next to a dark magical pond".

Comparison: One Object Methods for placing an object in another image can suffer from inconsistency between the object and the background environment. In Fig. 4 (Left), we compare our DD result with Blended Latent Diffusion (BLD) (Avrahami, Fried, and Lischinski 2022), a text-driven editing method that guides object placement using using a mask. DD uses the bounding box $\mathcal{B}$ to direct the object, which can be seen as analogous to a simple form of mask. In this comparison, note that the results shown for BLD are reproduced from the original paper, whereas we guessed a prompt to generate roughly similar images. In our results it is evident that DD generates strong and realistic interactions between the directed object and the background, including the man touching the gravestone, the shadows cast on the grass, the occlusion of the volcano by the horse, and the man's hand that catches on fire :) We also emphasize that BLD addresses a different (and difficult) purpose of editing real images. The comparison here is intended simply to highlight the realistic interactions between the directed object and the background that are obtained using our method.

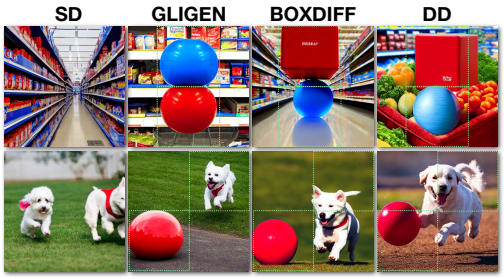

Figure 5: SS@12 results from SD, GLIGEN, BOXDIFF, and DD. The prompts are (first row) "A [red cube] above a [blue sphere] in the supermarket", (second row) "A [white running dog] chasing after a [red ball]." The directed object is denoted in bold and its bounding box is shown in green. Please enlarge to see details including the bounding box.

Figure 6: Comparison between DD-PF (Placement finetuning) and DD. From left to right, the first row shows the DD result based on the sliding bounding box associated with castle. The second row is the PF alternative. The third row shows using PF to reposition the dog, bird, and sphere.

Comparison: Two Objects Fig. 5 compares results of several methods in placing two objects. A core consideration in this task is scene consistency and "correlation" between the directed objects. Our results show natural interactions such as the running dog touching the red ball, and the cube and sphere supported naturally in a shopping basket.

Placement finetuning Fig. 6 shows results of the placement finetuning method. Fig. 6 (top) shows a sliding bounding box associated with the word "castle". Using PF (middle), the castle identity is preserved, while the surrounding environment such as the trees and lake reflection are reasonably synthesized. The bottom row shows the reconstruction of the latent image before and after PF. Each subject is successfully re-positioned while maintaining coherent interaction with the background (E.g., reflection, waves).

Figure 7: SS@12 results of the prompt "a [white dog] S* a [red ball]", where the token S* is replaced by *"and"* or *"chasing"* in the first row and second row, respectively. Note the result of each column share the same random seed.

Ablation: prompt conjunction We assess the efficacy of our scene compositing technique by replacing a conjunction with a transitive verb in the prompt, thus entailing a relation between the directed objects. Building on Fig. 5, Fig. 7 showcases the DD synthesis of two prompts: "A white dog *and* a red ball" and "A white dog *chasing* a red ball." The dog does appear to be chasing the ball in the second row, however, achieving such action relationships in general is not always possible due to CLIP's limited understanding of grammar (Feng et al. 2022) and other SD synthesis errors.

Limitations and Conclusion

Storytelling with images requires directing the *placement* of important objects in each image. Our algorithm, directed diffusion, is a step toward this goal. The algorithm is simple to implement, requiring only a few lines of modification of a widely used library.[4] Directed Diffusion inherits imitations of stable diffusion, including the need for trial-and-error exploration mentioned earlier. Some random seeds fail to produce the desired subject or produce distorted objects such as animals with missing limbs. In common with (Meng et al. 2022; Liew et al. 2022), it is necessary to specify the number of steps over which editing is active. Significant additional advances will be needed before video storytelling is possible. However, our method may be sufficient for the creation of storybooks, comic books, etc. when used in conjunction with other existing tools. Please see (Ma et al. 2023) for implementation details and further experiments and discussion.

[4]https://github.com/huggingface/diffusers

Acknowledgements

We thank Jason Baldridge for helpful feedback.

References

Arijon, D. 1976. *Grammar of the Film Language*. Focal Press.

Avrahami, O.; Fried, O.; and Lischinski, D. 2022. Blended Latent Diffusion. *CoRR*, abs/2206.02779.

Avrahami, O.; Hayes, T.; Gafni, O.; Gupta, S.; Taigman, Y.; Parikh, D.; Lischinski, D.; Fried, O.; and Yin, X. 2022. SpaText: Spatio-Textual Representation for Controllable Image Generation. *CoRR*, abs/2211.14305.

Balaji, Y.; Nah, S.; Huang, X.; Vahdat, A.; Song, J.; Kreis, K.; Aittala, M.; Aila, T.; Laine, S.; Catanzaro, B.; Karras, T.; and Liu, M. 2022. eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers. *CoRR*, abs/2211.01324.

Bar-Tal, O.; Yariv, L.; Lipman, Y.; and Dekel, T. 2023. MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation. *CoRR*, abs/2302.08113.

Chang, H.; Zhang, H.; Barber, J.; Maschinot, A.; Lezama, J.; Jiang, L.; Yang, M.; Murphy, K.; Freeman, W. T.; Rubinstein, M.; Li, Y.; and Krishnan, D. 2023. Muse: Text-To-Image Generation via Masked Generative Transformers. *CoRR*, abs/2301.00704.

Feng, W.; He, X.; Fu, T.; Jampani, V.; Akula, A. R.; Narayana, P.; Basu, S.; Wang, X. E.; and Wang, W. Y. 2022. Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis. *CoRR*, abs/2212.05032.

Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D. 2022. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. *CoRR*, abs/2208.01618.

Hessel, J.; Holtzman, A.; Forbes, M.; Bras, R. L.; and Choi, Y. 2021. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In *EMNLP*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33.

Ho, J.; and Salimans, T. 2021. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Hyvärinen, A. 2005. Estimation of Non-Normalized Statistical Models by Score Matching. *J. Mach. Learning Research*, 6: 695–709.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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. *CoRR*, abs/2301.07093.

Liew, J. H.; Yan, H.; Zhou, D.; and Feng, J. 2022. MagicMix: Semantic Mixing with Diffusion Models. *CoRR*, abs/2210.16056.

Liu, N.; Li, S.; Du, Y.; Torralba, A.; and Tenenbaum, J. B. 2022. Compositional Visual Generation with Composable Diffusion Models. In *ECCV*.

Ma, W.-D. K.; Lewis, J. P.; Lahiri, A.; Leung, T.; and Kleijn, W. B. 2023. Directed Diffusion: Direct Control of Object Placement through Attention Guidance. arXiv:2302.13153.

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.; and Ermon, S. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *Int. Conf. on Learning Representations (ICLR)*.

Nichol, A. Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *ICML*.

Park, D. H.; Luo, G.; Toste, C.; Azadi, S.; Liu, X.; Karalashvili, M.; Rohrbach, A.; and Darrell, T. 2022. Shape-Guided Diffusion with Inside-Outside Attention.

Patashnik, O.; Wu, Z.; Shechtman, E.; Cohen-Or, D.; and Lischinski, D. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. ICML*.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. *CoRR*, abs/2204.06125.

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 (CVPR)*, 10684–10695.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2022. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. *CoRR*, abs/2208.12242.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S. K. S.; Ayan, B. K.; Mahdavi, S. S.; Lopes, R. G.; Salimans, T.; Ho, J.; Fleet, D. J.; and Norouzi, M. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *CoRR*, abs/2205.11487.

Smith, E. 2022. A Traveler's Guide to the Latent Space.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, volume 32.

StabilityAI. 2023. Stable Diffusion 2.1 Demo. https://huggingface.co/spaces/stabilityai/stable-diffusion. Accessed: 2024-01-01.

Thomas, F.; and Johnston, O. 1981. *Disney animation : the illusion of life*. Abbeville Press New York, 1st ed. edition. ISBN 0896592332 0896592324.

Weng, L. 2021. What are diffusion models?

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. *CoRR*, abs/2307.10816.

Yu, J.; Xu, Y.; Koh, J. Y.; Luong, T.; Baid, G.; Wang, Z.; Vasudevan, V.; Ku, A.; Yang, Y.; Ayan, B. K.; Hutchinson, B.; Han, W.; Parekh, Z.; Li, X.; Zhang, H.; Baldridge, J.; and Wu, Y. 2022. Parti: Pathways Autoregressive Text-to-Image model.

Zhang, L.; and Agrawala, M. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. arXiv:2302.05543.