Universal Guidance for Diffusion Models

Arpit Bansal*[1], Hong-Min Chu*[1], Avi Schwarzschild[1],Soumyadip Sengupta[2],
Micah Goldblum[3], Jonas Geiping[1], Tom Goldstein[1]
[1]University of Maryland, [2]University of North Carolina at Chapel Hill,
[3]New York University

Abstract

Typical diffusion models are trained to accept a particular form of conditioning, most commonly text, and cannot be conditioned on other modalities without retraining. In this work, we propose a universal guidance algorithm that enables diffusion models to be controlled by arbitrary guidance modalities without the need to retrain any use-specific components. We show that our algorithm successfully generates quality images with guidance functions including segmentation, face recognition, object detection, and classifier signals. Code is available at github.com/arpitbansal297/Universal-Guided-Diffusion.

1. Introduction

Diffusion models are powerful tools for creating digital art and graphics. Much of their success stems from our ability to carefully control their outputs, customizing results for each user's individual needs. Most models today are controlled through *conditioning*. With conditioning, the diffusion model is built from the ground up to accept a particular modality of input from the user, be it descriptive text, segmentation maps, class labels, etc. While conditioning is a powerful tool, it results in models that are handcuffed to a single conditioning modality. If another modality is required, a new model needs to be trained, often from scratch. Unfortunately, the high cost of training makes this prohibitive for most users.

A more flexible approach to controlling model outputs is to use *guidance*. In this approach, the diffusion model acts as a generic image generator, and is not required to understand a user's instructions. The user pairs this model with a guidance function that measures whether some criterion has been met. For example, one could guide the model to minimize the CLIP score between the generated image and a text description of the user's choice. During each iteration of image creation, the iterates are nudged down the gradient of the guidance function, causing the final generated image to satisfy the user's criterion.

In this paper, we study guidance methods that enable any off-the-shelf model or loss function to be used as guidance for diffusion. Because guidance functions can be used without re-training or modification, this form of guidance is *universal* in that it enables a diffusion model to be adapted for nearly any purpose.

From a user perspective, guidance is superior to conditioning, as a *single* diffusion network is treated like a foundational model that provides universal coverage across many use cases, both commonplace and bespoke. Unfortunately, it is widely believed that this approach is infeasible. While early diffusion models relied on classifier guidance [6], the community quickly turned to classifier-free schemes [9] that require a model to be trained from scratch on class labels with a particular frozen ontology that cannot be changed [2, 18, 22].

The difficulty of using guidance stems from the domain shift between the noisy images used by the diffusion sampling process and the clean images on which the guidance models are trained. When this gap is closed, guidance can be performed successfully. For example, [18] successfully use a CLIP model as guidance, but only after re-training CLIP from scratch using noisy inputs. Noisy retraining closes the domain gap, but at a very high financial and engineering cost. To avoid the additional cost, we study methods for closing this gap by changing the sampling scheme, rather than the model.

To this end, our contributions are summarized as follows:

- We propose an algorithm that enables universal guidance for diffusion models. Our proposed sampler evaluates the guidance models only on denoised images, rather than noisy latent states. By doing so, we close the domain gap that has plagued standard guidance methods. This strategy provides the end-user with the flexibility to work with a wide range of guidance modalities and even multiple modalities simultaneously. The underlying diffusion model remains fixed and no fine-tuning of any kind is necessary.

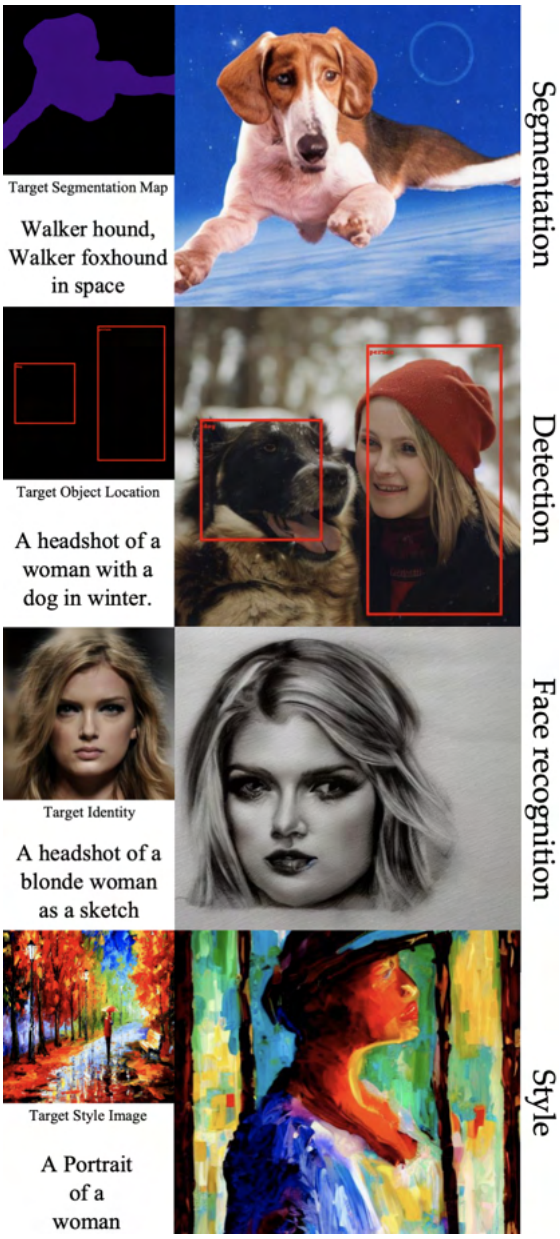

Figure 1. Diffusion guided by off-the-shelf networks.

- We demonstrate the effectiveness of our approach for a variety of different constraints such as *classifier labels*, *human identities*, *segmentation maps*, *annotations from object detectors*, and constraints arising from *inverse linear problems*.

2. Background

We first briefly review the recent literature on the core framework behind diffusion models. Then, we define the problem setting of controlled image generation and discuss previous related works.

2.1. Diffusion Models

Diffusion models are strong generative models that proved powerful even when first introduced for image generation [8, 25]. The approach has been successfully extended to a number of domains, such as audio and text generation [1, 11, 13, 14].

We introduce (unconditional) diffusion formally, as it is helpful in describing the nuances of different types of models. A diffusion model is defined as a combination of a T-step forward process and a T-step reverse process. Conceptually, the forward process gradually adds Gaussian noise of different magnitudes to a clean data point z_0, while the reverse process attempts to gradually denoise a noisy input in hopes of recovering a clean data point. More concretely, given an array of scalars representing noise scales $\{\alpha_t\}_{t=1}^T$ and an initial, clean data point z_0, applying t steps of the forward process to z_0 yields a noisy data point

$$z_t = \sqrt{\alpha_t} z_0 + (\sqrt{1-\alpha_t})\epsilon,\ \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{1}$$

A diffusion model is a learned denoising network ϵ_θ. It is trained so that for any pair (z_0, t) and any sample of ϵ,

$$\epsilon_\theta(z_t, t) \approx \epsilon = \frac{z_t - \sqrt{\alpha_t} z_0}{\sqrt{1-\alpha_t}}. \tag{2}$$

The reverse process takes the form $q(z_{t-1}|z_t, z_0)$ with various detail definitions, where $q(\cdot|\cdot)$ is generally parameterized as a Gaussian distribution. Different works also studied different approximations of the unknown $q(z_{t-1}|z_t, z_0)$ used to perform sampling. For example, denoising diffusion implicit model (DDIM) [24] first computed a *predicted* clean data point

$$\hat{z}_0 = \frac{z_t - (\sqrt{1-\alpha_t})\epsilon_\theta(z_t, t)}{\sqrt{\alpha_t}}, \tag{3}$$

and sample z_{t-1} from $q(z_{t-1}|z_t, \hat{z}_0)$ by replacing unknown z_0 with $\hat{z}_0$. On the other hand, while the details of individual sampling methods vary, all sampling methods produce z_{t-1} based on current sample z_t, current time step t and a predicted noise $\hat{\epsilon}$. To ease the notation burden, we define a function $S(\cdot, \cdot, \cdot)$ as an abstraction of the sampling method, where $z_{t-1} = S(z_t, \hat{\epsilon}, t)$.

2.2. Controlled Image Generation

In this paper, we focus on controlled image generation with various constraints. Consider a differentiable guidance function f, for example a CLIP feature extractor or a segmentation network. When applied to an image, we obtain a vector $c = f(x)$. We also consider a function $\ell(\cdot, \cdot)$ that measures the closeness of two vectors c and c'. Given a particular choice of c, which we call a *prompt*, the corresponding constraint (based on c, ℓ, and f) is formalized as

$\ell(c, f(z)) \approx 0$, and we aim to generate a sample z from the image distribution satisfying the constraint. In plain words, we want to generate an in-distribution image that matches the prompt.

Prior work that studied controlled generative diffusion mainly falls into two categories. We refer to the first category as conditional image generation, and the second category as guided image generation. Next, we discuss the characteristics of each category and better situate our work among existing methods.

Conditional Image Generation. Methods from this category require training new diffusion models that accept the prompt as an additional input [2, 9, 18, 27, 29]. For example, [9] proposed classifier-free guidance using class labels as prompts, and trained a diffusion model by linear interpolation between unconditional and conditional outputs of the denoising networks. [2] studied the case where the guidance function is a known linear degradation operator, and trained a conditional model to solve linear inverse problems. [18] further extended classifier-free guidance to text-conditional image generation with descriptive phrases as prompts, and trained a diffusion model to enforce the similarity between the CLIP [20] representations of the generated images and the text prompts. These methods are successful across different types of constraints, however the requirement to retrain the diffusion model makes them computationally intensive.

Guided Image Generation. Works in this category employed a frozen pre-trained diffusion model as a foundation model, but modify the sampling method to guide the image generation with feedback from the guidance function. Our method falls into this category. Prior work that studied guided image generation did so with a variety of restrictions and external guidance functions [3, 4, 6, 7, 12, 16, 28]. For example, [6] proposed classifier guidance, where they trained a classifier on images of different noise scales as the guidance function f, and included gradients of the classifier during the sampling process. However, a classifier for noisy images is domain-specific and generally not readily available – an issue our method circumvents. [28] assumed the external guidance functions to be linear operators, and generated the component of images residing in the null space of linear operators with the foundation model. Unfortunately, extending that method to handle non-linear guidance functions is non-trivial. [3] studied general guidance functions, and modified the sampling process with the gradient of guidance function calculated on the expected denoised images. Nevertheless, the authors only presented results with simpler non-linear guidance functions such as non-linear blurring.

In this work, we study universal guidance algorithms for guided image generation with diffusion models using any off-the-shelf guidance functions f, such as object detection or segmentation networks.

3. Universal Guidance

We propose a guidance algorithm that augments the image sampling method of a diffusion model to include guidance from an off-the-shelf auxiliary network. Our algorithm is motivated by an empirical observation that the reconstructed clean image $\hat{z}_0$ obtained by Eq. (3), while naturally imperfect, is still appropriate for a generic guidance function to provide informative feedback to guide the image generation. In Sec. 3.1, we motivate our *forward universal guidance* by extending classifier guidance [6] to leverage this observation and handle generic guidance functions. In Sec. 3.2, we propose a supplementary *backward universal guidance* to help enforce the generated image to satisfy the constraint based on the guidance function f. In Sec. 3.3, we discuss a simple yet helpful self-recurrence trick to empirically improve the fidelity of generated images.

3.1. Forward Universal Guidance

To guide the generation with information from the external guidance function f and the loss function ℓ, an immediate thought is to extend classifier guidance [6] to accept any general guidance function. Concretely, given a class prompt c, classifier guidance performs classification-guided sampling by replacing $\epsilon_\theta(z_t, t)$ in each sampling step $S(z_t, t)$ with

$$\hat{\epsilon}_\theta(z_t, t) = \epsilon_\theta(z_t, t) - \sqrt{1-\alpha_t}\nabla_{z_t} \log p(c|z_t). \quad (4)$$

Defining $\ell_{ce}(\cdot, \cdot)$ to be the cross-entropy loss and f_{cl} to be the guidance function that outputs classification probability, Eq. (4) can be re-writtern as

$$\hat{\epsilon}_\theta(z_t, t) = \epsilon_\theta(z_t, t) + \sqrt{1-\alpha_t}\nabla_{z_t} \ell_{ce}(c, f_{cl}(z_t)). \quad (5)$$

However, directly replacing f_{cl} and ℓ_{ce} with any off-the-shelf guidance and loss functions does not work in practice, as f is most likely trained on clean images and fails to provide meaningful guidance when the input is noisy.

To address the issue, we leverage the fact that $\epsilon_\theta(z_t, t)$ predicts the noise added to the data point, and we can therefore obtain a *predicted* clean image $\hat{z}_0$ by Eq. (3). We propose to instead calculate the guidance based on the predicted clean data point as

$$\hat{\epsilon}_\theta(z_t, t) = \epsilon_\theta(z_t, t) + s(t) \cdot \nabla_{z_t} \ell(c, f(\hat{z}_0)) \quad (6)$$

where $s(t)$ controls the guidance strength for each sampling step and

$$\nabla_{z_t} \ell(c, f(\hat{z}_0)) = \nabla_{z_t} \ell\left(c, f\left(\frac{z_t - \sqrt{1-\alpha_t}\epsilon_\theta(z_t, t)}{\sqrt{\alpha_t}}\right)\right)$$

as in Eq. (3). We term Eq. (6) forward universal guidance, or forward guidance in short. In practice, applying forward

Algorithm 1 Universal Guidance

Parameter: Recurrent steps k, gradient steps m for backward guidance and guidance strength $s(t)$,
Required: z_T sampled from $\mathcal{N}(0, I)$, diffusion model ϵ_θ, noise scales $\{\alpha_t\}_{t=1}^T$, guidance function f, loss function ℓ, and prompt c
for $t = T, T-1, \ldots, 1$ **do**
  for $n = 1, 2, \ldots, k$ **do**
    Calculate $\hat{z}_0$ as Eq. (3)
    Calculate $\hat{\epsilon}_\theta$ using forward universal guidance as Eq. (6)
    if $m > 0$ **then**
      Calculate Δz_0 by minimizing Eq. (7) with m steps of gradient descent
      Perform backward universal guidance by $\hat{\epsilon}_\theta \leftarrow \hat{\epsilon}_\theta - \sqrt{\alpha_t/(1-\alpha_t)}\Delta z_0$ (Eq. (9))
    end if
    $z_{t-1} \leftarrow S(z_t, \hat{\epsilon}_\theta, t)$
    $\epsilon' \sim \mathcal{N}(0, I)$
    $z_t \leftarrow \sqrt{\alpha_t/\alpha_{t-1}} z_{t-1} + \sqrt{1-\alpha_t/\alpha_{t-1}}\epsilon'$
  end for
end for

guidance effectively brings the generated image closer to the prompt while keeping the generation trajectory in the data manifold. We note that a related approach is also studied in [3], where the guidance step is computed based on $E[z_0|z_t]$. The approach drew inspiration from the score-based generative framework [26], but resulted in a different update method.

3.2. Backward Universal Guidance

As will be shown in Sec. 4.2, we observe that forward guidance sometimes over-prioritizes maintaining the "realness" of the image, resulting in an unsatisfactory match with the given prompt. Simply increasing the guidance strength $s(t)$ is suboptimal, as this often results in instability as the image moves off the manifold faster than the denoiser can correct it.

To address the issue, we propose backward universal guidance, or backward guidance in short, to supplement forward guidance and help enforce the generated image to satisfy the constraint. The key idea of backward guidance is to optimize for a clean image that best matches the prompt based on $\hat{z}_0$, and linearly translate the guided change back to the noisy image space at step t. Concretely, instead of directly calculating $\nabla_{z_t}\ell(c, f(\hat{z}_0))$, we compute a guided change Δz_0 in clean data space as

$$\Delta z_0 = \arg\min_{\Delta} \ell(c, f(\hat{z}_0 + \Delta)). \tag{7}$$

Empirically, we solve Eq. (7) with m-step gradient descent, where we use $\Delta = 0$ as a starting point. Since $\hat{z}_0 + \Delta z_0$ minimizes $\ell(c, f(z))$ directly, Δz_0 is the change in clean data space that best enforces the constraint. Then, we translate Δz_0 back to the noisy data space of z_t by calculating the *guided denoising prediction* $\tilde{\epsilon}$ that satisfies

$$z_t = \sqrt{\alpha_t}(\hat{z}_0 + \Delta z_0) + \sqrt{1-\alpha_t}\tilde{\epsilon}. \tag{8}$$

Reusing Eq. (3), we can rewrite $\tilde{\epsilon}$ as an augmentation to the original denoising prediction $\epsilon_\theta(z_t, t)$ by

$$\tilde{\epsilon} = \epsilon_\theta(z_t, t) - \sqrt{\alpha_t/(1-\alpha_t)}\Delta z_0. \tag{9}$$

Comparing to forward guidance, backward guidance (as Eq. (9)) produces an optimized direction for the generated image to match the given prompt, and hence prioritizes enforcing the constraint. Furthermore, calculation of a gradient step for Eq. (7) is computationally cheaper than forward guidance (Eq. (6)), and we can therefore afford to solve Eq. (7) with multiple gradient steps, further improving the match with the given prompt.

We note that the names "forward" and "backward" are used analogously to the forward and backward Euler methods.

Figure 2. An example of how self-recurrence helps segmentation-guided generation. The left-most figure is the given segmentation map, and the images generated with recurrence steps of 1, 4 and 10 follow in order.

3.3. Per-step Self-recurrence

Unfortunately, when we apply our universal guidance to standard generation pipelines, we often find images with artifacts and strange behaviors that clearly separate them from natural images. Similar observations have been made in [16, 28], where linear guidance functions are studied. Our attempts to prioritize realness by decreasing $s(t)$ proved ineffective; the sweet spot that both ensures the realness and guidance constraint satisfaction doesn't always exist, especially for complex guidance functions. We conjecture that the guidance direction produced by our universal method is not always related to the realness of the images when the guidance function creates too much information loss, causing the image to stray from the natural image sampling trajectory.

Inspired by [16, 28], we address the issue by applying per-step self-recurrence. More concretely, after $z_{t-1} = S(z_t, \hat{\epsilon}_t, t)$ is sampled, we re-inject random Gaussian noise

$\epsilon' \sim \mathcal{N}(0, \mathbf{I})$ to z_{t-1} to obtain z'_t by

$$z'_t = \sqrt{\alpha_t/\alpha_{t-1}} \cdot z_{t-1} + \sqrt{1 - \alpha_t/\alpha_{t-1}} \cdot \epsilon'. \quad (10)$$

Eq. (10) ensures z'_t to have proper noise scale for input at time step t. We repeat the self-recurrence k times before continuing the sampling for step $t-1$. Intuitively, the self-recurrence allows exploration of different regions of the data manifold at the same noise scale, allowing more budget to find a solution that satisfies both guidance and image quality. Empirically, we find that our self-recurrence can keep the realness of the generated image with a proper guidance strength $s(t)$ that ensures the match with the given prompt. We illustrate an example of how self-recurrence improves the harmony of generated images in Fig. 2.

We summarize our universal guidance algorithm composed of forward universal guidance, backward universal guidance and per-step self-recurrence in Algorithm 1. For simplicity, the algorithm assumes only one guidance function, but can be easily adapted to handle multiple pair of (f, l). Additionally, the objectives of the forward and backward guidance do not have to be identical, allowing different ways to simultaneously utilize multiple guidance functions.

4. Experiments

In this section, we present results testing our proposed universal guidance algorithm against a wide variety of guidance functions. Specifically, we experiment with Stable Diffusion [22], a diffusion model that is able to perform text-conditional generation by accepting text prompt as additional input, and experiment with a purely unconditional diffusion model trained on ImageNet [5], where we use pre-trained model provided by OpenAI [6]. We note that Stable Diffusion, while being a text-conditional generative model, can also perform unconditional image generation by simply using an empty string for the text prompt. We first present the experiment on Stable Diffusion for different guidance functions in Sec. 4.1, and present the results on ImageNet diffusion model in Sec. 4.2.

4.1. Results for Stable Diffusion

In this section, we present the results of guided image generation using Stable Diffusion as the foundation model. The guidance functions we experiment with include the CLIP feature extractcor [20], a segmentation network, a face recognition network and an object detection network. For experiments on Stable Diffusion, we discover that applying forward guidance already produce high-quality images that match the given prompt, and hence set $m = 0$. To perform forward guidance on Stable Diffusion, we forward the predicted clean latent variable computed by Eq. (3) through the image decoder of Stable Diffusion to obtain predicted clean images. We discuss the results and implementation details for each guidance function in its corresponding subsection.

Figure 3. We compare the ability to match given text prompts between our universal guidance algorithm and text-conditional model trained from scratch. The results demonstrate that our universal algorithm is comparable to specialized conditional model on the ability to generate quality images that satisfy the text constraints.

CLIP Guidance. CLIP [20] is a state-of-the-art text-to-image similarity model developed by OpenAI. To apply our algorithm to text-guided image generation, we use the image feature extractor of CLIP as the guidance function. We construct a loss function that calculates the negative cosine similarity between an image embedding and the CLIP text embedding produced by a given text prompt. We use $s(t) = 10\sqrt{1-\alpha_t}$ and $k = 8$ and use Stable Diffusion as an unconditional image generator.

We generate images guided by a number of text prompts. To further assess our universal guidance algorithm and compare guidance and conditioning, we also generate images using classical, text-conditional generation by Stable Diffusion with identical prompts as inputs, and summarize the results in Fig. 3. The results in Fig. 3 show that our algorithm can guide the generation to produce high-quality images that match the given text description, and are comparable with images generated by the specialized text-conditioning model.

Segmentation Map Guidance. To perform guided image generation using a segmentation map as prompt, we use a MobileNetV3-Large [10] with a segmentation head, and a publicly available pre-trained model in PyTorch [19]. As the segmentation network outputs per-pixel classification

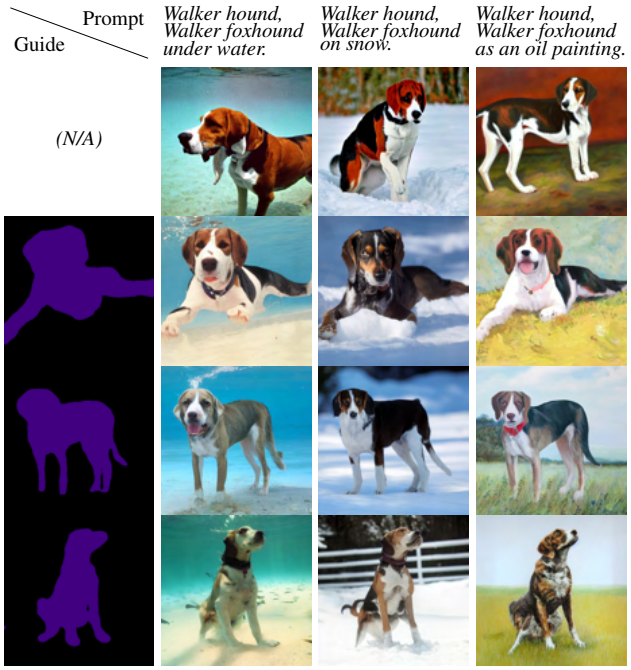

Figure 4. In addition to matching the text prompts (above each column), these images are guided by an image segmentation pipeline. Each column contains examples of images generated to match the prompt and the segmentation map in the left-most column. The top-most row contains examples generated without guidance.

Figure 5. In addition to matching the text prompts (above each column), these images are guided by a facial recognition system. Each column contains examples of images generated to match the prompt and the identity of the images in the left-most column. The top-most row contains examples generated without guidance.

probability, we construct a loss function ℓ as the sum of per-pixel cross-entropy loss between a given prompt and the predicted segmentation of generated images. We set $s(t) = 400 \cdot \sqrt{1 - \alpha_t}$ and $k = 10$.

In our experiment, we combine segmentation maps that depict objects of different shapes with new text prompts. We use the text prompt as a fixed additional input to Stable Diffusion to perform text-conditional sampling, and guide the text-conditional generated images to match the given segmentation maps. Results are presented in Fig. 4. From Fig. 4, we see that the generated images show a clear separation between object and background that matches the given segmentation map nearly perfectly. The generated object and background also each match their descriptive text (i.e. dog breed and environment description). Furthermore, the generated images are overall highly realistic.

Face Recognition Guidance. To guide image generation to resemble the face of a given person, we compose a guidance function that combines a face detection module and a face recognition module. This setup produces a facial attribute embedding from an input face image. We use multi-task cascaded convolutional networks (MTCNN) [30] as the face detection module, and use facenet [23] as the face recognition module. The guidance function f hence crops out the detected face and outputs a facial attribute embedding as prompt, while we use l_1-loss between embedding as the loss function ℓ. We note that to compute the guidance direction in our algorithm, we only backpropagate through the facenet and treat the face cropping mask produced by MTCNN as an oracle input, as MTCNN utilizes non-maximum suppression [17] which is non-differentiable. Here we set $s(t) = 20000 \cdot \sqrt{1 - \alpha_t}$ and and $k = 2$.

We explore different combinations of face guidance and text prompts. Similarly to the segmentation case, we use the text prompt as a fixed additional conditioning to Stable Diffusion and guide this text-conditional trajectory with our algorithm so that the face in the generated image looks similar to the face prompt. In Fig. 5, we clearly see that the facial characteristics of a given face prompt are reproduced almost perfectly on the generated images. The descriptive text of either background, material, or style is also realized correctly and blends nicely with the generated faces.

Object Location Guidance For Stable Diffusion, we also present the results guiding image generation with an object detection network. For this experiment, we use Faster-RCNN [21] with Resnet-50-FPN backbone [15], a publicly

Figure 6. In addition to matching the text prompts (above each column), these images are guided by an object detector. Each column contains examples of images generated to match the prompt and the bounding boxes used for guidance. The top row contains examples generated without guidance.

Figure 7. In addition to matching the text prompts (above each column), these images are guided by a style image. Each column contains examples of images generated to match the text prompt and the style image used for guidance. The top-most row contains examples generated without style guidance.

available pre-trained model in Pytorch, as our object detector. We use bounding boxes with class labels as our object location prompt. We construct a loss function ℓ by the sum of three individual losses, namely (1) anchor classification loss, (2) bounding box regression loss and (3) region label classification loss, where (1) and (2) are computed on the region proposal head while (3) is computed on the region classification head. We note that, compared to standard R-CNN training, we drop the additional bounding box alignment loss on region classification head. We found that our loss construction helps to produce objects of correct categories for each location prompt. We set $s(t) = 100 \cdot \sqrt{1-\alpha_t}$ and $k = 3$.

We again experiment with different combinations of text prompt and object location prompt, and similarly use the text prompt as a fixed conditioning to Stable Diffusion. Using our proposed guidance algorithm, we perform guided image generation that generates and matches the objects presented in the text prompt to the given object locations. The results are presented in Fig. 6. We observe from Fig. 6 that objects in the descriptive text all appear in the designated location with the appropriate size indicated by the given bounding boxes. Each location is filled with appropriate, high-quality generations that align with varied image content prompts, ranging from "beach" to "oil painting".

Style Guidance Finally, we conclude our experiments on Stable Diffusion by guiding the image generation based on a reference style given by a style image. To achieve so, we capture the reference style from the style image by the image feature extractor from CLIP, and use the resulting image embedding as prompts. The loss function calculates the negative cosine similarity between the embedding of generated images and the embedding of the style image. Similar to previous experiments, we control the content using text input as additional conditioning to the Stable Diffusion model. We experiment with combinations of different style images and different text prompts, and present the results in Fig. 7. From Fig. 7, we can see that the generated images contain contents that match the given text prompts, while exhibiting style that matches the given style images. In this experiment we set $s(t) = 6 \cdot \sqrt{1-\alpha_t}$ and $k = 6$. Furthermore, in order to control the amount of content we set the scale γ, a parameter of Stable Diffusion that balances the text-conditional generation and unconditional generation, as 3.0, 3.0, and 4.0 respectively for each column.

4.2. Results for ImageNet Diffusion

In this section, we present results for guided image generation using an unconditional diffusion model trained on

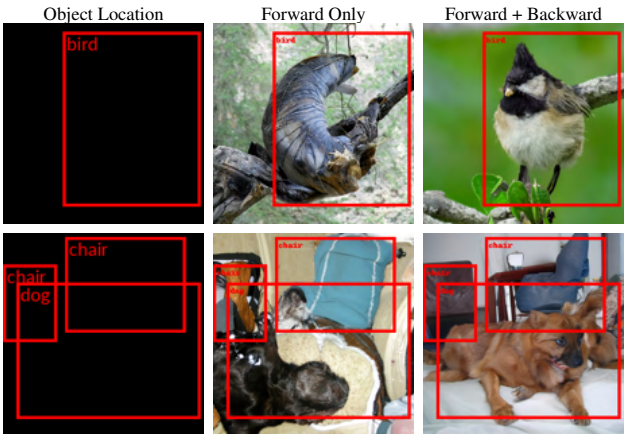

Figure 8. Generation guided by object detection with the unconditional ImageNet model. Images generated with both forward and backward guidance are realistic and have the desired objects in the designated locations. In contrast, images generated using only forward guidance exhibit objects of the incorrect category or with inaccurate position/size.

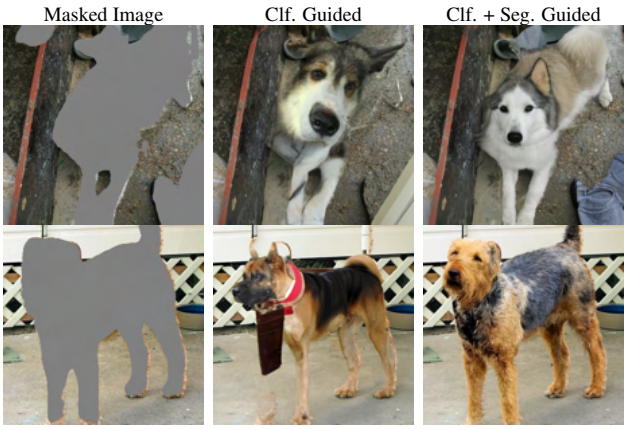

Figure 9. Our guidance algorithm can incorporate feedback from multiple guidance functions. The first column shows the prompt for inpainting. The second column shows classifier-guided inpainting, where dog images with close matches to inpainting prompt are generated. The third column shows images generated with both classifier and segmentation guidance, where realistic dogs are generated exactly on the masked regions. The results show that our algorithm handles multiple guidance functions effectively.

ImageNet. We experiment with object location guidance and a hybrid guided image generation task which we term segmentation-guided inpainting. We also include additional experiments where we use CLIP guidance in the appendix. We will discuss results of each guidance separately.

Object Location Guidance. Similar to object location guidance for Stable Diffusion, we also use the same network architecture and the same pre-trained model as our object detection network, and construct an identical loss function ℓ for our guidance algorithm. However, unlike Stable Diffusion, object locations are the only prompts available for guided image generation. For this experiment, we use $s(t) = 100\sqrt{1-\alpha_t}$ and $k = 3$. We experiment with different object location prompts using either (1) only forward universal guidance and (2) both forward and backward universal guidance. We observe from Fig. 10 that applying both forward and backward guidance generates images that are realistic and the objects matches the prompt nicely. On the other hand, while images generated using only forward guidance remain realistic, they feature objects with mismatching categories and locations. The results demonstrate the effectiveness of our universal guidance algorithm, and also validate the necessity of our backward guidance.

Segmentation-Guided Inpainting. In this experiment, we aim to explore the ability of our algorithm to handle multiple guidance functions. We perform guided image generation with combined guidance from an inpainting mask, a classifier and a segmentation network. We first generate images with masked regions as the prompt for inpainting. We then pick an object class c as the prompt for classification and generate a segmentation mask where the masked regions are considered foreground objects of the same class c. We use ℓ_2 loss on the non-masked region as the loss function for inpainting, and set the corresponding $s(t) = 0$, or equivalently only use backward guidance for inpainting. We use the same segmentation network as described in Sec. 4.1 with $s(t) = 200\sqrt{1-\alpha_t}$. For classification guidance, we use the classifier that accepts noisy input [6], and perform the original classifier guidance Eq. (4) instead of our forward guidance. The results summarized in Fig. 9 show that when using both inpainting and classifier as guidance, our algorithm generates realistic images that both match the inpainting prompt and can be classified correctly to the given object class. Adding in segmentation guidance, our algorithm further improves the generated images with a near-perfect match to both the segmentation map and inpainting prompt while maintaining realism. This demonstrates that our algorithm can effectively combine the feedback from individual guidance functions.

5. Conclusion

In this paper, we propose a universal guidance algorithm that is able to perform guided image generation with any off-the-shelf guidance function based on a fixed foundation diffusion model. Our algorithm only requires guidance and loss functions to be differentiable, and avoids any retraining to adapt either the guidance function or the foundation model to a specific type of prompt. We demonstrate promising results with our algorithm on complex guidance including segmentation, face recognition and object detection systems. Even multiple guidance functions can be combined and used in conjunction.

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. 2

[2] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022. 1, 3

[3] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 3, 4

[4] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *arXiv preprint arXiv:2206.00941*, 2022. 3

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[6] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. volume 34, 2021. 1, 3, 5, 8

[7] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *arXiv preprint arXiv:2206.09012*, 2022. 3

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 32, 2020. 2

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1, 3

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 5

[11] Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*, 2022. 2

[12] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022. 3

[13] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 2

[14] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022. 2

[15] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021. 6

[16] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 3, 4

[17] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006. 6

[18] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1, 3

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 5

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 6

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of CVPR*, 2022. 1, 5

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 6

[24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *International Conference on Learning Representations*, 2021. 2

[25] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[26] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021. 4

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

[28] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022. 3, 4

[29] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16293–16303, 2022. 3

[30] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016. 6