Laplacian-guided Entropy Model in Neural Codec with Blur-dissipated Synthesis Supplementary Material (Appendix)

1. Denosing Diffusion Models

Denoising diffusion models are hierarchical latent variable models which generate sample through gradually removing noise from a randomly sampled white noise vector. The training procedure is comprised of two processes: diffusion or forward and denoising or backward. Diffusion process destroy the clean image and convert it to an approximately pure Gaussian noise during T time steps. The learnable denoising process then reconstructs the data distribution from white noise by reversing the diffusion process.

Diffusion Process: The diffusion process [1] can be described as a Markov chain, wherein each step of the forward path is defined by a Gaussian transition kernel:

$$q(\boldsymbol{z_t}|\boldsymbol{z_{t-1}}) = \mathcal{N}(\boldsymbol{z_t}; \alpha_{t|t-1}\boldsymbol{z_{t-1}}, \sigma^2_{t|t-1}\boldsymbol{I}), \tag{1}$$

where $\alpha_{t|t-1} \in R^+$ governs the extent to which the previous latent is retained, while $\sigma_{t|t-1} \in R^+$ regulates the magnitude of the added noise. The dimension of the latent variables $\boldsymbol{z_1}, ..., \boldsymbol{z_T}$ is the same as that of the data $\boldsymbol{x}$ or $\boldsymbol{z_0}$. An important property of the forward process is that any desired step $\boldsymbol{z_t}$ can be directly sampled from $\boldsymbol{x}$ using a closed-form solution, without needing to compute preceding steps:

$$q(\boldsymbol{z_t}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z_t}; \alpha_t\boldsymbol{x}, \sigma^2_t\boldsymbol{I}), \tag{2}$$

where $\alpha_{t|t-1} = \alpha_t/\alpha_{t-1}$ and $\sigma^2_{t|t-1} = \sigma^2_t - \alpha^2_{t|t-1}\sigma^2_{t-1}$. The pre-specified hyperparameters α_t typically exhibit a monotonically decreasing pattern from 1 to 0, while σ_t monotonically increases from 0 to 1. This pattern leads to a gradual corruption of the input image by Gaussian noise as t increases, resulting in $q(\boldsymbol{z_T}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

Denoising Process: The true denoising distribution, which is tractable when conditioned on $\boldsymbol{x}$ [1], can be written:

$$q(\boldsymbol{z_{t-1}}|\boldsymbol{z_t}, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z_{t-1}}; \boldsymbol{\mu_{t\to t-1}}(\boldsymbol{x}, \boldsymbol{z_t}), {\sigma^2}_{t\to t-1}\boldsymbol{I}), \tag{3}$$

where the distribution parameters can be computed as:

$$\begin{aligned} &\sigma_{t\to t-1} = \sigma_{t|t-1}\sigma_{t-1}/\sigma_t \\ &\boldsymbol{\mu_{t\to t-1}} = (\alpha_{t|t-1}\sigma^2_{t-1}/\sigma^2_t)\boldsymbol{z_t} + (\alpha_{t-1}\sigma^2_{t|t-1}/\sigma^2_t)\boldsymbol{x} \end{aligned} \tag{4}$$

To generate data, the true denoising process can be estimated by a learned denoising distribution $p_\theta(\boldsymbol{z_{t-1}}|\boldsymbol{z_t}) := q(\boldsymbol{z_{t-1}}|\boldsymbol{z_t}, \hat{\boldsymbol{x}} = \phi_\theta(\boldsymbol{z_t}, t))$, where $\hat{\boldsymbol{x}}$ is predicted from diffused sample $\boldsymbol{z_t}$ using a neural network ϕ_θ. Similar to Eq. 4, $p_\theta(\boldsymbol{z_{t-1}}|\boldsymbol{z_t})$ can be expressed by the approximation $\hat{\boldsymbol{x}}$:

$$\begin{aligned} p_\theta(\boldsymbol{z_{t-1}}|\boldsymbol{z_t}) &= \mathcal{N}(\boldsymbol{z_{t-1}}; \mu_\theta(\boldsymbol{z_t}, \boldsymbol{t}), {\sigma^2}_{t\to t-1}\boldsymbol{I}), \\ &= \mathcal{N}(\boldsymbol{z_{t-1}}; \boldsymbol{\mu_{t\to t-1}}(\hat{\boldsymbol{x}}, \boldsymbol{z_t}), {\sigma^2}_{t\to t-1}\boldsymbol{I}). \end{aligned} \tag{5}$$

Training Objective: The likelihood function $\log p_\theta(\boldsymbol{x})$ is challenging to compute directly for training the model. So, during training, its evidence lower bound is maximized (ELBO $\leq \log p_\theta(\boldsymbol{x})$), which can be expressed as:

$$\begin{aligned} \text{ELBO} = \boldsymbol{E}_q[&-\overbrace{D_{KL}(q(\boldsymbol{z_T}|\boldsymbol{x})||p(\boldsymbol{z_T}))}^{L_T} \overbrace{+\log p_\theta(\boldsymbol{x}|\boldsymbol{z_1})}^{L_0} \\ &+ \sum_{t=2}^{T} -\overbrace{D_{KL}(q(\boldsymbol{z_{t-1}}|\boldsymbol{z_t}, \boldsymbol{x})||p_\theta(\boldsymbol{z_{t-1}}|\boldsymbol{z_t})}^{L_{t-1}}]. \end{aligned} \tag{6}$$

Within a well-defined noise scheduling, both L_0 and L_T tend to approach approximately 0 and remain constant. Therefore, for training the diffusion model, it becomes adequate to optimize the L_{t-1} term, which is equivalent to comparing the learnable denoising process with the true denoising distribution. As both of these distributions are Gaussian, the expressions for the KL divergences have closed-form solutions and can be written as follows:

$$L_{t-1} \propto \boldsymbol{E}_q[||\boldsymbol{\mu_{t\to t-1}} - \mu_\theta(\boldsymbol{z_t}, \boldsymbol{t})||^2] = \boldsymbol{E}_q[||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2]. \tag{7}$$

In above formulation, the neural network directly predicts $\hat{\boldsymbol{x}}$. However, [1] discovered that optimization becomes simpler by predicting Gaussian noise instead. Hence, if we express $\boldsymbol{z_t} = \alpha_t\boldsymbol{x} + \sigma_t\boldsymbol{\epsilon}$, then the neural network ϕ_θ generates $\hat{\boldsymbol{\epsilon}} = \phi_\theta(\boldsymbol{z_t}, t)$, resulting in:

$$\hat{\boldsymbol{x}} = (1/\alpha_t)\boldsymbol{z_t} - (\sigma_t/\alpha_t)\hat{\boldsymbol{\epsilon}}. \tag{8}$$

As demonstrated in [2], using this specific parameterization, the final loss is obtained as follows:

$$\boldsymbol{E}_{t,\boldsymbol{x},\boldsymbol{\epsilon}}[||\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}||^2] = \boldsymbol{E}_{t,\boldsymbol{x},\boldsymbol{\epsilon}}[||\boldsymbol{\epsilon} - \phi_\theta(\alpha_t\boldsymbol{x} + \sigma_t\boldsymbol{\epsilon}, t)||^2. \tag{9}$$

2. Additional Details on Blurring Diffusion Model

Heat Dissipation as Gaussian Diffusion: The heat dissipation process or blurring [4] can be expressed as a type of Gaussian diffusion. First, the marginal distribution of any time step noisy latent $\boldsymbol{z_t}$ can be defined as follows:

$$q(\boldsymbol{z_t}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z_t}; \boldsymbol{A_t}\boldsymbol{x}, \sigma^2\boldsymbol{I}), \tag{10}$$

where $\boldsymbol{A_t} = \boldsymbol{V}\boldsymbol{D_t}\boldsymbol{V^T}$ represents the dissipation or blurring operation. $\boldsymbol{V^T}$ contains orthogonal Discrete Cosine Transform (DCT) basis, while the diagonal matrix $D_t = \exp(-\boldsymbol{\Lambda}\tau_t)$ corresponds to the exponentiation of a weighting matrix for the frequencies $\boldsymbol{\Lambda}$. $\boldsymbol{\Lambda}$ contains squared frequencies $\lambda_{n,m} = -\pi^2(n^2/W^2 + m^2/H^2)$, where W and H are the width and height of the image, and $n \in \{0, \dots, W-1\}$ and $m \in \{0, \dots, H-1\}$. According to Eq. 10, any latent state $\boldsymbol{z_t}$ is created by introducing a constant level of noise to a progressively blurred data point. When we transform the variables using the following transformations: $\boldsymbol{f_t} = \boldsymbol{V^T}\boldsymbol{z_t}$ and $\boldsymbol{f_x} = \boldsymbol{V^T}\boldsymbol{x}$, the Gaussian diffusion process can be formulated in frequency space:

$$\begin{aligned} &q(\boldsymbol{V^T}\boldsymbol{z_t}|\boldsymbol{V^T}\boldsymbol{x}) = \mathcal{N}(\boldsymbol{V^T}\boldsymbol{z_t}; \boldsymbol{V^T}\boldsymbol{A_t}\boldsymbol{x}, \sigma^2\ \boldsymbol{V^T}\boldsymbol{I}\boldsymbol{V}) \Leftrightarrow \\ &q(\boldsymbol{f_t}|\boldsymbol{f_x}) = \mathcal{N}(\boldsymbol{f_t}; \boldsymbol{D_t}\boldsymbol{f_x}, \sigma^2\ \boldsymbol{I}). \end{aligned} \tag{11}$$

If we define a vector $\boldsymbol{\lambda}$ containing the diagonal elements of $\boldsymbol{\Lambda}$, we can express $\boldsymbol{d_t}$ as $\exp(-\boldsymbol{\lambda}\tau_t)$, which corresponds to the diagonal elements of the matrix $\boldsymbol{D_t}$. With this reinterpretation, the diffusion process in frequency space can be written as follows:

$$q(\boldsymbol{f_t}|\boldsymbol{f_x}) = \mathcal{N}(\boldsymbol{f_t}; \boldsymbol{d_t} \odot \boldsymbol{f_x}, \sigma^2\boldsymbol{I}), \tag{12}$$

where $\odot$ denotes elementwise vector multiplication. Eq. 12 shows that the marginal distribution of $\boldsymbol{f_t}$ can be decomposed into individual scalar elements $\boldsymbol{f_t^{(i)}}$. Likewise, the learnable inverse heat dissipation model $p_\theta(\boldsymbol{f_{t-1}}|\boldsymbol{f_t})$ can also be decomposed in a fully factorized manner. As a result, we have the option to describe the heat dissipation process and its inverse using scalar representations for each dimension i:

$$\begin{aligned} &q(f_t^{(i)}|f_x^{(i)}) = \mathcal{N}(f_t^{(i)}; d_t^{(i)}u_x^{(i)}, \sigma^2) \Leftrightarrow \\ &f_t^{(i)} = d_t^{(i)}f_x^{(i)} + \sigma\epsilon, \text{with} \epsilon \sim \mathcal{N}(0, 1). \end{aligned} \tag{13}$$

Eq. 13 can be identified as a particular case of the standard Gaussian diffusion process that operates in frequency space, i.e., $f_t^{(i)} = \alpha_t f_x^{(i)} + \sigma_t\epsilon$, where $\alpha_t = d_t^{(i)}$ and $\sigma_t = \sigma$. What distinguishes this type of diffusion process from the standard one is the utilization of distinct noise schedules, denoted as α_t and σ_t, for each scalar element of the latent variable $\boldsymbol{f_t}$. In other words, the noise applied in this process exhibits non-isotropic characteristics. It's worth noting that while the marginal variance σ is shared across all scalar elements $f_t^{(i)}$, the specific noise schedules provide individual adjustments for each element.

In heat dissipation models, the Markov process $q(\boldsymbol{f_t}|\boldsymbol{f_{t-1}})$ can be defined, corresponding to their chosen marginal distribution $q(\boldsymbol{f_t}|\boldsymbol{f_x})$. By establishing an equivalence with Gaussian diffusion, this process can be effectively described using the following formulation:

$$\begin{aligned} &q(\boldsymbol{f_t}|\boldsymbol{f_{t-1}}) = \mathcal{N}(\boldsymbol{f_t}; \boldsymbol{\alpha_{t|t-1}}\boldsymbol{f_{t-1}}, \boldsymbol{\sigma^2_{t|t-1}}\boldsymbol{I}), \\ &\text{where } \boldsymbol{\alpha_t} = \boldsymbol{d_t}, \sigma_t^{(i)} = \sigma \Rightarrow \boldsymbol{\alpha_{t|t-1}} = \frac{\boldsymbol{d_t}}{\boldsymbol{d_{t-1}}}, \\ &\Rightarrow \boldsymbol{\sigma^2_{t|t-1}} = (1 - (\frac{\boldsymbol{d_t}}{\boldsymbol{d_{t-1}}})^2)\sigma^2. \end{aligned} \tag{14}$$

When $\boldsymbol{d_t}$ is designed to have smaller values for higher frequencies, $\boldsymbol{\sigma_{t|t-1}}$ will introduce greater noise to the higher frequencies at each timestep. This results in the heat dissipation model erasing information from those frequencies more rapidly compared to the standard diffusion process.

Inverse Heat Dissipation: Similar to the standard diffusion model [1], the analytical expression for the true inverse heat dissipation process is obtained and can be written as follows:

$$q(\boldsymbol{f_{t-1}}|\boldsymbol{f_t}, \boldsymbol{f_x}) = \mathcal{N}(\boldsymbol{f_{t-1}}; \boldsymbol{\mu_{t\to t-1}}, \boldsymbol{\sigma^2}_{t\to t-1}\boldsymbol{I}), \tag{15}$$

where:

$$\begin{aligned} &q(\boldsymbol{f_{t-1}}|\boldsymbol{f_t}, \boldsymbol{f_x}) \propto q(\boldsymbol{f_{t-1}}|\boldsymbol{f_x})q(\boldsymbol{f_t}|\boldsymbol{f_{t-1}}, \boldsymbol{f_x}) = \\ &q(\boldsymbol{f_{t-1}}|\boldsymbol{f_x})q(\boldsymbol{f_t}|\boldsymbol{f_{t-1}}) \Rightarrow \boldsymbol{\sigma_{t\to t-1}} = \boldsymbol{\sigma_{t|t-1}}\boldsymbol{\sigma_{t-1}}/\boldsymbol{\sigma_t}, \\ &\boldsymbol{\mu_{t\to t-1}} = (\boldsymbol{\alpha_{t|t-1}}\boldsymbol{\sigma^2_{t-1}}/\boldsymbol{\sigma^2_t})\boldsymbol{f_t} + (\boldsymbol{\alpha_{t-1}}\boldsymbol{\sigma^2_{t|t-1}}/\boldsymbol{\sigma^2_t})\boldsymbol{f_x} \end{aligned} \tag{16}$$

As discussed, the true denoising process can be approximated using a learned denoising distribution, $p_\theta(\boldsymbol{f_{t_1}}|\boldsymbol{f_t})$.

3. Algorithms

Algorithms 1 and 2 summarize the training and decoding procedures of our neural codec.

4. Architecture of Diffusion-based Decoder

Fig. 1 illustrates our diffusion-based decoder design, employing a U-Net architecture for the diffusion model [1], incorporating ResNet blocks and self-attention modules. We've employed six units for both encoding and decoding within the U-Net framework. In the encoding pathway, the channel dimension is determined from the set

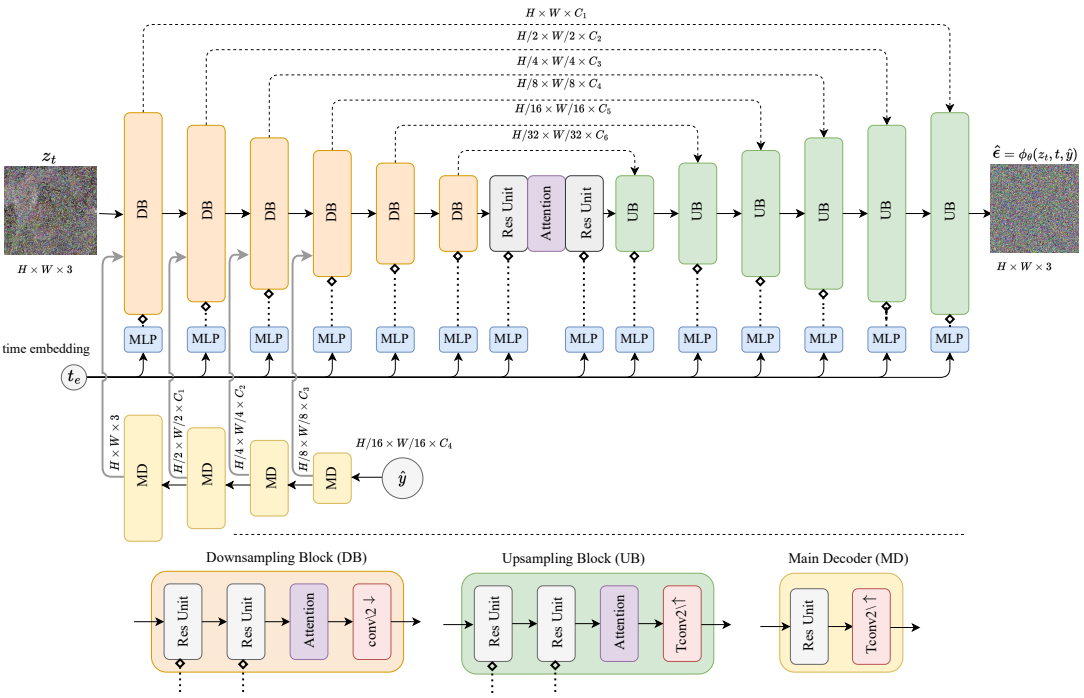

Figure 1. Architect of diffusion-based decoder. W and H correspond to the width and height of the input image, respectively.

Algorithm 1 Training Neural Codec

Sample $\boldsymbol{x}$ ∼dataset
repeat
  Sample $t \sim \mathcal{U}(0, T)$
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{z_t} = \boldsymbol{V}\boldsymbol{\alpha_t}\boldsymbol{V^T}\boldsymbol{x} + \boldsymbol{V}\boldsymbol{\sigma_t}\boldsymbol{V^T}\boldsymbol{\epsilon}$
  $\tilde{\boldsymbol{y}} \sim \mathcal{U}(En_\zeta(\boldsymbol{x}) - 0.5, En_\zeta(\boldsymbol{x}) + 0.5)$
  $\hat{\boldsymbol{x}}_t = \boldsymbol{V}(1/\boldsymbol{\alpha_t})(\boldsymbol{V^T}\boldsymbol{z_t} - \boldsymbol{\sigma_t}\boldsymbol{V^T}\phi_\theta(\boldsymbol{z_t}, t, \tilde{\boldsymbol{y}}))$
  $L_{Dif} = ||\boldsymbol{\epsilon} - \phi_\theta(\boldsymbol{z_t}, t, \tilde{\boldsymbol{y}})||^2$
  $L_T = (1-\beta)L_{Dif} + \beta d_{\text{LPIPS}}(\boldsymbol{x}, \hat{\boldsymbol{x}}_t) - \lambda \log p_{\tilde{\boldsymbol{y}}}(\tilde{\boldsymbol{y}})$
  $(\zeta, \theta) = (\zeta, \theta) - \eta \nabla_{\zeta,\theta} L_T$ (η : Learning Rate)
until converged

Algorithm 2 Decoding Compressed File

$\hat{\boldsymbol{y}} \Leftarrow$ Entropy decoded binary file using entropy model $p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}})$
Sample $\boldsymbol{z_T} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, ..., 1$ **do**
  $\boldsymbol{f_t} = \boldsymbol{V^T}\boldsymbol{z_t}$ and $\boldsymbol{f_{\hat{\epsilon}}} = \boldsymbol{V^T}\phi_\theta(\boldsymbol{z_t}, t, \hat{\boldsymbol{y}})$
  $\boldsymbol{\sigma_{t\to t-1}} = \boldsymbol{\sigma_{t|t-1}}\boldsymbol{\sigma_{t-1}}/\boldsymbol{\sigma_t}$
  $\hat{\boldsymbol{\mu}}_{\boldsymbol{t\to t-1}} = \frac{\boldsymbol{\alpha_{t|t-1}}\boldsymbol{\sigma^2_{t-1}}}{\boldsymbol{\sigma^2_t}}\boldsymbol{f_t} + \frac{\boldsymbol{\sigma^2_{t|t-1}}}{\boldsymbol{\alpha_{t|t-1}}\boldsymbol{\sigma^2_t}}(\boldsymbol{f_t} - \boldsymbol{\sigma_t}\boldsymbol{f_{\hat{\epsilon}}})$
  $\boldsymbol{z_{t-1}} \leftarrow \boldsymbol{V}(\hat{\boldsymbol{\mu}}_{\boldsymbol{t\to t-1}} + \boldsymbol{\sigma_{t\to t-1}}\boldsymbol{f_{\hat{\epsilon}}})$
end for
Return $\hat{\boldsymbol{x}} = \boldsymbol{z_0}$

$\{C_1 = 64, C_2 = 128, C_3 = 192, C_4 = 256, C_5 = 320, C_6 = 384\}$. The decoding process mirrors the encoding process in reverse. The main decoder (MD) comprises ResNet blocks and transposed convolutions, which serve to upscale the quantized latent representation $\hat{\boldsymbol{y}}$ to match the spatial dimensions of the inputs from the initial 4 U-Net encoding units. This setup enables us to introduce conditioning by concatenating the output of the main decoder layers with the input from the corresponding U-Net layer.

The time step t is initially linearly embedded into a vector with a dimension of 64. Subsequently, the resulting time embedding t_e is further processed through MLP layers, which are responsible for expanding it to align with the channel size of the corresponding DB/UB layers.

5. Additional Qualitative Comparisons

As shown in Fig. 2, our model tends to generate fewer artifacts and is capable of decoding images with greater realism compared to both the HiFiC [3] and CDC [5] networks, even when using a significantly lower bit-rate.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[2] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan

Figure 2. Additional Visual comparison of our method to the HiFiC and CDC models.

Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021. 1

[3] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33:11913–11924, 2020. 3

[4] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. *arXiv preprint arXiv:2206.13397*, 2022. 2

[5] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *arXiv preprint arXiv:2209.06950*, 2022. 3