Latent Diffusion Models

Om Prakash Choudhary
omprakashc@iisc.ac.in

Abstract

The problem of generating realistic images is immensely challenging as it involves sampling meaningful data embedded in tiny clusters within a very high-dimensional image space. Broadly, generative models can be categorized into two types: (1) **Likelihood-based methods**, which explicitly model the data distribution via maximization of a variational lower bound (**ELBO**), and (2) **Implicit methods**, such as GANs [1], which learn to generate samples directly without an explicit likelihood. While GANs produce high-quality samples with efficient inference, they often suffer from issues like mode collapse and sensitivity to hyperparameters and architecture. Recently, likelihood-based approaches—particularly Denoising Diffusion Probabilistic Models (DDPM) [2], DDIM [3], and Score-Based Generative Models [4]—have achieved state-of-the-art results across a range of image generation tasks. These models benefit from stable training and even offer **fine-grained control** over the generation process (**Guidance**) due to their probabilistic formulation. In this work we explore text-conditioned image generation using Latent Diffusion Models.

1 Introduction

Recently, text-conditioned image generation has seen tremendous progress, with models like DALLE-3, GPT-SORA, and Stable Diffusion XL (SDXL) producing remarkably realistic and diverse images. Among these, SDXL stands out as a prominent example of a **Latent Diffusion Model (LDM)**, which combines the efficiency of compressed latent representations with the powerful generative capabilities of diffusion models. A key advantage of SDXL is its ability to run efficiently on commonly available **consumer GPUs**, making high-quality generative modeling accessible to a wider audience.

(a) Prompt: *Palette knife painting of an autumn cityscape*

(b) Prompt: *A kid on a high place watching a shooting star with a rainbow trail*

Figure 1: Examples of images generated by SDXL

To understand the foundations behind such models, we first review the **Denoising Diffusion Probabilistic Model (DDPM)**, the core framework upon which latent diffusion builds.

1.1 Breif introdution to DDPM

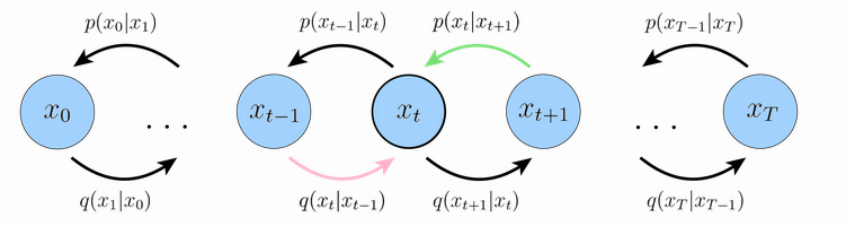


Figure 2: DDPM, adapted from [5]

DDPMs define a latent variable model with latent states $\mathbf{x}_1, \ldots, \mathbf{x}_T$ of the same dimensionality as the observed variable $\mathbf{x}_0$. The forward process is a Markov chain where noise is gradually added at each step, defined by the transition:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t;\ \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}),$$

where α_t is a predefined noise schedule chosen such that the marginal $q(\mathbf{x}_t)$ approaches the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as $t \to T$.

It turns out that the reverse process, which de-noises $\mathbf{x}_t$ given x_0 to produce x_{t-1} is also a Markov chain and the transition is given as:

$$p_(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}(\mathbf{x}_t, t), \boldsymbol{\Sigma}(\mathbf{x}_t, t)),$$

where both the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ have closed-form expressions derived from the forward process and an estimate of $\mathbf{x}_0$.

In practice, the model is trained to predict the clean image $\hat{\mathbf{x}}_0$ given a noisy input $\mathbf{x}_t$, and this estimate is used as proxy to $\mathbf{x}_0$ in the reverse distribution: $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_0)$, allowing the model to iteratively denoise $\mathbf{x}_t$ back to a high-quality sample $\hat{\mathbf{x}}_0$.

1.2 Rate-Distortion Tradeoff:

The diffusion process can be interpreted as a communication between sender and receiver through a channel given by learned data distribution. The sender has access to ground-truth denoising distribution $p_(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and for a given $\mathbf{x}_0$ and he sequentially de-noises to produce $\mathbf{x}_{t-1}$ given $\mathbf{x}_t$ and $\mathbf{x}_0$ and transmits $\mathbf{x}_{t-1}$ to receiver and the amount of information required per pixel is called **rate** (bits/dim), the receiver with limited information of $\mathbf{x}_{t-1}$ tries to predict $\hat{\mathbf{x}}_0$ and the error between $\hat{\mathbf{x}}_0$ and $\mathbf{x}_0$ is called **Distortion** (RMSE). A detailed derivation and quantification of the rate and distortion terms can be found in the **Appendix** B.

The rate-distortion trade off is that for less distortion the sender has to reveal a lot of information equivalently, a large inference steps is required to produce good reconstruction.

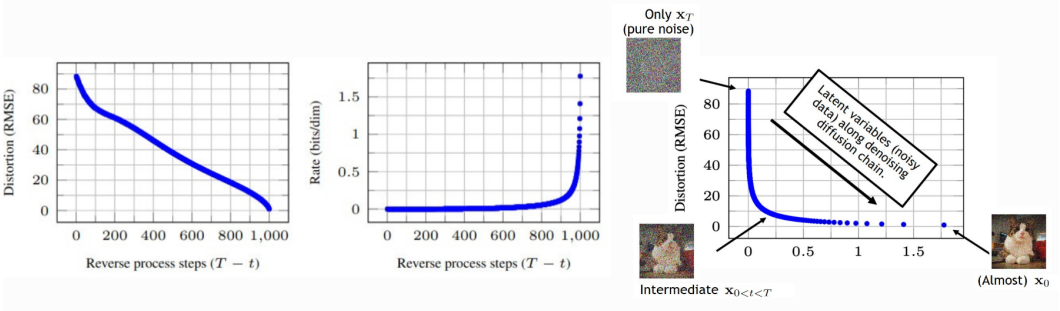


Figure 3: DDPM on CIFAR10 test set rate-distortion vs. time. Adapted from DDPM[2]

The left plot shows how the **distortion** between original and predicted original image change over the course of the reverse process. The center plot tracks how much information **rate** about the original image is transferred at each step. The right plot illustrates the rate-distortion curve — initially, moderate rate yields **significant improvement** in reconstruction, but as the curve flattens, further reduction in distortion requires **disproportionately higher** rate.

Luo et al. [5] argue that the diffusion process unfolds in two distinct phases: **semantic compression** and **perceptual compression**. This interpretation is visualized below.

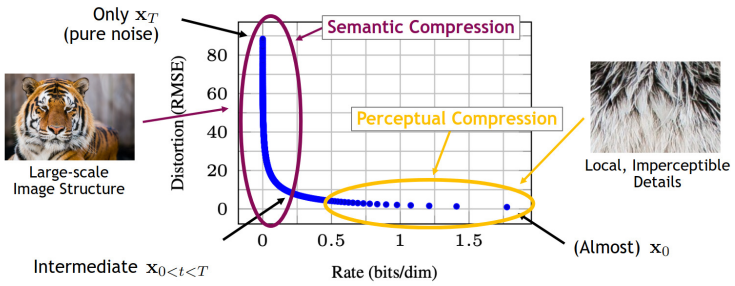


Figure 4: Perceptual and Semantic Compression

In the semantic compression phase, the model captures high-level content such as object shapes and layouts, while ignoring fine details. The subsequent perceptual compression phase refines local textures and fine-grained visual features. The DDPM authors [6] note that a disproportionate amount of model's capacity is spend to recover these **imperceptible low-level details** (See **Appendix** B).

1.3 Latent Diffusion Model

Latent Diffusion Models (LDMs) address the rate-distortion trade-off by operating in a compressed latent space where high-frequency perceptual details are intentionally discarded. This is achieved by first learning a lower-dimensional representation of images using a **regularized** autoencoder such as a **Vector-Quantized VAE**[7] or a **KL-regularized VAE**. These autoencoders are trained to encode images into spatially smaller latent maps—typically $8\times$ smaller in each dimension than the original—while the regularization (quantization or KL penalty) encourages the encoder to focus on semantic content and discard imperceptible texture details.

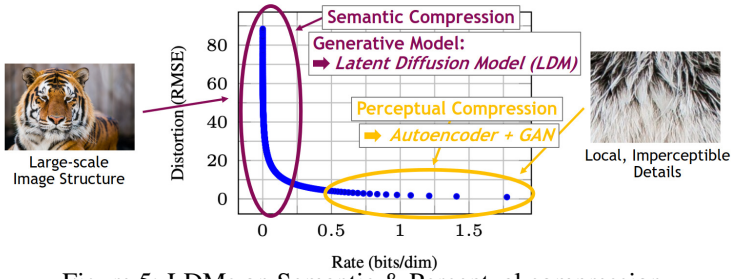


Figure 5: LDMs on Semantic & Perceptual compression.

This strategy offers several key advantages:

1. A universal autoencoder can be pre-trained on a broad dataset and reused across different downstream tasks, enabling efficient and modular training of LDMs on smaller, task-specific datasets. This also significantly reduces the **computational load**, as the diffusion model operates on smaller latent inputs.

2. LDMs are able generate high-quality latents in **fewer diffusion steps** and as the autoencoder reconstructs perceptual details in a **single** forward pass, we get very fast sampling speed.

3. The reduced spatial size of the latent space enables the use of **cross-attention layers** in the diffusion model, which is critical for tasks like **text-to-image generation** (see Applications).

1.4 Guidance and its Applications

While diffusion models are capable of learning the data distribution $p(\mathbf{x})$, in many applications we wish to control the generation process based on additional information, such as text description, class labels, semantic maps and much more. This is achieved using a technique known as *guidance*, which enables conditional generation by modeling $p(\mathbf{x} \mid \mathbf{y})$ for some conditioning variable $\mathbf{y}$.

According to score interpretation of diffusion model, diffusion process can be though of as learning the gradient of likelihood and diffusion process denoise the image **along the gradient** to maximally increase its likelihood.

Under the score-based interpretation of diffusion models, the denoising process can be though of as learning the gradient of likelihood $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ and we simply follow the gradient to denoise image and increase its likelihood. To condition the generation on information $\mathbf{y}$ we learn the conditional score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{y})$ this can be done in the following ways.

1.4.1 Classifier Guidance

When the conditioning variable $\mathbf{y}$ is categorical (e.g., class labels), we can use a pretrained classifier to guide the generation. We can approximate the conditional score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{y})$ as follows:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid y) &= \nabla_{\mathbf{x}_t} \log \left(\frac{p(\mathbf{x}_t)\, p(y \mid \mathbf{x}_t)}{p(y)} \right) \\
&= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(y \mid \mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log p(y) \\
&= \underbrace{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)}_{\text{unconditional score}} + \underbrace{\nabla_{\mathbf{x}_t} \log p(y \mid \mathbf{x}_t)}_{\text{classifier gradient}}
\end{aligned}$$

The term $\nabla_{\mathbf{x}_t} \log p(y \mid \mathbf{x}_t)$ can be computed via backpropagation through a classifier trained on noisy inputs $\mathbf{x}_t$. Adding this to the unconditional score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ allows us to approximate conditional score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid y)$.

In practice, since noise prediction and score learning are **equivalent**, we replace score with corresponding noise and also scale the importance of classifier gradient with γ (**Guidance Scale**).

$$\begin{aligned}
\nabla \log p(\boldsymbol{x}_t \mid y) &= \nabla \log p(\boldsymbol{x}_t) + \gamma \nabla \log p(y \mid \boldsymbol{x}_t) \\
\epsilon(x_t, t, y) &= \epsilon(x_t, t) + \gamma \nabla \log p(y \mid \boldsymbol{x}_t) \text{ (Noise Interpretation)}
\end{aligned}$$

1.4.2 Classifier-Free Guidance

Classifier guidance only works for categorical conditioning and it puts a overhead of training a classifier, and since the model has to classify noisy input the gradient may not always be reliable.

Classifier-Free Guidance (CFG) is a more flexible alternative that works with arbitrary conditioning information, such as text prompts, semantic maps, or blurry sketches. Instead of relying on a separate classifier, the diffusion model itself is trained to support both conditional and unconditional denoising.

During training, the model is occasionally provided with a special null token ϕ (e.g., empty text or zero embedding) in place of the actual condition y. This allows the model to learn:

- $\boldsymbol{\epsilon(\mathbf{x}_t, t, y)}$ — the **conditional** noise prediction
- $\boldsymbol{\epsilon(\mathbf{x}_t, t, \phi)}$ — the **unconditional** noise prediction

At inference, we combine these two predictions to approximate the conditional score using:

$$\begin{aligned}
\nabla \log p(\mathbf{x}_t \mid y) &= \nabla \log p(\mathbf{x}_t) + \gamma \nabla \log p(y \mid \mathbf{x}_t) \\
&= \gamma \nabla \log p(\mathbf{x}_t \mid y) + (1 - \gamma) \nabla \log p(\mathbf{x}_t)
\end{aligned}$$

This corresponds to the following interpolation of noise estimates:

$$\begin{aligned}\epsilon_{\text{guided}}(\mathbf{x}_t, t, y) &= \gamma \cdot \epsilon(\mathbf{x}_t, t, y) + (1-\gamma) \cdot \epsilon(\mathbf{x}_t, t, \phi) \\ &= \epsilon(\mathbf{x}_t, t, \phi) + \gamma \cdot (\epsilon(\mathbf{x}_t, t, y) - \epsilon(\mathbf{x}_t, t, \phi))\end{aligned}$$

This final form is widely used in implementations, where the scalar $\gamma = 0$ corresponds to unconditioned generation and $\gamma > 1$ increases the influence of the conditioning, typically improving sample fidelity at the cost of diversity.

2 Methodology

I implemented text-conditioned image synthesis using Latent Diffusion Models (LDMs) with the Hugging Face `diffusers` library. For the autoencoder, I used the pretrained KL-regularized VAE from Stable Diffusion (`CompVis/stable-diffusion-v1-4`). This autoencoder compresses RGB images of resolution $(3, 256, 256)$ into latent representations of shape $(4, 32, 32)$, achieving an $8\times$ spatial downsampling. For text prompts embedding I used CLIP text encoder (`openai/clip-vit-large-patch14`), which produces suitable embeddings for guiding image generation.

I used the CelebA-Dialog dataset, which contains 30,000 high-resolution CelebA face with captions. Training was performed using a single **NVIDIA P100 GPU** on **Kaggle**. The total training time was approximately 3 hours. The autoencoder and text encoder frozen only LDM was trained.

3 Experiments

Below we present results from the text-to-image generation model. Each grid corresponds to a specific text prompt. Along each row, we vary the guidance scale while keeping the number of inference steps fixed. Along each column, we increase the number of inference steps while keeping the guidance scale fixed. The title of each image indicates its corresponding **guidance scale** and **number of steps**.

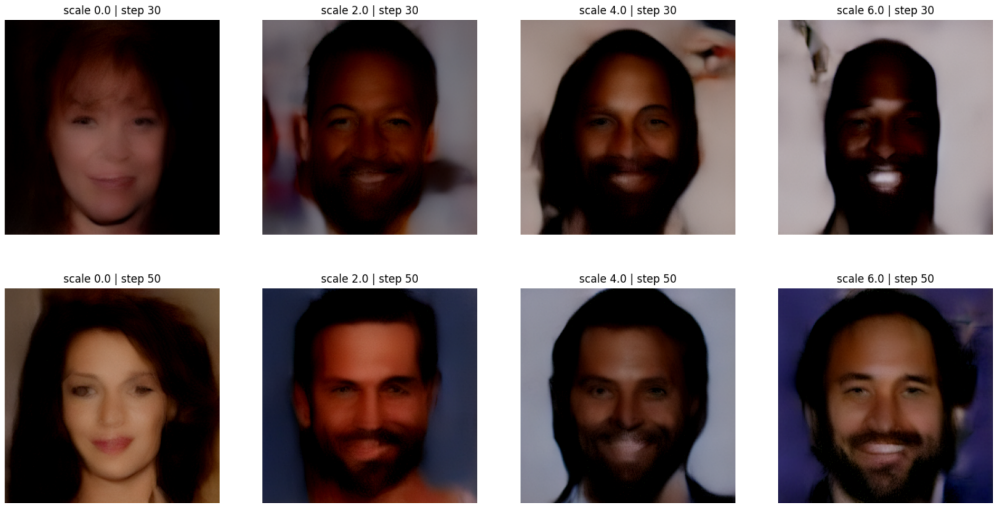


Figure 6: Prompt: *"person in thirties with think beard and has a mild smile"*

As the guidance scale increases, the generated images align more closely with the key attributes in the prompt—particularly *forties*, *beard*, and *smile*—showing the model's ability to focus more on semantic fidelity.

Figure 7: Prompt: *"teenager with pointed beard, glasses and no smile"*

Again, we observe that higher guidance scales lead to images that better reflect the prompt details, such as the presence of *glasses*, *beard*, and *no smile*. Increasing the number of inference steps also improves image quality, especially in terms of sharpness and coherence.

4 Conclusion

In this work, I explored text-conditioned image synthesis framework based on Latent Diffusion Models. By leveraging a pretrained KL-regularized VAE from Stable Diffusion to compress images into an efficient latent space and using a CLIP text encoder for semantic conditioning, the model is capable of generating reasonably good quality images aligned with prompts. The experiments on the CelebA-Dialog dataset shows that increasing the guidance scale leads to images that more closely adhere to key attributes specified in the prompts, demonstrating the success of this approach.

Source Code

The source code used for building and training diffusion models as well as sampling images is available in the below linked GitHub repository. Download links to our trained models are available in the repository.

`GitHub Link`

References

[1] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML]. URL: https://arxiv.org/abs/1406.2661.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[3] Jiaming Song, Chenlin Meng, and Stefano Ermon. *Denoising Diffusion Implicit Models*. 2022. arXiv: 2010.02502 [cs.LG]. URL: https://arxiv.org/abs/2010.02502.

[4] Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2021. arXiv: 2011.13456 [cs.LG]. URL: https://arxiv.org/abs/2011.13456.

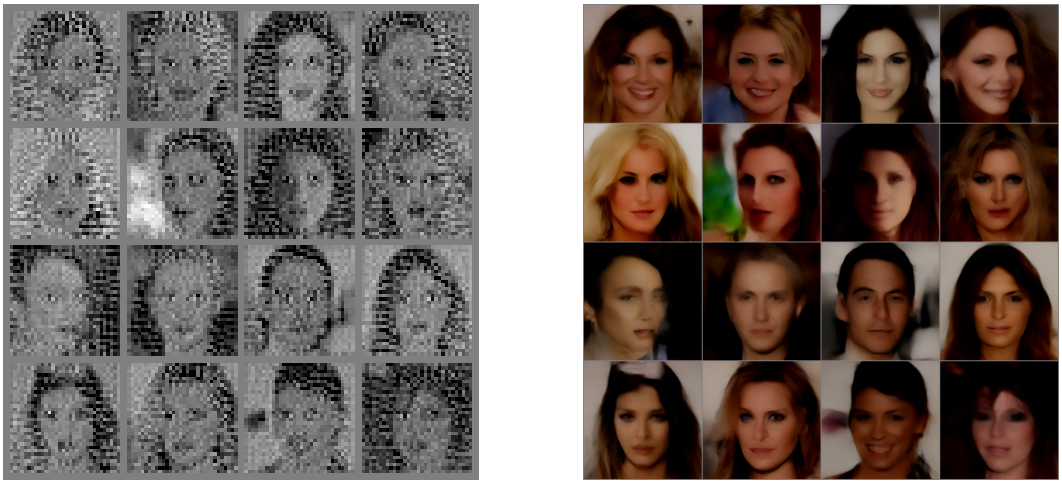
[5] Calvin Luo. *Understanding Diffusion Models: A Unified Perspective*. 2022. arXiv: 2208.11970 [cs.LG]. URL: https://arxiv.org/abs/2208.11970.

[6] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: 2112.10752 [cs.CV]. URL: https://arxiv.org/abs/2112.10752.

[7] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. *Neural Discrete Representation Learning*. 2018. arXiv: 1711.00937 [cs.LG]. URL: https://arxiv.org/abs/1711.00937.

5 Appendix

A Additional Samples



(a) Latent Space Sample (b) Reconstructed Image

Figure 8: Latent Sample and corresponding Reconstruction

The latent sample, originally of shape (4, 32, 32), has been converted to a grayscale image by averaging across the channel dimension. The corresponding reconstruction is shown at full resolution with shape (3, 256, 256).

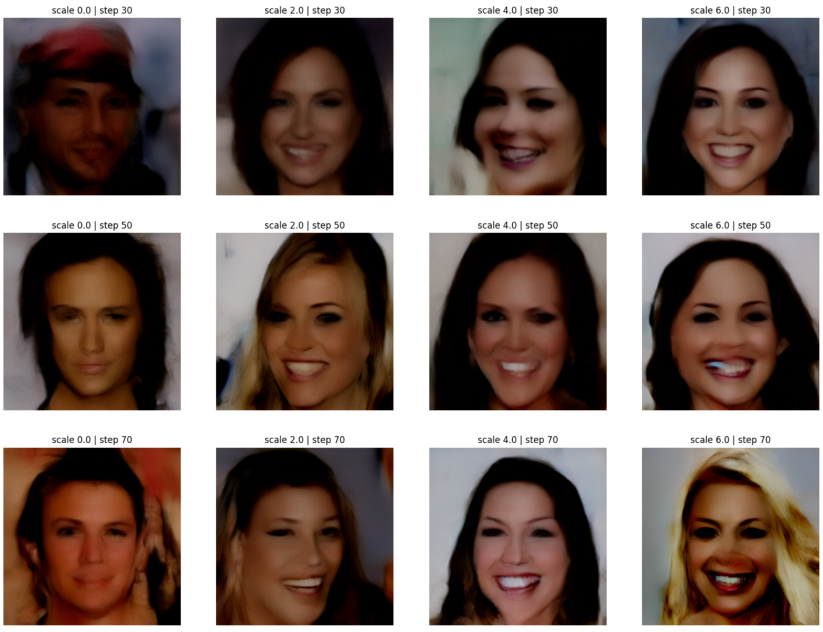


Figure 9: Additional Sample, Prompt :*she has a big smile on her face and has no glasses , and no fringe . she is in the thirties.*

B Rate-Distortion Tradeoff

The ELBO loss for DDPM can be expressed as:

$$\mathbb{E}_q \left[\underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]$$

Here, L_0 is referred to as the **Distortion** term, while the sum $L_1 + L_2 + \cdots + L_T$ constitutes the **Rate**. According to rate-distortion theory, using techniques such as minimal random coding, a sample $\mathbf{x} \sim q(x)$ can be transmitted using approximately $D_{\text{KL}}(q(x) \parallel p(x))$ bits on average, for any distributions p and q. Thus, the rate provides a quantitative measure of how much information about the original clean image is being transmitted.

The authors of DDPM [2] report that their model trained on CIFAR-10 achieves a rate of 1.78 bits/dim and a distortion of 1.97 bits/dim on the test set. This distortion corresponds to a root mean squared error (RMSE) of 0.95 on average.

This indicates that more than half of the lossless code-length learned by the model is devoted to encoding imperceptible distortions.