Flow Matching Variant for Denoising Diffusion Codebook Models

Wenhao Chai

In this blog, we introduce Denoising Diffusion Codebook Models (DDCM) [2] and extend it to the flow matching [1] framework.

1. Introduction.

Denoising Diffusion Codebook Model (DDCM) [2] integrates Denoising Diffusion Models (DDMs) with pre-defined codebooks of fixed Gaussian noise vectors to enable high-quality image generation alongside losslessly compressed bit-stream representations. By substituting the standard Gaussian noise sampling in the reverse diffusion process with selections from compact codebooks, DDCM maintains the sample quality and diversity of traditional DDMs, even with extremely small codebooks. This approach is leveraged for state-of-the-art perceptual image compression, where the codebook entries that best match a given image are encoded into a condensed bit-stream, achieving efficient lossy compression. Beyond compression, the method is extendable to conditional image generation tasks (e.g., inpainting, restoration) by defining noise selection rules tailored to specific conditions, allowing generated outputs to be paired with compact representations.

2. On DDPM

The Denoising Diffusion Probabilistic Model (DDPM) sampling process is defined as:

$$x_{t-1} = \mu(x_t) + \sigma_t \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, I)$$

DDCM modifies this by restricting noise samples to a finite set:

$$x_{t-1} = \mu(x_t) + \sigma_t \varepsilon_t, \quad \varepsilon_t \sim C_t$$

where C_t is a codebook containing K pre-sampled noise vectors from $\mathcal{N}(0, I)$.
Under the condition of known x_0, DDPM's conditional sampling formula becomes:

$$p(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \frac{\alpha_t \bar{\beta}_{t-1}^2}{\bar{\beta}_t^2} x_t + \frac{\bar{\alpha}_{t-1}\beta_t^2}{\bar{\beta}_t^2} x_0, \frac{\bar{\beta}_{t-1}^2 \beta_t^2}{\bar{\beta}_t^2} I\right)$$

This can be rewritten as:

$$x_{t-1} = \mu(x_t) + \frac{\bar{\alpha}_{t-1}\beta_t^2}{\bar{\beta}_t^2}(x_0 - \bar{\mu}(x_t)) + \frac{\bar{\beta}_{t-1}\beta_t}{\bar{\beta}_t}\varepsilon_t$$

For encoding, DDCM selects the optimal noise from the codebook:

$$\varepsilon_t = \arg\max_{\varepsilon \in C_t} \varepsilon \cdot (x_0 - \bar{\mu}(x_t))$$

3. On Flow Matching

Flow Matching is based on an ordinary differential equation (ODE):

$$\frac{dx_t}{dt} = v(x_t, t)$$

where $v(x_t, t)$ is the learned velocity field, with $t \in [0, 1]$, x_0 being the real sample, and x_1 being noise.

Discretization of the Velocity Field To create a Flow Matching version of DDCM, we need to discretize the continuous velocity field:

We discretize the time axis into $t_1, t_2, ..., t_T$, with corresponding states $x_{t_1}, x_{t_2}, ..., x_{t_T}$.

The discretized update rule is:

$$x_{t_{i+1}} = x_{t_i} + \int_{t_i}^{t_{i+1}} v(x_s, s) ds$$

Which can be approximated as:

$$x_{t_{i+1}} \approx x_{t_i} + \Delta t_i \cdot v(x_{t_i}, t_i)$$

Introduction of a Finite Direction Set Similar to DDCM, we introduce a finite set of directions $D_i = \{d_1, d_2, ..., d_K\}$, where each d_k is a unit vector in $\mathbb{R}^d$.

The modified update rule becomes:

$$x_{t_{i+1}} = x_{t_i} + \Delta t_i \cdot s_i \cdot d_i, \quad d_i \in D_i, \quad s_i \in \mathbb{R}^+$$

where d_i is the direction and s_i is the step size.

Conditional Flow Matching For a given x_0, we aim to find the optimal path.

We define a conditional velocity field:

$$v_c(x_t, t, x_0) = v(x_t, t) + \lambda(t)(f(x_0, t) - g(x_t, t))$$

where $f(x_0, t)$ is the target value, $g(x_t, t)$ is the predicted value, and $\lambda(t)$ is a weighting function.

DDCM Encoding Formula in Flow Matching The optimal direction is selected as:

$$d_i = \arg\max_{d \in D_i} d \cdot v_c(x_{t_i}, t_i, x_0)$$

The optimal step size is:

$$s_i = \frac{|v_c(x_{t_i}, t_i, x_0)|}{\Delta t_i}$$

Reconstruction Process Given the encoding sequence $(d_1, d_2, ..., d_T)$, the reconstruction process is:

$$x_{t_{i+1}} = x_{t_i} + \Delta t_i \cdot s_i \cdot d_i$$

starting from x_1 (noise) and iterating until x_0.

Theoretical Approximation Guarantees As $K \to \infty$ (infinite direction set), the discretized version should converge to the original Flow Matching:

$$\lim_{K\to\infty} \text{DDCM-Flow} = \text{Flow Matching}$$

If we introduce importance sampling instead of taking the argmax:

$$p(d) \propto \exp\left(-\frac{1}{2}\left\|d - \frac{v_c(x_{t_i}, t_i, x_0)}{|v_c(x_{t_i}, t_i, x_0)|}\right\|^2\right), \quad d \in D_i$$

In this case, as K increases, the encoding process transitions more smoothly to standard Flow Matching.

4. Conclusion

Adapting DDCM to the Flow Matching framework offers promising directions for efficient image compression and discrete encoding. The theoretical formulation provided here bridges these two approaches, combining the deterministic efficiency of Flow Matching with the discrete representation capability of DDCM. This could potentially lead to faster and more efficient image tokenization methods for multimodal learning systems.

References

[1] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

[2] Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models. *arXiv preprint arXiv:2502.01189*, 2025.