Exploring Classifier-Free Guidance in Diffusion Models

Dhruman Gupta, Rushil Gupta
Introduction to Machine Learning Final Project

May 12, 2025

Abstract

We present an expository study of classifier-free guidance (CFG) in diffusion-based image generation. Leveraging a custom UNet trained on MS COCO captions with a frozen VAE and CLIP text encoder from Stable diffusion v1.4[5], we explore the qualitative effects of varying CFG weights on sample fidelity and diversity. Results showcase both successful generations on seen prompts and failure modes on novel objects.

1 Introduction and Background

Generative modeling has witnessed a renaissance with the advent of diffusion-based methods, which achieve remarkable fidelity by simulating a gradual denoising process. In this section, we first review the fundamentals of diffusion models, then discuss how conditioning can steer generation and introduce classifier guidance, and finally describe the more flexible classifier-free guidance approach that forms the core of our project.

1.1 Diffusion Models

Diffusion models [1] formulate generation as the reversal of a known, fixed noising process. The underlying idea is to "add noise" to the data distribution $p_{\text{data}}(x)$, and then learn to reverse this process. By adding small amounts of noise, it becomes easier to learn the reverse process.

Consider a data distribution over images $x_0 \sim p_{\text{data}}(x)$, and define a forward Markov chain that progressively adds Gaussian noise:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t;\ \sqrt{1-\beta_t}\, x_{t-1},\ \beta_t I\big), \quad t = 1, \ldots, T, \tag{1}$$

where $\{\beta_t\}$ is a small, increasing variance schedule. After T steps, x_T is essentially isotropic Gaussian noise. To generate new samples, we learn a parameterized reverse process

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\Big(x_{t-1};\ \mu_\theta(x_t, t),\ \sigma_t^2 I\Big), \tag{2}$$

with mean

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \epsilon_\theta(x_t, t)\Big),$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. The network $\epsilon_\theta(x_t, t)$ predicts the noise component at each step. Training minimizes the denoising objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0, \epsilon, t}\Big[\big\|\epsilon - \epsilon_\theta(x_t, t)\big\|^2\Big], \quad x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon,\ \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

In practice, this simple loss yields stable training and high-quality samples.

Algorithm 1 Unconditional DDPM Sampling

Input: noise schedule $\{\beta_t\}_{t=1}^{T}$, learned model ϵ_θ
$x_T \sim \mathcal{N}(0, I)$
for $t = T, \dots, 1$ **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t, t)$
  $\mu \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}\right)$
  Sample $x_{t-1} \sim \mathcal{N}(x_{t-1};\, \mu, \sigma_t^2 I)$
end for
return x_0

1.2 Conditioning and Classifier Guidance

While unconditional diffusion can produce diverse samples, many applications demand control over the output, such as generating an image of a specified class or matching a textual prompt y. Ideally, one would sample from the conditional distribution $p(x_0 \mid y)$. Classifier guidance [2] achieves this by leveraging an auxiliary classifier $p_\phi(y \,|\, x_t)$ trained on noisy inputs.

Since we want to sample from $p(x_0 \mid y)$, we can use Bayes' rule and score decomposition to decompose the gradient:

$$\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y \mid x_t),$$

we adjust the denoising mean in the reverse step:

$$\tilde{\mu} = \mu_\theta(x_t, t) + s\, \sigma_t^2\, \nabla_{x_t} \log p_\phi(y \mid x_t), \tag{4}$$

where s is the guidance strength. Intuitively, the classifier gradient pushes x_t toward regions where the classifier predicts y.

Algorithm 2 Classifier-Guided Sampling

Input: schedule $\{\beta_t\}$, model ϵ_θ, classifier p_ϕ, strength s
$x_T \sim \mathcal{N}(0, I)$
for $t = T, \dots, 1$ **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t, t)$
  $\mu \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}\right)$
  $g \leftarrow \nabla_{x_t} \log p_\phi(y \,|\, x_t)$
  $\mu \leftarrow \mu + s\, \sigma_t^2\, g$
  Sample $x_{t-1} \sim \mathcal{N}(x_{t-1};\, \mu, \sigma_t^2 I)$
end for
return x_0

Although effective, classifier guidance requires training a separate classifier on every noise level and can incur significant compute overhead. Moreover, classifier gradients may introduce artifacts if the classifier is imperfect.

Another flaw of the classifier guidance is that it only works for fixed classes, which the classifier is trained on. However, in many cases, we may want to generate samples from an evolving distribution of classes, which is not possible with classifier guidance.

1.3 Classifier-Free Guidance

Classifier-free guidance (CFG) [3] elegantly sidesteps the need for an external classifier by training the diffusion model itself to operate both with and without conditioning. The existing diffusion models can be thought of as conditional models that sample with condition $\varnothing$. Given training data with appropriate condition labels, we can train the model to sample from these conditions.

In classifier-free guidance, we train the model to sample from these conditions. However, for a more generalisable model, we also want it to sample unconditionally - this gives higher diversity. Thus, during training, the model is trained to sample from both $\varnothing$ and c:

During training, each sample's condition c (e.g. text embedding) is randomly dropped with probability p_{drop} (for example, 0.2), so the model learns to predict

$$\epsilon_\theta(x_t, t, c) \quad \text{and} \quad \epsilon_\theta(x_t, t, \varnothing)\,.$$

At inference, one interpolates between these two predictions:

$$\epsilon_{\text{CFG}} = (1 + w)\,\epsilon_\theta(x_t, t, c) \;-\; w\,\epsilon_\theta(x_t, t, \varnothing), \tag{5}$$

where $w \geq 0$ is the guidance weight. Substituting ϵ_{CFG} into the reverse mean (Eq. 2) yields a sample that balances fidelity to the condition against sample diversity.

Note: by adding the weight w, we actually end up samping from the following distribution:

$$p_{\text{CFG}}(x_0) = p(x_0 \mid c)^{1+w} p(x_0)^{-w}.$$

While this is not the same as sampling from $p(x_0 \mid c)$, it often works better for generating images. Stable Diffusion uses $w = 7.5$ as a general default. This value is often exposed as a parameter to the user in the pipeline.

Algorithm 3 Classifier-Free Guidance Sampling

Input: schedule $\{\beta_t\}$, model ϵ_θ, weight w, condition c
$x_T \sim \mathcal{N}(0, I)$
for $t = T, \ldots, 1$ **do**
  $\epsilon_c \leftarrow \epsilon_\theta(x_t, t, c)$
  $\epsilon_u \leftarrow \epsilon_\theta(x_t, t, \varnothing)$
  $\epsilon \leftarrow (1 + w)\,\epsilon_c - w\,\epsilon_u$
  $\mu \leftarrow \frac{1}{\sqrt{\alpha_t}}\big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\big)$
  Sample $x_{t-1} \sim \mathcal{N}(x_{t-1};\, \mu, \sigma_t^2 I)$
end for
return x_0

CFG offers several advantages: it requires no extra classifier, allows a single model to flexibly trade off between unconditional and conditional generation, and has been shown to deliver state-of-the-art sample quality with minimal overhead.

1.4 UNet Architecture

Finally, we need to understand the UNet architecture, which is the core of our model. The UNet architecture, originally introduced for biomedical image segmentation [4], has become a cornerstone for image-to-image tasks, including denoising and generative modeling. Its design enables the model to reverse a corruption process by progressively refining features at multiple spatial resolutions. The key idea is to combine high-resolution spatial information with deep, coarse features, thereby capturing both local and global context.

The UNet consists of three main components:

- **Downsampling Path:** Stacks of convolutional layers (Conv→ReLU→Conv→ReLU) progressively downsample the input, extracting increasingly abstract features.
- **Bottleneck:** The deepest layer, which captures the most abstract representation of the input. No pooling is used here—just convolutions.
- **Upsampling Path:** Upsampling layers (typically transposed convolutions) restore the spatial size, halving the number of channels at each step. Skip connections concatenate encoder feature maps to the decoder at each level, preserving fine-grained details lost during downsampling.

Each block also has a cross-attention layer, which allows the model to attend to the text prompt effectively. This architecture is particularly well-suited for diffusion models, as it allows the network to learn both global structure and local details necessary for high-fidelity image synthesis.

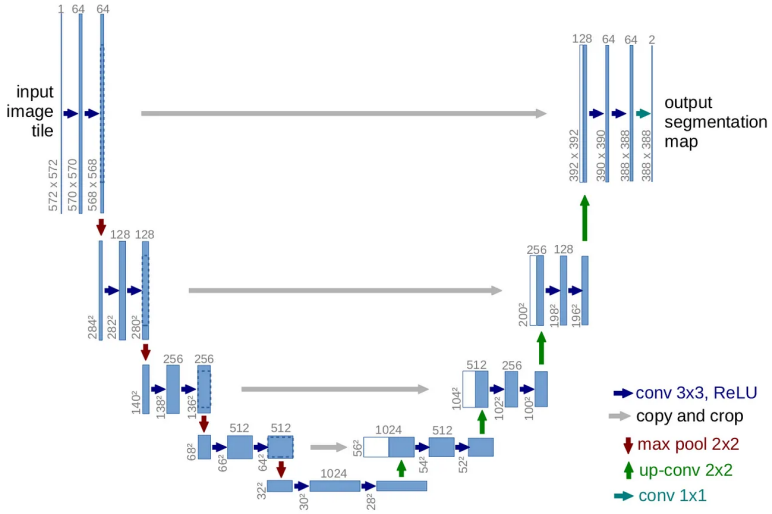


Figure 1: UNet architecture used for denoising in diffusion models.

2 Dataset and Preprocessing

We use the MS COCO 2014 captions dataset, which contains ∼100,000 images annotated for object detection, segmentation, and captioning. The dataset comprises 80 categories, including common objects such as cars, bicycles, and animals, as well as more specific categories like umbrellas, handbags, and sports equipment. For our experiments, all images are resized to 128×128 pixels and center-cropped to ensure uniformity and compatibility with the model architecture.

During training, we employ a caption dropout strategy: for each sample, the text condition (caption) is randomly dropped with a fixed probability. This encourages the model to learn both conditional and unconditional generation, which is essential for classifier-free guidance.

3 Model and Training

Architecture. Our model leverages a frozen VAE and CLIP text encoder from Stable Diffusion v1.4[5] to encode and decode images and text prompts, respectively. The core denoising network is a custom UNet, constructed with four downsampling and four upsampling blocks, each containing two layers. The cross-attention dimensions are set to match the CLIP encoder (768 channels).

Training Setup. The diffusion process is set to 1,000 timesteps. We optimize the model using AdamW (learning rate 1×10^{-4}, weight decay 1×10^{-2}) and apply a CosineAnnealingLR schedule over 450 epochs. The batch size is 320, with gradient accumulation over 4 steps to accommodate hardware constraints.

Algorithm 4 Training Procedure for Classifier-Free Guided Diffusion

Input: Dataset $\mathcal{D} = \{(x_0, c)\}$, noise schedule $\{\beta_t\}_{t=1}^{T}$, dropout probability P_{drop}
for each minibatch **do**
  $(x_0, c) \sim \mathcal{D}$ // sample random data and condition
  $t \sim \text{Uniform}(\{1, \ldots, T\})$ // sample timestep
  With probability P_{drop}, $c \leftarrow \varnothing$ // drop condition
  $e_c = \text{CLIP}(c)$ // encode c using CLIP
  $z_0 = \text{VAE}(x_0)$ // encode x_0 using VAE encoder
  $\epsilon \sim \mathcal{N}(0, I)$ // sample noise
  $z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$ // add noise to z_0
  $\hat{\epsilon}_\theta = \epsilon_\theta(z_t, t, e_c)$ // predict noise
  Optimise for $\mathcal{L} = \|\epsilon - \hat{\epsilon}_\theta\|^2$
end for

Algorithm 5 Sampling Procedure with Classifier-Free Guidance

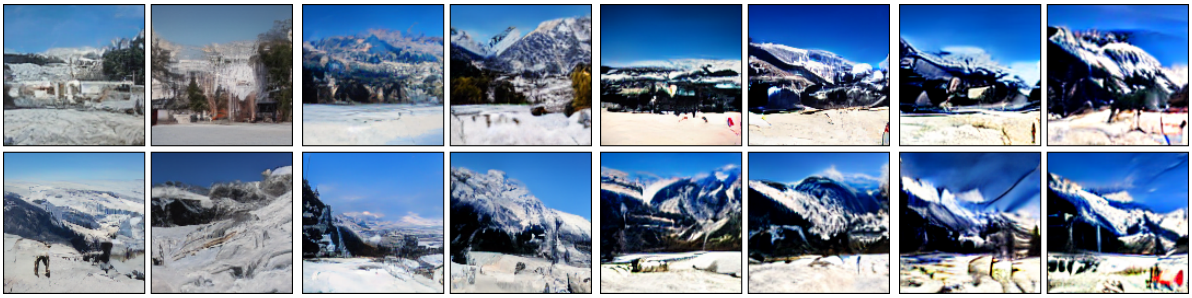
Input: Condition c, guidance weight w, number of steps T
Encode c using CLIP: $e_c \leftarrow \text{CLIP}(c)$
$z_T \sim \mathcal{N}(0, I)$ // Sample initial latent
for $t = T, \ldots, 1$ **do**
  Predict $\epsilon_c = \epsilon_\theta(z_t, t, e_c)$
  Predict $\epsilon_u = \epsilon_\theta(z_t, t, \varnothing)$
  Compute guided noise: $\epsilon = (1 + w)\epsilon_c - w\epsilon_u$
  Compute mean: $\mu = \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right)$
  Sample $z_{t-1} \sim \mathcal{N}(\mu, \sigma_t^2 I)$
end for
Decode image: $\hat{x}_0 \leftarrow \text{VAE\_dec}(z_0)$
return $\hat{x}_0$

4 Inference and Sampling

For evaluation, we assemble systematically vary the classifier-free guidance weights $w \in \{1.0, 3.0, 5.0, 7.0\}$. To ensure reproducibility, both the random seed and the number of timesteps are fixed across all experiments. For each prompt and guidance weight, we generate a 2×2 grid of samples, allowing for qualitative comparison of fidelity and diversity.

5 Results

We present qualitative results illustrating the effect of guidance weight on sample quality. Each figure shows nine samples for a given prompt and guidance weight.



(a) $w = 1.0$ (b) $w = 3.0$ (c) $w = 5.0$ (d) $w = 7.0$

Figure 2: Samples for the prompt "a beautiful snowy mountain landscape" at different guidance weights.

(a) $w = 1.0$ (b) $w = 3.0$ (c) $w = 5.0$ (d) $w = 7.0$

Figure 3: Samples for the prompt "a beach" at different guidance weights.

(a) $w = 1.0$ (b) $w = 3.0$ (c) $w = 5.0$ (d) $w = 7.0$

Figure 4: Samples for the prompt "a tennis court" at different guidance weights.

6 Discussion

Increasing the guidance weight w enhances adherence to the prompt, resulting in samples that more closely match the intended description. However, this comes at the cost of reduced sample diversity, and at high values of w, the model can exhibit mode collapse. In our experiments, the limited dataset size and model capacity led to visible artifacts at high guidance weights. Failure cases on novel or uncommon objects further highlight the limitations of the MS COCO dataset, which lacks sufficient examples of rare items. The chosen UNet size represents a trade-off between computational feasibility and the ability to model complex scenes, and may underfit more intricate prompts.

7 Future Work

Future directions include expanding the training corpus with larger captioned datasets such as OpenImages or LAION to improve object diversity and generalization. Parameter-efficient fine-tuning methods (e.g., LoRA) could enable the use of larger backbone models within existing compute constraints. Additionally, exploring alternative noise schedules and reducing the number of diffusion timesteps may accelerate training and yield sharper images.

8 Conclusion

We implemented classifier-free guidance in a custom diffusion model and systematically explored its qualitative impact. Our results demonstrate that classifier-free guidance provides a simple yet effective mechanism for balancing fidelity and diversity in generative image modeling. The findings underscore the importance of dataset scale and model capacity for achieving reliable and high-quality generative performance.

References

[1] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[2] Dhariwal, Prafulla and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[3] Ho, Jonathan and Tim Salimans. Classifier-Free Diffusion Guidance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[4] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv preprint arXiv:1505.04597, 2015. `https://arxiv.org/abs/1505.04597`

[5] Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. `https://arxiv.org/abs/2112.10752`