Enhancing Image Layout Control with Loss-Guided Diffusion Models: Supplementary Materials

Zakaria Patel
LeapTools
1255 Bay St. Unit 403, Toronto, ON M5R 2A9
zakaria@leaptools.com

Kirill Serkh
University of Toronto
40 St. George St., Toronto, ON M5S 2E4
kserkh@math.toronto.edu

A. Diffusion Models

A.1. Denoising Diffusion Probabilistic Models

Diffusion models [4] are characterized by two principle algorithms. The first is the forward process, wherein the data $\mathbf{x}_0$ is gradually corrupted by Gaussian noise until it becomes pure noise, which we denote by $\mathbf{x}_T$. The reverse process moves in the opposite direction, attempting to recover the data by iteratively removing noise. The denoiser $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ is typically a UNet [11] which accepts an image $\mathbf{x}_t$, and predicts its noise content $\boldsymbol{\epsilon}$. Removing a fraction of this noise yields a slightly denoised image $\mathbf{x}_{t-1}$. Repeating this process over T steps produces a noise-free image $\mathbf{x}_0$.

Operating directly on the image $\mathbf{x}_t$ in pixel-space is computationally expensive. As an alternative, latent diffusion models have been proposed to curtail this high cost, in which the denoising procedure is performed in latent space, whose dimensionality is typically much lower than pixel space. Stable Diffusion [10] is one example of a latent diffusion model which achieves state-of-the-art performance on various image synthesis tasks. It leverages a powerful autoencoder to project to and from latent space, where the standard denoising procedure is performed. Images in latent space are typically denoted by $\mathbf{z}_t$, and the encoder and decoder are denoted by $\mathcal{E}$ and $\mathcal{D}$, respectively, $\mathbf{z}_t = \mathcal{E}(\mathbf{x}_t)$ and $\mathbf{x}_t = \mathcal{D}(\mathbf{z}_t)$.

During training, samples from the true data distribution $q(\mathbf{x}_0)$ are corrupted via the forward process. By training a diffusion model to learn a reverse process in which it iteratively reconstructs these noisy samples into noise-free samples, it is possible to generate images from pure noise at inference time. This corresponds to sampling from an approximation $p_\theta(\mathbf{x}_0)$ to the data distribution, $q(\mathbf{x}_0)$. This generation process can be guided by introducing an additional input vector $\mathbf{y}$, which is often a text prompt. In this case, the model produces samples from an approximation $p_\theta(\mathbf{x}_0|\mathbf{y})$ to the conditional distribution $q(\mathbf{x}_0|\mathbf{y})$.

In denoising diffusion probabilistic models (DDPM) [4], the forward process is characterized by the Markov chain $q(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$, for some noise schedule β_t. In this case, $q(\mathbf{x}_t|\mathbf{x}_0) \sim \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$. The reverse process is typically modeled by a learned Markov chain $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \sim \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$, where σ_t is an untrained time dependent constant, usually with $\tilde{\beta}_t \leq \sigma_t^2 \leq \beta_t$ and $\tilde{\beta}_t = \beta_t(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)$, or with σ_t simply chosen equal to $\sqrt{\beta_t}$.

It is not efficient to optimize the log-likehood $\mathbb{E}[-\log p_\theta(x_0)]$ directly, since computing $p_\theta(\mathbf{x}_0)$ requires marginalizing over $\mathbf{x}_{1:T}$. Instead, one can use importance sampling to write

$$p_\theta(\mathbf{x}_0) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]. \tag{1}$$

Then, by Jensen's inequality,

$$-\log p_\theta(\mathbf{x}_0) \leq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]. \tag{2}$$

The right hand side is the usual evidence lower bound (ELBO), which is minimized instead. Ho *et al.* [4] show that minimizing the ELBO is equivalent to minimizing

$$\mathbb{E}_t[\lambda(t)\mathbb{E}_{q(\mathbf{x}_0),\epsilon}[\|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon\|_2^2]], \tag{3}$$

for some positive function $\lambda(t)$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_t(\mathbf{x}_0, \epsilon) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, and

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\Big). \tag{4}$$

A.2. Score Matching

Since $q(\mathbf{x}_t|\mathbf{x}_0)$ is a normal distribution, we know that

$$\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}}. \tag{5}$$

Thus, minimizing (3) is equivalent to minimizing

$$\mathbb{E}_t[\lambda(t)\mathbb{E}_{q(\mathbf{x}_0)}\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t|\mathbf{x}_0)\|_2^2]], \tag{6}$$

for some positive function $\lambda(t)$, where

$$\mathbf{s}_\theta(\mathbf{x}_t, t) := -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}. \tag{7}$$

It is known that this loss is minimized when $\mathbf{s}_\theta(\mathbf{x}_t, t) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$ [14], so given enough parameters, $\mathbf{s}_\theta(\mathbf{x}_t, t)$ will converge to $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$ almost everywhere. Given an approximation to the score function, it is possible to sample from $p_\theta(\mathbf{x}_0)$ using annealed Langevin dynamics [12].

Letting the forward process posterior mean $\tilde{\mu}_t$ be defined by $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \sim \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$, we have that

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \tag{8}$$

(see [4]). With this, the mean $\mu_\theta(\mathbf{x}_t, t)$ of the reverse process can be understood as

$$\mu_\theta(\mathbf{x}_t, t) = \tilde{\mu}_t(\mathbf{x}_t, D_\theta(\mathbf{x}_t, t)), \tag{9}$$

where

$$D_\theta(\mathbf{x}_t, t) := \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1-\bar{\alpha}_t)s_\theta(\mathbf{x}_t, t)) \tag{10}$$

is an approximation to Tweedie's formula

$$\begin{aligned}&\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1-\bar{\alpha}_t)\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)) \\ &\qquad = \frac{1}{\sqrt{\bar{\alpha}_t}}\mathbb{E}_q[\sqrt{\bar{\alpha}_t}\mathbf{x}_0|\mathbf{x}_t] = \mathbb{E}_q[\mathbf{x}_0|\mathbf{x}_t]\end{aligned} \tag{11}$$

(see [3]).

A.3. Stochastic Differential Equations

Song *et al*. [13] showed that the forward process of DDPM can viewed as a discretization of the stochastic differential equation (SDE)

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\,dt + \sqrt{\beta(t)}\,d\mathbf{w}, \tag{12}$$

where $\mathbf{w}$ denotes the Wiener process. There, the authors point out that any SDE of the form $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t) + g(t)d\mathbf{w}$, where $\mathbf{x}_0 \sim p_0(\mathbf{x}_0)$ can be reversed by the SDE $d\mathbf{x} = (\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla_{\mathbf{x}} \log p_t(\mathbf{x}))\,dt + g(t)\,d\bar{\mathbf{w}}$, where $\bar{\mathbf{w}}$ is the standard Wiener process in the reverse time direction, and where $\mathbf{x}_T \sim p_T(\mathbf{x}_T)$. Furthermore, each SDEs admits a family of related SDEs that share the same marginal distributions $p_t(\mathbf{x}_t)$. One of these SDEs is purely deterministic, and is known as the probability flow ordinary differential equation (ODE).

If the score function $\mathbf{s}_\theta(\mathbf{x}_t, t)$ is available, then it is possible to sample from $p_\theta(\mathbf{x}_0)$ by solving the probabily flow ODE, starting with samples from $p_\theta(\mathbf{x}_T)$. This results in a deterministic mapping from noises images $\mathbf{x}_T$ to clean images $\mathbf{x}_0$. This sampling process can be performed quickly with the aid of ODE solvers [8].

A.4. Classifier-Free Guidance

In order to generate images following a user-supplied text prompt, the denoiser $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y})$ of a latent diffusion model is trained with an additional input given by a sequence of token embeddings $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$. A single denoiser, usually a UNet, is trained over a variety of text prompts, and the token embeddings influence the denoiser by a cross-attention mechanism in both the contractive and expansive layers. Ho and Salimans [5] found that, rather than sampling images using the conditional denoiser alone, better results can be obtained by taking a combination of conditional and unconditional noise estimates,

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t, t, \mathbf{y}) = (1+w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \{\}), \tag{13}$$

where w represents the intensity of the additive term $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y}) - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \{\})$. For $-1 \leq w \leq 0$, this noise prediction can be viewed as an approximation to ($-\sigma_t$ times) the score function of the marginal distribution $\tilde{p}_\theta(\mathbf{z}_t|\mathbf{y}) \propto p_\theta(\mathbf{z}_t|\mathbf{y})^{1+w} p_\theta(\mathbf{z}_t|\{\})^{-w}$. In classifier-free guidance (CFG), $w \gg 0$, which does not have a simple interpretation in terms of the marginal distributions of the new denoising process.

B. Additional Experiments

We provide two additional sets of comparisons between our proposed method (iLGD), BoxDiff [16], Chen *et al*. [2], MultiDiffusion [1], and Stable Diffusion [10]. In Figure B.1, we compare the three methods using same prompts and bounding boxes as in Figure 3, but using a different random seed for each set of images. In Figure B.2, we compare the methods using an entirely new set of prompts and bounding boxes. We also provide a set of examples generated using just our proposed method (iLGD) in Figure B.3.

C. Detailed Methods

Implementation Details We implement our method, illustrated graphically in Figure C.1, on the official Stable Diffusion v1.4 model [10] from HuggingFace. All images are generated using 50 denoising steps and a classifier-free guidance scale of 7.5, unless otherwise noted. We use the noise scheduler `LMSDiscreteScheduler` [6] provided by HuggingFace. Experiments are conducted on an NVIDIA TESLA V100 GPU.

We perform attention injection over all attention maps. When performing injection, we resize the mask m to the appropriate resolution, depending on which layer of the UNet the attention maps are taken from. For loss guidance, we again use all of the model's attention maps, but resize them 16×16 resolution, and compute the mean of each map over all pixels. We apply the softmax function over these means to obtain a weight vector $\mathbf{w}$, where each entry w_j is

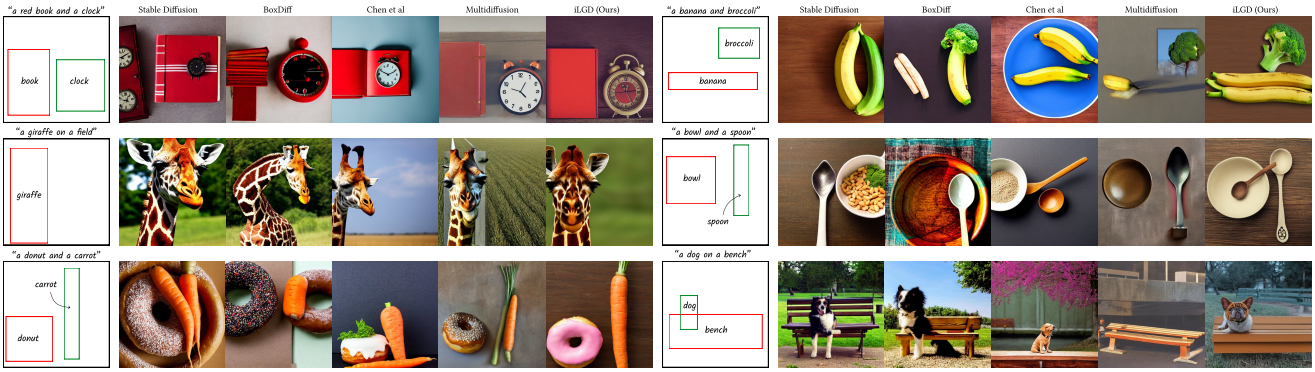

Figure B.1. A comparison of iLGD against BoxDiff, Chen *et al*., MultiDiffusion, and Stable Diffusion, using the same prompts as Figure 3 but different random seeds, with the seed kept the same across each set of images.

Figure B.2. A comparison of iLGD against BoxDiff, Chen *et al*., MultiDiffusion, and Stable Diffusion. The random seed kept the same across each set of images.

the scalar weight associated with the j-th resized attention map. Finally, we obtain the attention map A_t by taking a weighted average over all resized attention maps at time t, using the appropriate weight w_j for each map.

When attempting to control the layout of a generated image, we find that skipping the first step, so that it remains a standard denoising step, leads to better results. We do this for all experiments conducted in this paper which use either injection or loss guidance or both. In iLGD, we use $\eta = 0.48$, $\nu' = 0.75$, $t_{\text{loss}} = 25$, and $t_{\text{inject}} = 10$, unless otherwise noted. In our ablation experiments, we keep the injection strength at $\nu' = 0.75$ when performing just attention injection. When performing just loss guidance, we increase the loss-guidance strength to $\eta = 1$, in order to make loss guidance alone exert sufficient influence over the final image layouts.

In our comparisons with BoxDiff, we maintain the default parameters the authors provide in their implementation. We start with $\alpha_T = 20$, which decays linearly to $\alpha_0 = 10$, and perform guidance for 25 iterations out of a total of 50 denoising steps.

In our comparisons with the method of Chen *et al*., we also maintain the default parameters the authors provide in their implementation, setting the loss scale factor to $\eta = 30$.

Evaluation with YOLOv4 In this section, we describe in detail how we obtain the AP@50 scores in Table 2. In classical object detection, a model is trained to detect and localize objects of certain classes in an image, typically by predicting a bounding box which fully encloses the object. The accuracy of the model's predicted bounding box, B_p, is evaluated by comparison to the corresponding ground truth bounding box, B_{gt}. More specifically, we compute the intersection over union (IOU) over the pair of bounding boxes:

$$\text{IOU} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})}. \tag{14}$$

The IOU is then compared to a threshold t, such that, if IOU $\geq t$, then the detection is classified as correct. If not, then the detection is classified as incorrect. In our case, we follow Li *et al*. [7] and treat the object detection model as

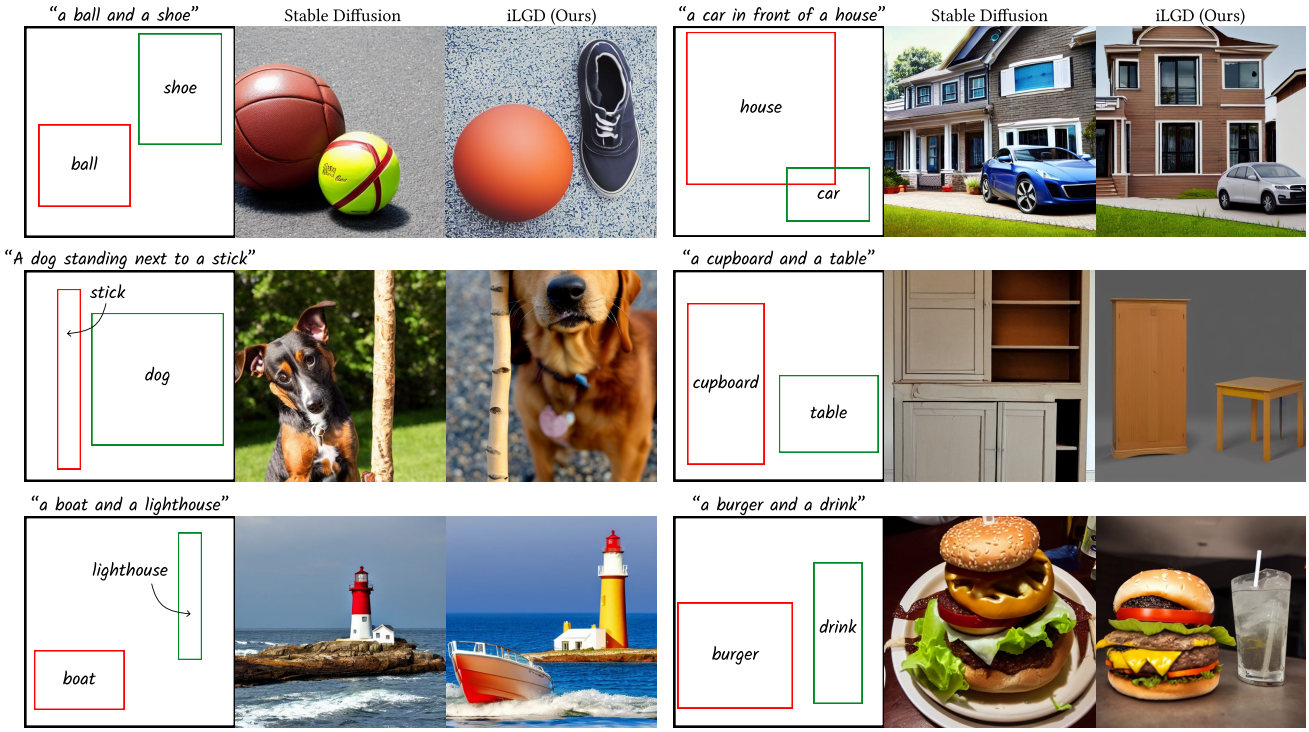

Figure B.3. Injection loss guidance (iLGD) uses attention injection and loss guidance to generate high quality images conforming to a given layout. The first column of each set of images depicts the bounding box input to the diffusion model. The second column is the output of Stable Diffusion alone. The third column is our method, using the same random seed.

an oracle, where we assume that it provides the bounding boxes of objects in a given image with perfect accuracy. In particular, we first define a layout through a set of ground truth bounding boxes, describing the desired positions of each object. We then generate an image according to this layout, and subsequently apply the object detection model to the generated image to obtain a set of predicted bounding boxes. Finally, to evaluate how similar the layout of the generated image is to the desired layout, we compare each predicted bounding box, B_p, to the corresponding ground truth bounding box, B_{gt}, by computing their IOU. We use a IOU threshold of 0.5.

To calculate the average precision, we first need to compute the number of true positives (TP), false positives (FP), and false negatives (FN). We count a false negative when no detection is made on the image, even though a ground truth object exists, or when the detected class is not among the ground truth classes. We also count a false negative as well as a false positive when the correct detection is made, but IOU < 0.5, and a true positive when IOU ≥ 0.5. Using these quantities, we compute the precision P and recall R as:

$$P = \frac{TP}{TP + FP}, \tag{15}$$

$$R = \frac{TP}{TP + FN}. \tag{16}$$

We repeat this for classifier confidence thresholds of 0.15 to 0.95, in steps of 0.05, so that we end up with 17 values for precision and recall, respectively. We then construct a precision-recall curve, and compute the average precision using 11-point interpolation [9]:

$$\text{AP}_{11} = \frac{1}{11} \sum_{R \in \{0, 0.1, \ldots, 0.9, 1\}} P_{\text{interp}}(R), \tag{17}$$

where

$$P_{\text{interp}}(R) = \max_{\tilde{R} \geq R} P(\tilde{R}). \tag{18}$$

Image Quality Assessment Wang *et al*. [15] suggest using the pair {good photo, bad photo} instead of {high quality, low quality} to measure quality, as they find that it corresponds better to human preferences. However, we choose the latter to remain agnostic to the image's style, as we believe the former carries with it a stylistic bias, due to the word "photo."

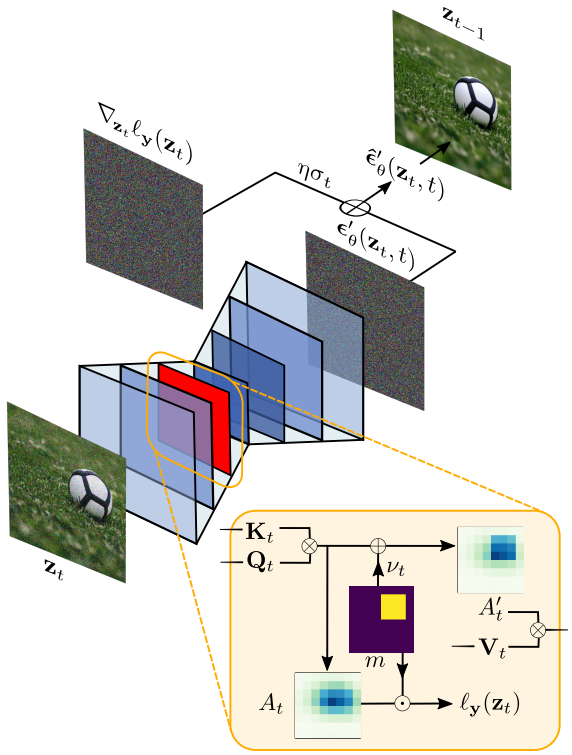

Figure C.1. A graphical depiction of injection loss guidance (iLGD).

Contrast Calculation We calculate the RMS contrast by using OpenCV's `.std()` method on a greyscale image.

Saturation Calculation We calculate the saturation by working in HSV space and using OpenCV's `.mean()` method on the image's saturation channel.

References

[1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 202:1737–1752, 2023. 2

[2] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *IEEE Wint. Conf. Appl.*, pages 5343–5353, 2024. 2

[3] Bradley Efron. Tweedie's formula and selection bias. *J. Am. Stat. Assoc.*, 106(496):1602–1614, 2011. 2

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. volume 33, pages 6840–6851, 2020. 1, 2

[5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS*, Workshop: Deep Generative Models and Downstream Applications, 2021. 2

[6] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, volume 35, 2022. 2

[7] Zejian Li, Jingyu Wu, Immanuel Koh, Yongchuan Tang, and Lingyun Sun. Image synthesis from layout with locality-aware mask adaption. In *ICCV*, pages 13819–13828, 2021. 3

[8] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, volume 35, 2022. 2

[9] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. A survey on performance metrics for object-detection algorithms. In *Int. Conf. Syst. Signal.*, pages 237–242. IEEE, 2020. 4

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 1

[12] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019. 2

[13] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 2

[14] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Comput.*, 23(7):1661–1674, 2011. 2

[15] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, volume 37, pages 2555–2563, 2023. 4

[16] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *ICCV*, pages 7452–7461, 2023. 2