End-to-End Diffusion Latent Optimization Improves Classifier Guidance

Bram Wallace
Salesforce AI
b.wallace@salesforce.com

Akash Gokul
Salesforce AI
agokul@salesforce.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Nikhil Naik
Salesforce AI
nnaik@salesforce.com

Abstract

Classifier guidance—using the gradients of an image classifier to steer the generations of a diffusion model—has the potential to dramatically expand the creative control over image generation and editing. However, currently classifier guidance requires either training new noise-aware models to obtain accurate gradients or using a one-step denoising approximation of the final generation, which leads to misaligned gradients and sub-optimal control. We highlight this approximation’s shortcomings and propose a novel guidance method: Direct Optimization of Diffusion Latents (DOODL), which enables plug-and-play guidance by optimizing diffusion latents w.r.t. the gradients of a pre-trained classifier on the true generated pixels, using an invertible diffusion process to achieve memory-efficient backpropagation. Showcasing the potential of more precise guidance, DOODL outperforms one-step classifier guidance on computational and human evaluation metrics across different forms of guidance: using CLIP guidance to improve generations of complex prompts from DrawBench, using fine-grained visual classifiers to expand the vocabulary of Stable Diffusion, enabling image-conditioned generation with a CLIP visual encoder, and improving image aesthetics using an aesthetic scoring network.

1. Introduction

Text-conditioned denoising diffusion models (DDMs), such as Latent/Stable Diffusion, Imagen and numerous others, have been widely adopted for synthesizing realistic im-
ages given an input text prompt [35, 36, 38, 34, 2]. The conditioning setup requires DDMs to be trained using paired data of images and the conditioning modality—image-caption pairs in the case of text conditioning. Once trained, the DDM can be steered to generate images using the conditioning modality. However, the conditioning paradigm constrains the image generation capabilities of a trained DDM model; a text-conditioned diffusion model cannot readily utilize other modalities such as depth maps or image classifiers or audio as conditioning signals. While DDMs conditioned on multiple modalities have been trained [30, 36] a “plug-and-play” approach that allows for a pretrained DDM to be guided by any external function that determines whether some generation criterion is satisfied is desirable.

In principle, classifier guidance [43, 8] enables this capability in DDMs. Classifier guidance, named so since it was first demonstrated using pretrained image classification models, combines the score estimate of the diffusion model with the gradient of the image classifier to steer the generation process to produce images that correspond to a particular class. Any differentiable loss function can be used for classifier guidance. In addition to class-conditional generation, it has been shown to improve compositionality using cross-modal guidance [24]. There are two existing ways of incorporating classifier guidance. In the first approach, we train a noise-aware classifier that can be used to compute an accurate gradient w.r.t. an intermediate generation step [43, 8]. This approach requires re-training the classifier model, which can be computationally expensive/ineffable due to lack of access to training data. In the second approach, at a time step \(t\), we denoise the image with a single application of the DDM and compute the gradient using this approximately denoised image [24]. This one-step approximation is necessitated by the prohibitive memory requirements of computing a gradient w.r.t. the latents through the entire diffusion process containing many steps. Since this approach only obtains gradients using an one-step denoising approximation of the final generation, the approximately denoised images are often misaligned with the final generations which classifier guidance is aiming to modify (Figure 2), leading to sub-optimal guidance signal.

To enable flexible and exact model guidance, without noise-aware classifiers or approximations, we propose Direct Optimization Of Diffusion Latents (DOODL). DOODL optimizes the initial diffusion noise vectors w.r.t. a model-based loss on images generated from the full-chain diffusion process. We leverage EDICT, a recently developed drop-in discretely invertible diffusion algorithm [46], which admits backpropagation with constant memory cost w.r.t the number of diffusion steps, to compute classifier gradients on the pixels of the final generation w.r.t. the original noise vectors. This enables efficient iterative optimization of diffusion latents w.r.t. any differentiable loss on the image pixels and accurate calculation of gradients for classifier guidance.

We demonstrate the efficacy of DOODL through a diverse set of guidance signals (Figure 1) using quantitative and human evaluation studies. First, we show that CLIP classifier guidance using DOODL improves generation of images guided by text prompts from the DrawBench [38] dataset, which test compositionality and the ability to guide using unusual captions. Second, we show the ability to expand the vocabulary of a pretrained Stable Diffusion model using fine-grained visual classifiers; a capability that one-step classifier guidance does not have. Third, we demonstrate that DOODL can be used for personalized entity generation (e.g. “A dog in sunglasses”), with zero retraining of any new network—a first to our knowledge. Finally, we utilize DOODL to perform a novel task; increasing the perceived aesthetic quality of generated/real images. We hope that DOODL can enable and inspire diverse plug-and-play capabilities for pretrained diffusion models.

2. Related Work

2.1. Text-to-Image Diffusion Models

Text-to-image diffusion models [17, 43, 41] such as GLIDE [28], DALLE-2 [34], Imagen [38, 16], Latent Diffusion [35, 36], and eDiffi [2] have recently emerged at the forefront of image generation, based on methods in non-equilibrium thermodynamics [39]. Classifier guidance [43, 8], and its adoptions [28, 24], use the gradients of pretrained classifier models to guide such generations. Instead of sequential denoising, [43] traverses at a fixed noise level before each denoising step. Concurrent work [3] modifies classifier guidance to refine the gradient prediction at each noise level before continuing. DiffuseIT [23] is another work that uses pretrained recognition models in conjunction with DDMs. Specifically, it performs either text- or image-guided image translation, leveraging a DINO ViT [5] model to enforce structural similarity of a DDM generation to a source image via a contrastive loss and a CLIP model to steer the image semantics towards the desired tar-
Table 1: Learning-Based methods require data and training while Guidance-Based methods require pretrained recognition networks (trained on non-noisy data in our setting).

get. DiffuseIT calculates updates at incremental noise levels as opposed to DOODL which optimizes $x_t$ based off of $x_0$. Table 1 shows the requirements for learned methods such as ControlNet [48] vs. these guidance-based methods. The former require data and training, while the latter substitute this need for pretrained recognition models. Directly optimizing latent variables of other generative models such as GANs or VAEs [11, 32, 29] w.r.t. a pixel-based loss has been shown to generate in-distribution images that focus solely on the applications to image editing and do not consider properties of Invertible Neural Networks (INNs) or similar processes. Methods such as DDIM are much faster than non-invertible equivalents (Section 2.2).

### 2.2. Invertible Neural Networks (INNs)

While neural networks tend to be non-dimensionality-preserving functions, there has been prior work on constructing reversible architectures. A predominant class of such INNs are normalizing flow models [9, 10, 22]. A modified version of the “coupling layers” in normalizing flow architectures are incorporated into the EDICT [46] algorithm employed in this work. [4] proposes an architecture guaranteed to be invertible via well-conditioned inverse problems instead of a closed-form solution. The memory savings of such architectures has been used in long-sequence recurrent neural networks [26] and to study inverse problems [1].

### 3. Background

#### 3.1. Invertible Neural Networks w.r.t Memory

When using gradient descent for optimizing neural networks, with network parameters $\Xi = \{\xi_p\}_{p=1}^{P}$, network input $x$, network output $y = f(x)$, and loss function $c$, the derivative $\frac{dc(y)}{d\xi}$ is calculated and gradient descent performed to minimize $E_{Data}c(y) = E_{Data}(f(x))$. Here $f$ is implicitly conditioned on $\Xi$. Consider $f$ as the composition of $n$ functions (layers) $f^n \circ f^{n-1} \circ \ldots \circ f^1$. To optimize $\xi$, a parameter of the $t^th$ layer $f^t$ the derivative $\frac{dc(y)}{d\xi}$ is calculated. Denote $f^k \circ f^{k-1} \circ \ldots \circ f^1 = F^k$. Since $y = F^n(x)$ the derivative w.r.t. $\xi$ can be calculated using the chain rule:

$$
\frac{dc(y)}{d\xi} = \frac{dc(F^n(x))}{d\xi} = \frac{dc(F^n(x))}{dF^n_1(x)} \cdot \frac{dF^n_1(x)}{dF^n_{1-1}(x)} \cdot \ldots \cdot \frac{dF^n_{1-1}(x)}{dF^n_{1-2}(x)} \cdot \ldots \cdot \frac{dF^n_1(x)}{dx}
$$

In the general case, calculating $\frac{dc(y)}{d\xi}$ requires storing all intermediate activations, a key bottleneck to backpropagation. Network sharding [33] across processors reduces the per-processor hardware memory requirement but the total remains the same. Gradient checkpointing [13, 49] decreases memory cost, but increases runtime linearly w.r.t. saved memory. INNs can recover intermediate states/inputs from the output, reducing memory costs by avoiding activation caching. If every $f^j$ is invertible in Equation (1), the denominator terms can be reconstructed during the backwards pass; such methods have been leveraged to train large INNs much faster than non-invertible equivalents (Section 2.2).

#### 3.2. Denoising Diffusion Models (DDMs)

Image DDMS are trained to predict the noise $\epsilon$ added to an image $x$ [40, 17, 8, 18, 42]. Noise levels are discretized into a set $T = \{0, 1, \ldots, T\}$ that index a noising schedule $\{\alpha_t\}_{t=0}^{T}$, $\alpha_0 = 1$. $t \in T$ are randomly sampled during training and paired with data $x^{(i)}$ (images or autoencoded representations) to generate noisy samples

$$
x^{(i)}_t = \sqrt{\alpha_t} x^{(i)} + \sqrt{1-\alpha_t} \epsilon
$$

where $\epsilon \sim N(0, I)$. DDMs conditioned on the timestep $t$ and auxiliary information (e.g., image caption) $C$ are trained to approximate the added noise $DDM(x_t^{(i)}, t, C) \approx \epsilon$. At generation time, an $x_T \sim N(0, 1)$ is drawn and the DDM is iteratively applied to hallucinate a real image from the noise. Following the DDIM [40] sampling model, the final generation $x_0$ is equal to the composition of $S$ denoising functions: applications of $\Theta$ conditioned on $C$ and varying timesteps $t$. Denoting $\Theta(x, t, C)$ as $\Theta(t, C) \circ x$,

$$
x_0 = [\Theta_{0, C} \circ \Theta_{1, C} \circ \ldots \circ \Theta_{T, C}] (x_T)
$$

#### 3.2.1 Classifier Guidance

In addition to $C$, other guidance signals can steer generation to objectives. The foremost example, “classifier
guidance” incorporates gradients of a loss ($c_{clf}$) from a classifier network $\Phi$) on estimated pixels into the noise prediction [43, 8]. From a theoretical perspective, this typically is the gradient of the log-conditional probability $\nabla \log p_{\Phi}(y|x_t)$ There are two primary ways of incorporating classifier guidance:

1. A noise-aware classifier is trained for direct use on intermediate (noisy) $x_t$, with $\nabla x_t c_{clf}(x_t)$ incorporated into the denoising prediction [28]. Training noise-aware models is effective but often infeasible due to computational expense and data availability (e.g. medical images). This results in there being very few publicly available noise-aware models.

2. $x_0$ is approximated with a single application of $\Theta(t,c)$ as in the training objective [24]. Here, the incorporated gradient is $\nabla x_t c_{clf}(x_0^0)$, where $x_0^0$ is a 1-step approximate solve by substituting $\epsilon$ for $\Theta(t,c)$ in Equation (3). While a standard (noise-unaware) model can be used, the gradients are calculated w.r.t. an approximation of $x_0$ (Figure 2) and as such may be misaligned with the true derivative of $\frac{d c_{clf}(x_t)}{dx_0}$.

### 3.2.2 Exact Inversion of the Diffusion Process

Recently, EDICT [46], an exactly invertible variant of the discrete (time-stepped) diffusion process was proposed. EDICT operates on a latent pair $(x_t, y_t)$ instead of a single variable. Initially $x_T = y_T \sim N(0, I)$, followed by iteratively denoising using the reverse diffusion process:

$$
\begin{align*}
    x_t^{\text{inter}} &= a_t \cdot x_t + b_t \cdot \Theta(t,c)(y_t) \\
    y_t^{\text{inter}} &= a_t \cdot y_t + b_t \cdot \Theta(t,c)(x_t^{\text{inter}}) \\
    x_{t-1} &= p \cdot x_t^{\text{inter}} + (1 - p) \cdot y_t^{\text{inter}} \\
    y_{t-1} &= p \cdot y_t^{\text{inter}} + (1 - p) \cdot x_{t-1}
\end{align*}
$$

(5)

where $(a_t, b_t)$ are time-dependent coefficients, and $p \in [0, 1]$ is a mixing parameter to mitigate latent drift. Intuitively, this process first updates the $x$ and $y$ sequences based on the current state of the counterpart, and then invertibly “averages” them together. The above equations admit linear solves to invert them, defining the inverse process:

$$
\begin{align*}
    y_{t+1}^{\text{inter}} &= (y_t - (1 - p) \cdot x_t)/p \\
    x_{t+1}^{\text{inter}} &= (x_t - (1 - p) \cdot y_{t+1}^{\text{inter}})/p \\
    y_{t+1} &= (y_{t+1}^{\text{inter}} - b_{t+1} \cdot \Theta(t+1,c)(x_{t+1}^{\text{inter}}))/a_{t+1} \\
    x_{t+1} &= (x_{t+1}^{\text{inter}} - b_{t+1} \cdot \Theta(t+1,c)(y_{t+1}))/a_{t+1}
\end{align*}
$$

(6)

We employ this construction in DOODL, and use Equation (6) to encode images $x_0$ to latents $x_T$ in Section 5.3.

### 4. Direct Optimization of Diffusion Latents

We aim to develop a method that overcomes the shortcomings of classifier guidance discussed in Section 3.2.1. Concretely, a method that (1) does not require retraining/finetuning an existing pretrained classifier model, (2) computes gradients w.r.t. the true output instead of a one-step approximation, and (3) incorporates the guidance in a semantically meaningful way (as opposed to an adversarial-style perturbation). We emphasize the last point in particular. As shown in the literature on adversarial attacks [44, 47, 14], gradients w.r.t. pixels can satisfy a classifier loss while not perceptually changing the content of an image. This is as opposed to techniques such as latent optimization in GANs, where the regularization provided by the decoder means that optimization happens in a space where perturbations typically result in perceptually meaningful changes that satisfy the desired objective. In this work, we aim to directly optimize diffusion latents, a first in the literature to our knowledge.

As shown in Equation (4), it is mathematically trivial to optimize $x_T$ for a desired outcome on $x_0$. There is a closed form expression for $\frac{dx_T}{dx_0}$ as in Equation (1). However, due to activation caching, the naive memory cost is linear in the number of DDIM sampling steps due to the $T$ applications of $\Theta$. With a typical value of $S = 50$, this memory cost nears a terabyte for state-of-the-art diffusion models which is impractical for most uses. Gradient checkpointing trades memory for computational complexity, resulting in the computational complexity of each backwards pass increasing by a factor of $S$ if memory costs are held constant.

We draw inspiration from INNs (Section 3.1) to optimize $x_T$ w.r.t. criteria on $x_0$ in feasible runtime. Using invertible $\Theta(t,c)$ in Equation (4), intermediate states can be reconstructed during the backwards pass using only a constant number of applications of $\Theta$ w.r.t. $T$, circumventing the prohibitive memory cost without sacrificing runtime.

We turn to the recently developed EDICT [46] as an invertible reverse diffusion process that admits constant-memory implementation of optimization of $x_t$. Given conditioning $C$, differentiable model-based cost function $c$ and a latent draw $x_T^{(0)}$, the EDICT generative process is performed (50 steps, $p = 0.93$, StableDiffusion v1.4) yielding initial output $f(x_T^{(0)}) = x_0^{(0)}$, which is then used to calculate a loss $c(x_0^{(0)})$ and corresponding gradient $\nabla x_t c(f(x_T^{(0)}))$. This gradient can then be used to perform a step of gradient descent optimization on $x_T^{(0)}$.

We modify vanilla gradient descent in several key ways to achieve realistic images that satisfy the guidance criteria. After each optimization step, the EDICT “fully noise” latent pair $x_t^{(0)}$ and $y_t^{(0)}$ (from Equations (5) and (6)) are averaged together and renormalized to the original norm of the initial draw $x_T^{(0)}$. The averaging prevents latent drift which degrades quality, as noted in [46]. Normalizing to the original norm keeps the latent on the “gaussian shell” and in-distribution for our diffusion model.
We note that DOODL can be theoretically grounded by considering it as Langevin dynamics. In the general form of Langevin dynamics, the movement of a particle in a space with energy potential $E$ (particle moves high to low potential) is described by the update rule:

$$x_{t+1} = x_t - \nabla E \Delta t + \epsilon \Delta t$$  \hspace{1cm} (7)

where $\epsilon$ is a normal noise Brownian motion term. The update rule of DOODL,

$$g_t = 0.9 \cdot g_{t-1} + Clip(-\lambda \cdot \nabla x_T L)$$  \hspace{1cm} (8)

$$x_T^{(i+1)} = x_T + g_i + \mathcal{N}(0, 10^{-4} \cdot I)$$  \hspace{1cm} (9)

is sum of the previous position, velocity (with momentum, induced by energy potential $L$), and drift term.

5. Applications and Results

Previous work has used dataset- or webly-supervised classifiers such as CLIP [32, 28]. We consider these as well as previously unexplored settings. The optimization learning rate, $\lambda$, is a key hyperparameter and we note its values throughout. The focus of our work is on how the gradients from non-noise-aware classifiers can be better incorporated into the diffusion process. As such, we do not consider noise-aware classifier methods [28]. For larger images and other example results, please refer to the Supplementary.

5.1. Reinforcing Text Guidance

Classifier guidance using a CLIP text-image similarity loss has been used to reinforce the text conditioning $C$ pro-
Algorithm 1 DOODL

Input: $\lambda$ (Learning rate), $C$ (Model conditioning), $c$ (Cost function), $T$ (# Diffusion steps), $m$ (# Optimization steps), $\Theta$ (Diffusion Model)

Initialize $x_T^{(0)} = N(0, I)$. 

$g_{-1} = 0$

for $i = 0$ to $m$

$x_T, y_T = x_T^{(i)}$

for $t = T - 1$ to 0

$x_t^{(i+1)} = \frac{x_t^{(i)} + \lambda \cdot \nabla x_t \cdot L \text{ Linear in } T \text{ b/c of EDICT}}{1 + \lambda \cdot \nabla x_t \cdot L \text{ Linear in } T \text{ b/c of EDICT}}$

$y_t = \text{Clip}(g_t, -10^{-3}, 10^{-3})$

$g_t = 0.9 \cdot g_{t-1} + g_i$

$x_T^{(i+1)} = x_T + g_t + N(0, 10^{-4} \cdot I)$

end for

yield $x_T^{(m)}$

We provide as input to a text-to-image diffusion model [28]. Previous work [24] shows that classifier guidance with gradients calculated on 1-step approximations of $x_0$ enhance the ability to generate complex prompts. This setting serves as an initial validation of our proposed method.

We employ a spherical distance loss from the literature:

$$d(x, y) = 2\lambda \sin^{-1}\left(\frac{|x - y|}{2}\right)$$

(10)

computing a distance between the embedding of $C$ by the CLIP text encoder, $CLIP_{text}(C)$, and the embedding of $x_0$ by the image encoder, $CLIP_{image}(x_0)$. $\lambda \in [10, 100]$ is effective for the baseline one-step guidance, we use $\lambda = 30$. For DOODL, we find $\lambda = 0.1$ to be preferable. We attribute the lower value to the gradient magnitudes increasing when computed through repeated applications of the network, as well as the incorporation of momentum.

We test on the DrawBench benchmark [38], a collection containing 11 categories of text prompts designed to probe different aspects of text-to-image generation. In particular, we evaluate on the DALLE and Reddit categories of DrawBench, which focus on compositionality and highly implausible scenes, being especially suited for testing fidelity to conditioning text. Image-text alignment is measured both automatically (CLIP score) and with human evaluation in Figure 5. Experiments are performed across 9 seeds for all method-prompt pairs. We use a LAION-trained CLIP model [20] for classifier guidance and an OpenAI-trained CLIP model [32] for CLIP score evaluation. While there is an overlap in the training sets of these models; we find them sufficiently independent to provide automated validation. Generated samples are shown in Figure 4.

DOODL relatively improves on baseline classifier guidance’s CLIP score by 3.1% and 2.6% on DALLE and Reddit respectively. For human study, we define prompt alignment success rate (PASR) as the percent of generations rated as humans to be aligned with the prompt in head-to-head comparisons (see Figure 5 caption). The PASR of DOODL is highest, notably increasing on the complex unusual prompts of Reddit by 11.3% as compared to vanilla Stable Diffusion, with a 6% increase over one-step classifier guidance.
Figure 5: DrawBench results. **Left:** CLIP Scores. **Right:** Human evaluation. 5 labelers are given a pairwise comparison of DOODL to another method for the same prompt-seed inputs and asked which represents the prompt better, with options for equal success or failure. We display the fraction of time for each method that it is classified by the majority of labelers as “Better” or “Both Achieve”. Value for DOODL averaged across two comparison runs.

![Figure 5](image.png)

Table 1: 5 labelers are given a pairwise comparison of DOODL to another method for the same prompt-seed inputs and asked which represents the prompt better, with options for equal success or failure. We display the fraction of time for each method that it is classified by the majority of labelers as “Better” or “Both Achieve”. Value for DOODL averaged across two comparison runs.

<table>
<thead>
<tr>
<th>Method</th>
<th>CLIP Score</th>
<th>% Win Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>DALLE Reddit</td>
<td>26</td>
<td>28</td>
</tr>
<tr>
<td>DALLE Reddit</td>
<td>28</td>
<td>30</td>
</tr>
<tr>
<td>DALLE Reddit</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>DALLE Reddit</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>DALLE Reddit</td>
<td>34</td>
<td>34</td>
</tr>
</tbody>
</table>

Figure 6: Qualitative generations for fine-grained classes with rare vocabulary. **Example Image** is an exemplar of the class from the CUB [45]/Dogs [21] dataset. Though the optimization process of DOODL starts from pixels not matching the target class (Original Generation) due to the rare vocabulary failure of StableDiffusion, it is able to produce an image of the targeted species using the gradients of a supervised classifier trained solely on the specialized dataset.

![Figure 6](image.png)

5.2. Vocabulary Expansion

Fine-grained visual classification (FGVC) models aim to classify datasets with subtle variations between classes, such as the Caltech-UCSD Birds (CUB) [45], Stanford Dogs (Dogs) [21], and FGVC-Aircraft (FGVC-A) [27] datasets. We seek to expand and refine the vocabulary of a pretrained DDM using DOODL such that it can generate a specific class learnt by an FGVC classifier. For classifier guidance, we use binary cross-entropy (BCE) loss from supervised models trained on FGVC datasets. Given a model \( m \) trained on a dataset with classes \( \{\text{class}_i\}_{i=1}^n \), we generate instances of class \( j \) via loss \( c(x^{(i)}_{0}) = \lambda \text{BCE}(m(x^{(i)}_{0}), j) \), \( (\lambda_{\text{DOODL}}, \lambda_{\text{Baseline}}) = (0.05, 5) \).

We evaluate DOODL’s ability to expand Stable Diffusion’s vocabulary using FGVC classifiers trained, respectively, on CUB, Dogs, and FGVC-A using WS-DAN [19]. The majority of concepts present in these datasets are rare or non-existent in the training data and as such, the pretrained DDM cannot generate accurate images for them (Figure 6-second row). We measure the FID [15] between a set of generated images (4 seeds) and the validation set of the FGVC dataset being studied. Despite overlap between ImageNet and the targeted datasets, FID has historically still been considered a useful generative evaluation metric [6]. In Table 3 we see that DOODL reduces FID compared to original Stable Diffusion generation on all datasets (average decrease of \( \sim 3\% \)), while classifier guidance does not improve the original generation FID on any. This indicates that DOODL is able to better incorporate the gradient signal from the classifiers as compared to the one-step approximation. Indeed, example images (Figure 6) show that DOODL can produce images with requisite fine-grained features that identify the object category in question (e.g., a bird species or a dog breed), such as colors, textures, and shapes. DOODL obtains the least relative improvement on Aircrafts, perhaps due to the class differences being largely subtle structural changes with very few textural cues that can act as guiding signals for the optimization process.

5.3. Aesthetic Improvement

Image generation research has emphasized the ability to generate aesthetically pleasing images. Notably, the publicly available Stable Diffusion model is trained on the LAION-5B dataset pre-filtered using an “Aesthetics Predictor” model. The “Aesthetics Predictor” model \( a \) is a linear head trained on top of CLIP visual embeddings to predict a scalar value in the range of \([1,10]\), indicating perceived aesthetic quality (trained on [31]). This can easily be incorporated into DOODL’s framework with the cost function \( c_{\text{aes}}(x^{(i)}_{0}) = \lambda \cdot |a(x^{(i)}_{0}) - A| \), \( \lambda_{\text{DOODL}} = 1 \) where \( A = 10 \) for aesthetic maximization and 1 for minimization.

While we find improvement signal for incorporating \( c_{\text{aes}} \) into text-to-image generations, we place the results in the Supplementary and instead focus on editing an existing image to *increase* its perceptual appeal, a novel task with a practical application for users who may wish to improve the aesthetic quality of their photographs. We employ EDICT to unconditionally invert an existing image to a latent pair \((x_T, y_T)\) and then optimize the latents w.r.t. \( c_{\text{aes}} \). Qualitative results of incorporating this cost function...
Figure 7: We use EDICT to invert a real-world image to latents which are then optimized using DOODL, targeting an “aesthetic score” of either 1 (top row) or 10 (bottom).

are shown in Figure 7 where in-the-wild images are edited to have lower or higher aesthetic quality. We quantify DOODL’s editing ability using human evaluation in Figure 8, seeing that DOODL produces satisfactory edits significantly more often than the baseline classifier guidance. For the baselines in the latter, we find that inverting the images to $t = T$ results in generation content largely decorrelated from the original image, and invert to $t = \frac{T}{2}$ which we find to produce satisfactory edits.

5.4. Visual Personalization Guidance

Visual personalization, making a diffusion model generate images that contain a highly specific entity or concept based off of a limited exemplar set of images, is an area that has garnered much research attention. Most methods, such as Dreambooth [37] or Textual Inversion [12] learn or finetune components such as text embeddings or subsets of the diffusion model parameters in order to generate visually personalized images. While highly effective, the training cost and subsequent model storage are drawbacks. We propose a novel on-the-fly personalization paradigm using DOODL to employ a pretrained recognition model in lieu of any additional generative model training or tuning.

We utilize the distance from Equation (10) taken between the image embedding of a conditioning image, $CLIP_{Image}(C_{Image})$, and that of the current generation $CLIP_{Image}(x_i^0)$ We find ensembling ViT/B-32, ViT/L-14, and ViT/g-14 CLIP models with loss weights of 0.5, 0.25, 0.25 respectively achieves good performance. Input text conditioning to the diffusion model is as standard. A similar regime of $\lambda_{DOODL}$ performs here as in Section 5.1, $\lambda_{DOODL} = 3$ for displayed results. This setting is unusually difficult for DOODL, requiring more (5×) optimization iterations than other reported experiments (additionally, we find a much higher $\lambda_{Baseline}$ needed).

We visualize generated samples in Figure 9, placing dogs pictured in the top row ($C_{Image}$) into various contexts. Prompts generically refer to “a dog”, with all identifying information coming from $C_{Image}$. DOODL maintains faithfulness to both the text prompt and visual conditioning; a first for non-learned methods to our knowledge.

We perform a human evaluation, sampling four target dog images from Imagenet [7] and using four prompt templates ([A dog at the Acropolis/swimming/in a bucket/wearing sunglasses]) across 6 random seeds, resulting in 96 generated images per method. For each image, three labelers are asked whether the dog in the generated image appears to be the same dog as the original with the context matching the caption. A majority of labelers classify the original generation and baseline as successful just 2% and 1% of the time respectively as opposed to 14% for DOODL, 7× more often than either baseline. Further analysis is given in the Supplementary.
Figure 9: Personalization generations from a single seed. Baseline Clf. Guidance does not depict the pictured dog without destabilizing. DOODL produces realistic images containing both the desired text and image conditioning.

6. Conclusion & Future Work

In this work, we demonstrated that Direct Optimization Of Diffusion Latents (DOODL) offers an exciting new way to incorporate the knowledge of pretrained recognition networks in the generative process of diffusion models. We expect future work to both expand the types of guidances incorporated as well as sophisticate and accelerate the optimization process of DOODL to incorporate the process into applications that require higher compute efficiency.

References


