A. Details on Multinomial Diffusion Models

Definition of Q_t with mask-and-replace strategy. Following mask-and-replace strategy as:

$$\boldsymbol{Q}_t = \begin{bmatrix} \alpha_t + \beta_t & \beta_t & \beta_t & \cdots & 0 \\ \beta_t & \alpha_t + \beta_t & \beta_t & \cdots & 0 \\ \beta_t & \beta_t & \alpha_t + \beta_t & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_t & \gamma_t & \gamma_t & \cdots & 1 \end{bmatrix}, \quad (7)$$

given $\alpha_t \in [0, 1], \beta_t = (1 - \alpha_t - \gamma_t)/K$ and γ_t the probability of a token to be replaced with a `[MASK]` token.

Cumulative transition matrix. The cumulative transition matrix $\overline{\boldsymbol{Q}}_t$ and $q(x_t|x_0)$ can be computed via closed form:

$$\overline{\boldsymbol{Q}}_t \boldsymbol{v}(x_0) = \bar{\alpha}_t \boldsymbol{v}(x_0) + \left(\bar{\gamma}_t - \bar{\beta}_t\right) \boldsymbol{v}(K+1) + \bar{\beta}_t \mathbb{1}, \quad (8)$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \bar{\gamma}_t = 1 - \prod_{i=1}^{t} (1 - \gamma_i)$, and $\bar{\beta}_t = (1 - \bar{\alpha}_t - \bar{\gamma}_t)/(K+1)$ can be calculated and stored in advance.

B. Analysis on Mutual Information

Here we provide an analysis to quantify the amount of information encoded in latent. Since the inversion involves model forward function call which is difficult to analyze. We describe in the following a simple yet prototypical example of DDPM, where the posterior mean can be computed in closed-form thus allows us to compute the mutual information.

Remark B.1. *Given a simple Gaussian DDPM with $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, latents $\{\boldsymbol{z}_t\}$ are obtained with DDPM inversion [26], then the mutual information between $\boldsymbol{z}_t$ and $\boldsymbol{x}_0$:*

$$I(\boldsymbol{z}_t; \boldsymbol{x}_0) = \frac{D}{2} \log\left(\frac{\beta_t^2 \overline{\alpha}_{t-1} + 1 - \overline{\alpha}_{t-1} + \alpha_t(1 - \overline{\alpha}_t)}{1 - \overline{\alpha}_{t-1} + \alpha_t(1 - \overline{\alpha}_t)}\right). \quad (9)$$

The mutual information between $\boldsymbol{z}_t$ and $\boldsymbol{x}_0$ is illustrated in Supplementary Materials. We observe that the amount of information encoded from $\boldsymbol{x}_0$ into $\boldsymbol{z}_t$ decreases as t increases, motivating us to explore different scheduling strategies for λ's.

Proof. We assumed that $\boldsymbol{x}_0$ satisfies standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_D)$. Since

$$\boldsymbol{x}_t = \sqrt{\alpha_t} \boldsymbol{x}_{t-1} + \sqrt{1 - \alpha_t} \boldsymbol{\epsilon}_t$$

where both $\boldsymbol{x}_{t-1}$ and $\boldsymbol{\epsilon}_t$ are independent standard Gaussian random variables, $\boldsymbol{x}_t$ is also standard Gaussian, and in each dimension

$$Cov(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) = \sqrt{\alpha_t},$$

which leads to

$$\hat{\mu}_t(\boldsymbol{x}_t) = \mathbb{E}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \sqrt{\alpha_t} \boldsymbol{x}_t.$$

Therefore,

$$\begin{aligned} \boldsymbol{z}_t &= \boldsymbol{x}'_{t-1} - \hat{\mu}_t(\boldsymbol{x}_t) \\ &= (\sqrt{\overline{\alpha}_{t-1}} \boldsymbol{x}_0 + \sqrt{1 - \overline{\alpha}_{t-1}} \boldsymbol{\epsilon}) \\ &\quad - \sqrt{\alpha_t}(\sqrt{\overline{\alpha}_t} \boldsymbol{x}_0 + \sqrt{1 - \overline{\alpha}_t} \boldsymbol{\epsilon}') \\ &= \beta_t \cdot \sqrt{\overline{\alpha}_{t-1}} \boldsymbol{x}_0 + \sqrt{1 - \overline{\alpha}_{t-1}} \boldsymbol{\epsilon} + \sqrt{\alpha_t(1 - \overline{\alpha}_t)} \boldsymbol{\epsilon}'. \end{aligned}$$

Let

$$E = \sqrt{1 - \overline{\alpha}_{t-1}} \boldsymbol{\epsilon} + \sqrt{\alpha_t(1 - \overline{\alpha}_t)} \boldsymbol{\epsilon}'$$

which is a Gaussian error term independent to $\boldsymbol{x}_0$ with mean 0 and variance $1 - \overline{\alpha}_{t-1} + \alpha_t(1 - \overline{\alpha}_t)$. Thus we can calculate the mutual information

$$\begin{aligned} I(\boldsymbol{z}_t; \boldsymbol{x}_0) &= H(\boldsymbol{z}_t) - H(\boldsymbol{z}_t|\boldsymbol{x}_0) \\ &= H(\boldsymbol{z}_t) - H(E) \\ &= \frac{D}{2} \log(2\pi e(\beta_t^2 \overline{\alpha}_{t-1} + 1 - \overline{\alpha}_{t-1} + \alpha_t(1 - \overline{\alpha}_t)) \\ &\quad - \frac{D}{2} \log(2\pi e(1 - \overline{\alpha}_{t-1} + \alpha_t(1 - \overline{\alpha}_t)) \\ &= \frac{D}{2} \log\left(\frac{\beta_t^2 \overline{\alpha}_{t-1} + 1 - \overline{\alpha}_{t-1} + \alpha_t(1 - \overline{\alpha}_t)}{1 - \overline{\alpha}_{t-1} + \alpha_t(1 - \overline{\alpha}_t)}\right). \end{aligned}$$

□

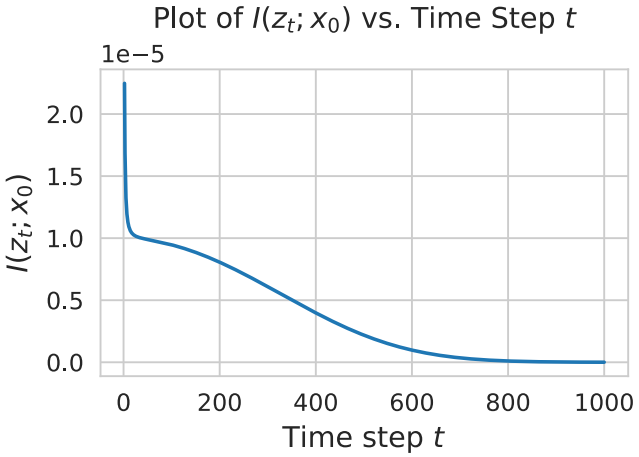

Figure 4. Mutual information between $\boldsymbol{z}_t$ and $\boldsymbol{x}_0$. Computed with a simple DDPM setting by assuming $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

We also provide the relationship between the mutual information of z_t, z_0 and the timestep t in Figure 4.

C. Implementation Details

For all reconstruction task, we employ a $\tau = 1.0$ and $\lambda_1 = 1.0, \lambda_2 = 0.0$ with 32 sampling steps and 26 renoising steps.

The hyper-parameters for Paella editing experiment is CFG$=10.0$, $\lambda_1 = 0.7$, $\lambda_2 = 0.3$ and $\tau = 0.9$. The hyper-parameters for VQ-Diffusion in editing is CFG$=5.0$, $\lambda_1 = 0.2$, $\lambda_2 = 0.8$.

For sentiment editing task with RoBERTa, we utilize two sets of hyperparameter: $\tau = 0.7$, $\lambda_1 = 0.2$, $\lambda_2 = 0.8$ and $\tau = 0.7$, $\lambda_1 = 0.25$, $\lambda_2 = 0.75$.

All models are implemented in PyTorch 2.0 and inferenced on a single NVIDIA A100 40GB.

D. Ablation Studies

D.1. Noise Injection Function

Addition. In the main text we have adopted the *addition* function as noise injection function,

$$\tilde{\boldsymbol{y}} = \log(\boldsymbol{\pi}) + \lambda_1 \cdot \boldsymbol{z} + \lambda_2 \cdot \boldsymbol{g}.$$

This is a natural form inspired by the Gumbel-Max trick: thinking of $\lambda_1 \cdot \boldsymbol{z}$ as a correction term, then $\log(\boldsymbol{\pi}) + \lambda_1 \cdot \boldsymbol{z}$ is the corrected logit and λ_2 is the inverse of temperature of the logit to control the sharpness of the resulting categorical distribution, as

$$\begin{aligned} &\arg\max\left(\log(\boldsymbol{\pi}) + \lambda_1 \cdot \boldsymbol{z} + \lambda_2 \cdot \boldsymbol{g}\right) \\ =&\arg\max\left(\frac{1}{\lambda_2}\left(\log(\boldsymbol{\pi}) + \lambda_1 \cdot \boldsymbol{z}\right) + \boldsymbol{g}\right), \quad \lambda_2 > 0. \end{aligned}$$

λ_1 then controls how much correction we would like to introduce in the original logit.

Variance preserving. From another perspective, $\boldsymbol{z}$ is the artificial "Gumbel" noise that could have been sampled to realize the target tokens. Then, if we treat $\boldsymbol{z}$ as Gumbel noise and want to perturb it with random Gumbel noise, addition does not result in a Gumbel distribution. One way is to approximate this sum with another Gumbel distribution. If $G_1 \sim \text{Gumbel}(\mu_1, \beta_1)$, $G_2 \sim \text{Gumbel}(\mu_2, \beta_2)$ and $G = \lambda_1 G_1 + \lambda_2 G_2$, then the moment matching *Gumbel approximation* for G is

$$\begin{aligned} &\text{Gumbel}(\mu_G, \beta_G), \quad \text{with} \\ &\beta_G = \sqrt{\lambda_1^2\beta_1^2 + \lambda_2^2\beta_2^2}, \\ &\mu_G = \lambda_1\mu_1 + \lambda_2\mu_2 + \gamma(\lambda_1\beta_1 + \lambda_2\beta_2 - \beta_G), \end{aligned}$$

where $\gamma \approx 0.5772$ is the Euler-Mascheroni constant. We consider the *variance preserving* form:

$$\tilde{\boldsymbol{y}} = \log(\boldsymbol{\pi}) + \sqrt{\lambda_1} \cdot \boldsymbol{z} + \sqrt{\lambda_2} \cdot \boldsymbol{g}, \quad \lambda_1 + \lambda_2 = 1.$$

Max. The third way is inspired by the property of Gumbel distribution [57], that if G_1, G_2 are iid random variables following Gumbel(μ, β) then $\max\{G_1, G_2\} - \beta\log 2$ follows the same distribution. We also consider the *max* function for noise injection:

$$\tilde{\boldsymbol{y}} = \log(\boldsymbol{\pi}) + \max\{\lambda_1 \cdot \boldsymbol{z}, \lambda_2 \cdot \boldsymbol{g}\}.$$

D.2. Hyperparameter Search

In this section, we analyze the impact of varying hyperparameters $\lambda_1, \lambda_2, \tau$, and CFG scale on the quality of image generation and adherence to textual descriptions, quantified through Structure Distance and CLIP similarity. The hyperparameters play specific roles: λ controls the amount of noise introduced in each reverse step, τ governs the percentage of tokens replaced with random tokens during inversion, and Classifier-Free Guidance (CFG) scales the influence of the text prompt during image synthesis. To limit the search space and simplify the ablation, we choose $\lambda_1 = \lambda$ and $\lambda_2 = 1 - \lambda$ and vary the value of λ. Evaluation metrics are given in Figure 5.

Effect of λ_1 and λ_2: With a fixed CFG of 10.0, the graphs indicate that increasing λ results in a rise in Structure Distance, suggesting a decline in structural integrity of the images. This increase in noise appears to allow for greater exploration of the generative space at the expense of some loss in image clarity.

Effect of τ: Higher τ values, particularly at 0.9, show a notable rise in Structure Distance as CLIP similarity increases. This implies that more token replacement can lead to images that align better with the text prompts but may suffer in maintaining structural fidelity, likely due to $\boldsymbol{x}_T$ contains less information of the original image while λ injects additional noise during editing phase.

Effect of CFG Scale: Varying CFG at a fixed λ of 0.7 and τ of 0.9 reveals that higher CFG values substantially improve Structure Distance, but to an extent (CFG of 10). Beyond this point, further increases in CFG do not yield significant improvements in structural quality, indicating a diminishing return on higher guidance levels. This plateau suggests that while increasing CFG helps in aligning the generated images more closely with the text prompts initially, the benefits in structural integrity and clarity become less visible as CFG values exceed a certain threshold. This finding underscores the need for a balanced approach in setting CFG, where too much guidance may not necessarily lead to better outcomes in terms of image quality and fidelity to the textual description.

Effect of noise injection function: We also conducted evaluations using a variance-preserving noise injection function by setting $\lambda_1 = \sqrt{\lambda}$ and $\lambda_2 = \sqrt{1-\lambda}$. The results of these experiments are presented in Figure 6. As for the `max` function, we performed a manual inspection of the visual examples generated with this function. The quality of these examples was noticeably inferior, we therefore omit the corresponding evaluation curves from our analysis.

In conclusion, this ablation study demonstrates that increasing λ and τ can enhance adherence to text prompts through broader explorations in generative spaces, yet this benefit is offset by a decrease in the structural quality of the images. On the other hand, raising CFG values enhances

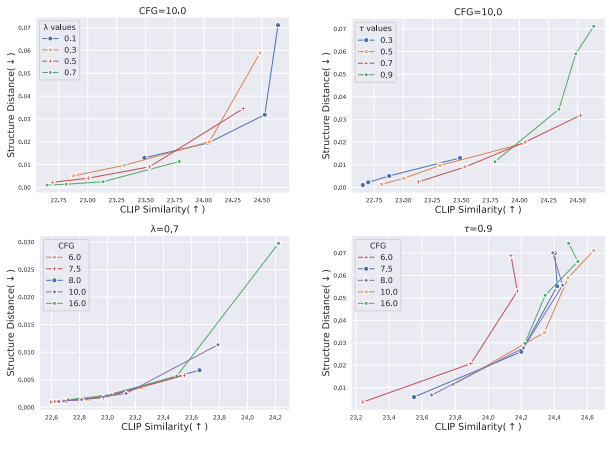

Figure 5. **The effect of hyperparameters $\lambda_1, \lambda_2, \tau$, CFG on the Structure Distance (↓) and CLIP similarity (↑) with addition function as noise inject function.** In our implementation, to limit the search space, we choose $\lambda_1 = \lambda$ and $\lambda_2 = 1 - \lambda$ for simplicity.

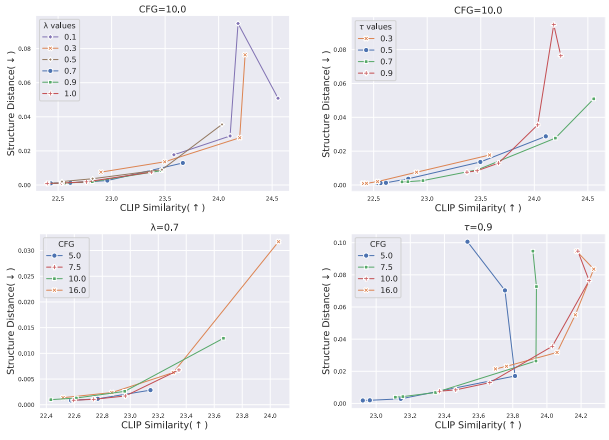

Figure 6. **The effect of hyperparameters λ_1, λ_2 with variance preserving scheme.** We set $\lambda_1 = \sqrt{\lambda}$ and $\lambda_2 = \sqrt{1-\lambda}$.

the structural integrity of images to a certain threshold, after which the improvements plateau, indicating a ceiling to the effectiveness of higher CFG settings. This analysis offers empirical guidance for selecting hyperparameters, balancing the trade-offs between text alignment and image quality to optimize image synthesis outcomes.

E. Additional Results on Image Editing

Reconstruction result with Paella. In Figure 8 we demonstrates the inversion reconstruction result with Paella using our proposed method.

Image editing with diversity. As shown in Figure 10, our method enables diverse image editing results through stochastic variation. The first three rows demonstrate the impact of varying both the inversion masks and the injected Gumbel noise, while the last two rows focus on variations

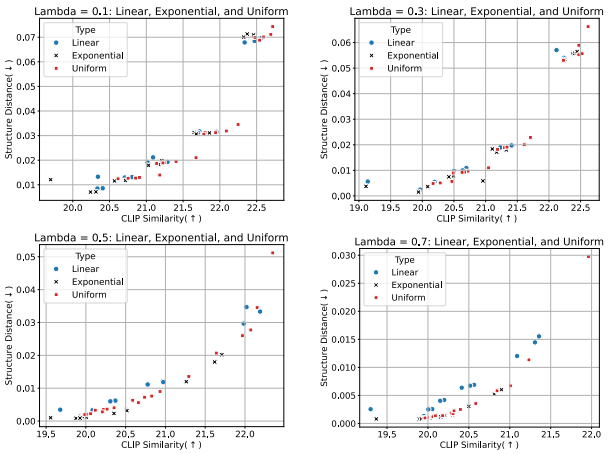

Figure 7. **The effect of different λ schedule on the Structure Distance (↓) and CLIP similarity (↑).** In our implementation, to limit the search space, we choose $\lambda_1 = \lambda$ and $\lambda_2 = 1 - \lambda$ for simplicity.

produced by changing only the inversion masks.

Additional baselines. We compare with SDEdit [36] and ControlNet [61][1]. Results are shown in Figure 11 and Table 8.

Noise injection functions. We compare various noise injection functions, including taking the maximum of Gumbel noise and the recorded noise, as well as the variance-preserving noise injection function.

Mask schedule functions. In Figure 12, we present four types of mask scheduling functions: (a, c) concave up and (b, d) concave down. Our results indicate that concave up mask scheduling functions perform better than their concave down counterparts. Quantitative results are shown in Table 9.

Comparison between inclusive and random masks. To understand the impact of randomness in the masking schedule, we illustrate masks that are inclusive compared to totally random. Inclusive mask is mask schedule that are increasingly growing, which is used in Paella, compared to randomly sampled masks.

F. Details on Text Editing Experiments

Dataset generation. To generate the dataset, we utilize ChatGPT-4o with the following prompt:

[1]We use the ControlNet-InPaint model based on Stable Diffusion v1.5: https://github.com/mikonvergence/ControlNetInpaint

User

Generate 200 pairs of sentences that contains the same meaning, but one with positive sentiment and one with negative sentiment. For both positive sentiment and negative sentiment, you need to write two sentences with the first part being a hint of the sentiment and the second part being the actual content. The first part for both sentences should be same. write in the format like:
hint. positive.
hint. negative.
Make sure that there are two lines for each pairs. Also, the hint should provide enough context and both positive and negative sentiment should be related to the hint. Do not repeat the hint, also make sure that there is only two sentences in each of the line, one is the hint and the other is about the sentiment.

ChatGPT

1. Thanks to her efforts. The event was a huge success.

   Despite her efforts. The event was a complete disaster.

2. ...

The sentences is then added with a prefix to indicates the sentiment of the context. Here we demonstrates a subset of our generated dataset:

Dataset

1. Positive Sentiment: Thanks to her efforts. The event was a huge success.

   Negative Sentiment: Despite her efforts. The event was a complete disaster.

2. Positive Sentiment: This book is definitely interesting. I can't put it down; it's full of surprises.

   Negative Sentiment: This book is definitely interesting. I can't wait to finish it; it's so predictable.

3. Positive Sentiment: Regarding the lecture. It was insightful and engaging.

   Negative Sentiment: Regarding the lecture. It was dull and confusing.

4. Positive Sentiment: Despite the initial problems. The project was a success.

   Negative Sentiment: Despite the initial problems. The project ended in failure.

5. Positive Sentiment: Regarding the new app. It's user-friendly and very helpful.

   Negative Sentiment: Regarding the new app. It's complicated and not useful.

6. Positive Sentiment: Reflecting on my environmental initiatives. Implementing changes has reduced my carbon footprint.

   Negative Sentiment: Reflecting on my environmental initiatives. It's challenging to maintain, and progress is slow.

7. Positive Sentiment: The business proposal was well-received. The ideas were innovative, and the presentation was convincing.

   Negative Sentiment: The business proposal was rejected. The ideas were impractical, and the presentation was unconvincing.

8. Positive Sentiment: The training program was highly effective. It boosted skills and confidence, and everyone left motivated.

   Negative Sentiment: The training program was ineffective. It didn't teach much, and most people left feeling unmotivated.

9. ...

G. Additional Results on Sentiment Editing

Evaluation. Below, we demonstrate the prompt used for evaluating the editing results:

Negative Prompt	Our Edited Results
Negative Sentiment: This book is definitely interesting. I can't wait to finish it; it's so predictable. It's cramped and lacks proper facilities.	Positive Sentiment: This book is definitely interesting. I can't wait to see it; it sounds so beautiful. It's spacious and has great facilities.
Negative Sentiment: Despite her efforts. The event was a complete disaster.	Positive Sentiment: Thanks to her efforts. This event was a fantastic comedy game.
Negative Sentiment: Regarding the lecture. It was dull and confusing.	Positive Sentiment: Regarding the lecture. It was clear and surprising.
Negative Sentiment: Despite the initial problems. The project ended in failure.	Positive Sentiment: Despite the initial problems. New project still in progress.
Negative Sentiment: Regarding the new app. It's complicated and not useful.	Positive Sentiment: Regarding the new app. It's On and It's Epic.
Negative Sentiment: Reflecting on my environmental initiatives. It's challenging to maintain, and progress is slow.	Positive Sentiment: Reflecting on my environmental initiatives. It's easy to understand, and progress is undeniable.

Table 7. **Editing results of our method with RoBERTa.** The sentences in black are the prompts used for inversion and editing in their respective column. The sentence in red is the one being inverted, and the blue sentence represents the editing result.

User

Given three sentences, confirm that the second sentence is roughly the same sentence structure as the first sentence, then confirm that the second sentence has positive sentiment. Output only two numbers with each number indicating whether the corresponding criteria is satisfied. Use 1 for satisfied and 0 for not satisfied. The sentences are given below:
The event was a complete disaster.
This event was a fantastic comedy game.

ChatGPT

1 1

Comparison between masked inpainting and DICE. In Figure 9 we demonstrates the reconstruction and editing results with our DICE and Masked Inpainting.

H. Limitations

While Discrete Inversion shows promise, we empirically find that editing with multinomial diffusion models may not work as robustly as with masked generative models. Furthermore, it may appear less effective in style transfer tasks, such as transforming an image of a cat into a silver cat statue. Interesting future directions include: (1) developing a more theoretical analysis of mutual information and convergence for continuous and discrete inversion algorithms, (2) extending Discrete Inversion to score distillation sampling, and (3) exploring the integration of Semantic Guidance within discrete settings.

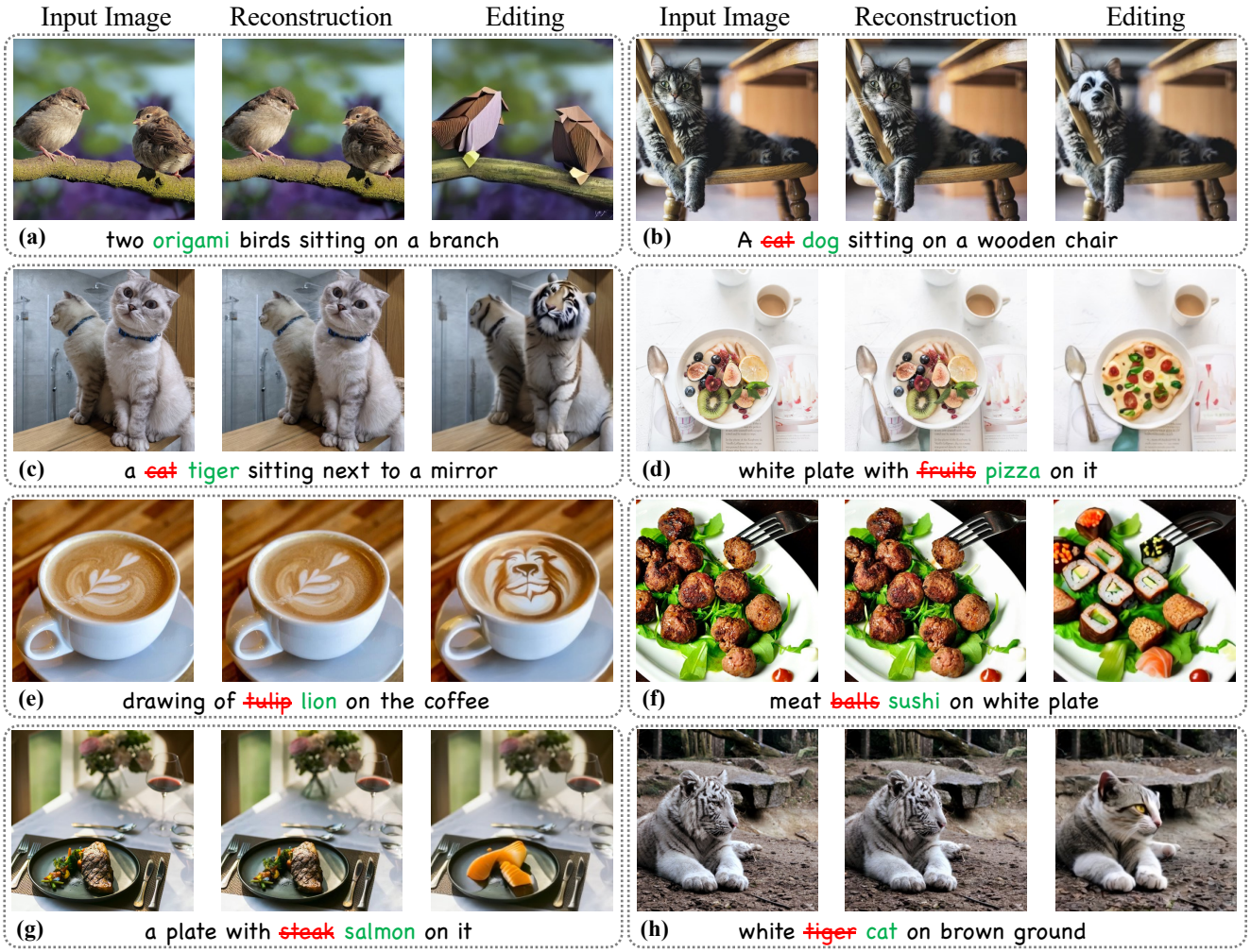

Figure 8. **Reconstruction and editing result with DICE+Paella.**

Method		Structure	CLIP Similarity	
Inversion+Model	Editing	Distance$_{\times 10^3}$ ↓	Whole ↑	Edited ↑
ControlNet-InPaint (scale=0.5) + SD1.5	Prompt	65.12	25.50	22.85
ControlNet-InPaint (scale=1.0) + SD1.5	Prompt	60.87	24.35	21.40
SDEdit ($t_0 = 0.4$) + Paella	Prompt	30.52	23.14	20.72
SDEdit ($t_0 = 0.6$) + Paella	Prompt	38.62	23.22	20.86
Inpainting + Paella	Prompt	91.10	25.36	23.42
Ours + Paella	Prompt	11.34	23.79	21.23

Table 8. **Additional baselines.** We compare with SDEdit [36] and ControlNet [61].

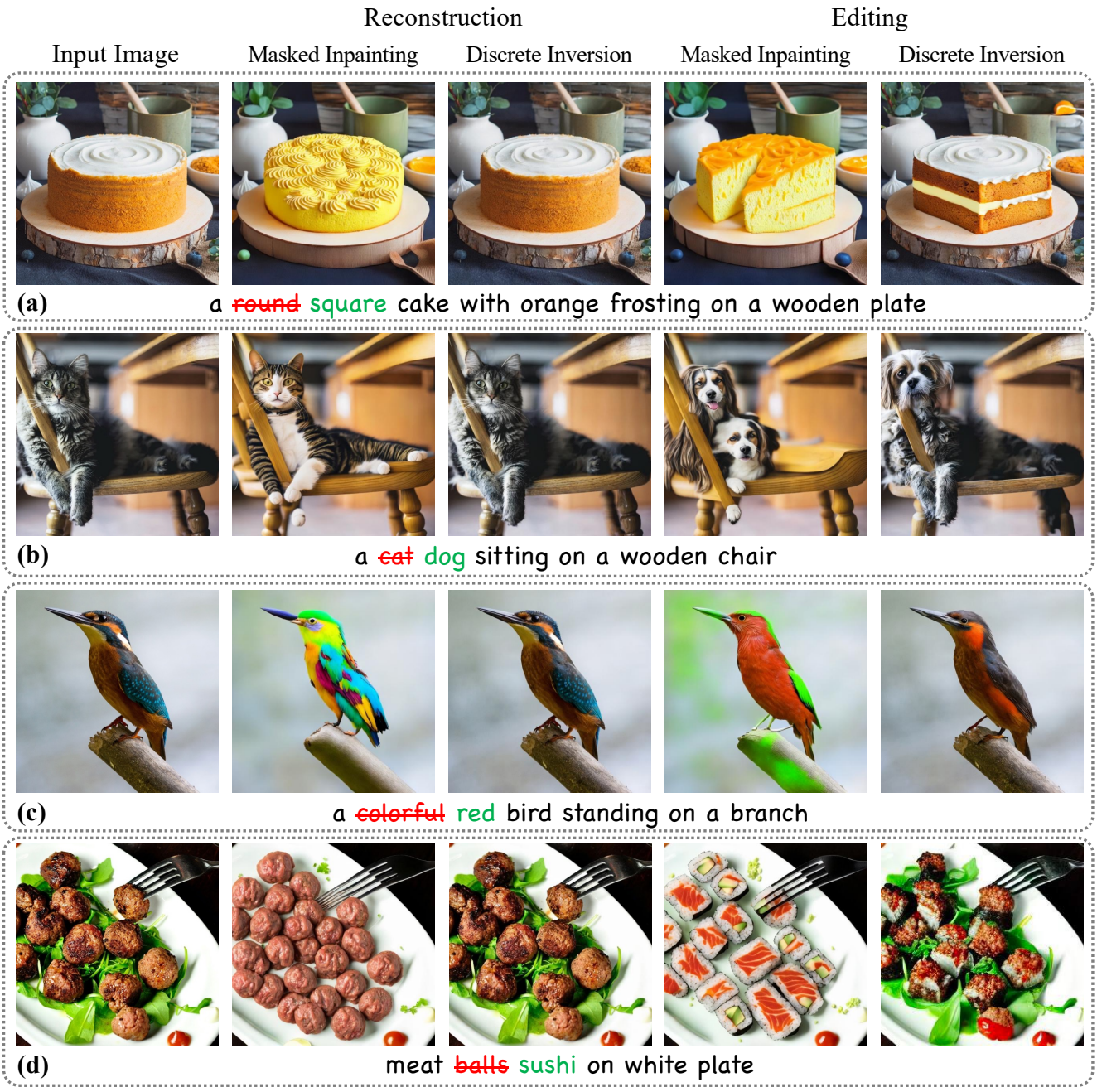

Figure 9. **Reconstruction and editing result with DICE and masked inpainting.** Notice that for reconstruction, we use the red prompt, but for editing we use the green prompt.

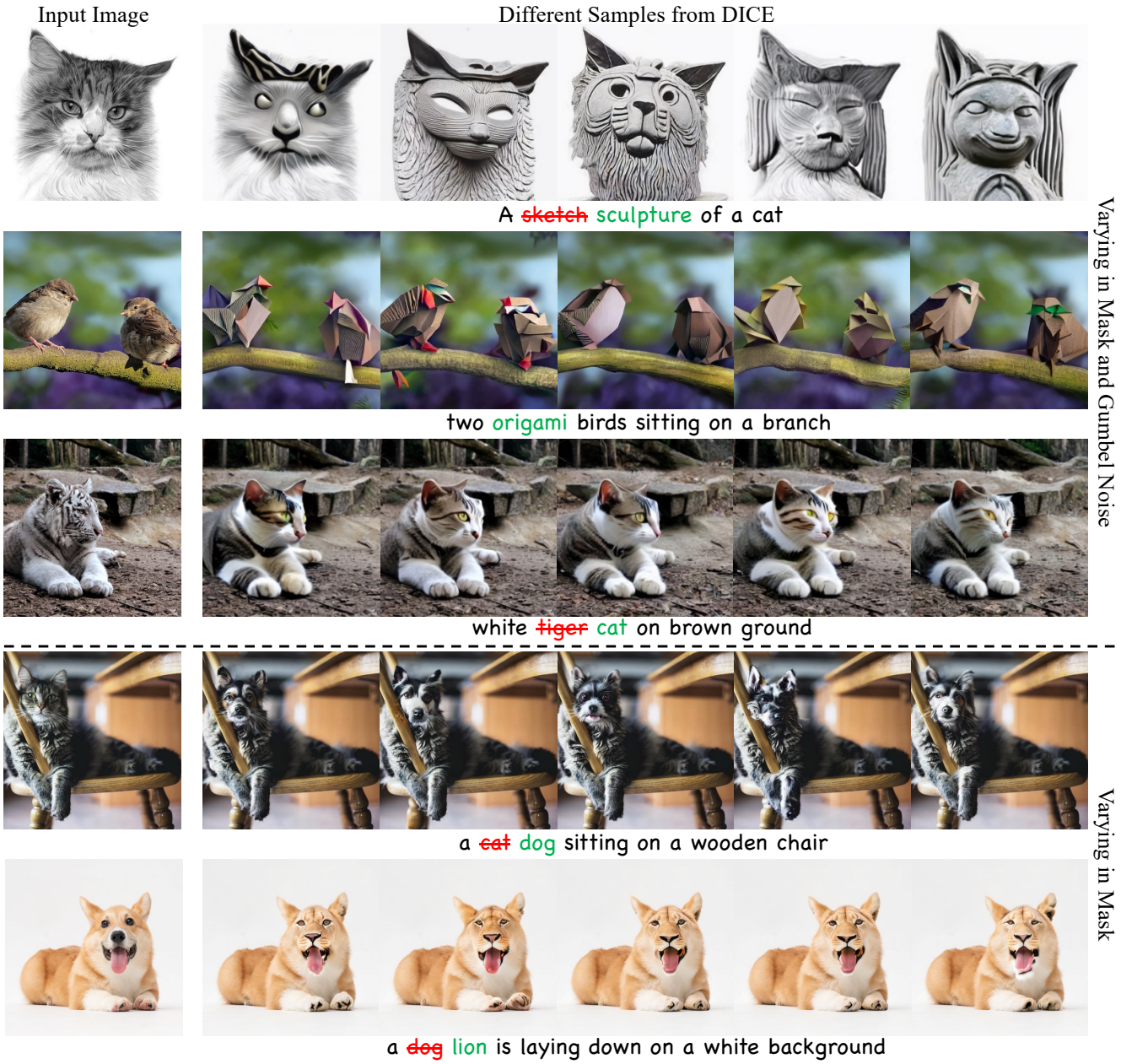

Figure 10. **Image Editing with Diversity.** Due to the stochastic nature of our method, we can generate diverse outputs. The first three rows illustrate variations in both inversion masks and injected Gumbel noise ($\lambda_1 = 0.7$, $\lambda_2 = 0.3$). The last two rows demonstrate variations using only inversion masks ($\lambda_1 = 1$, $\lambda_2 = 0$).

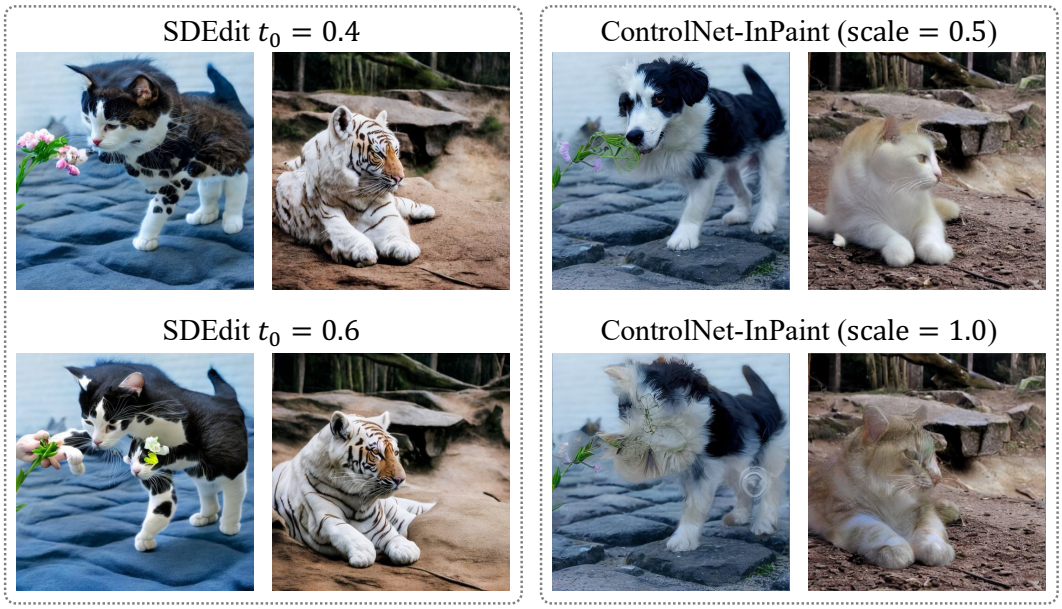

Figure 11. **Editing results with SDEdit and ControlNet.** For SDEdit we show examples of $t_0 = 0.4, 0.6$. For ControlNet we show examples of conditioning scale of 0.5 and 1.

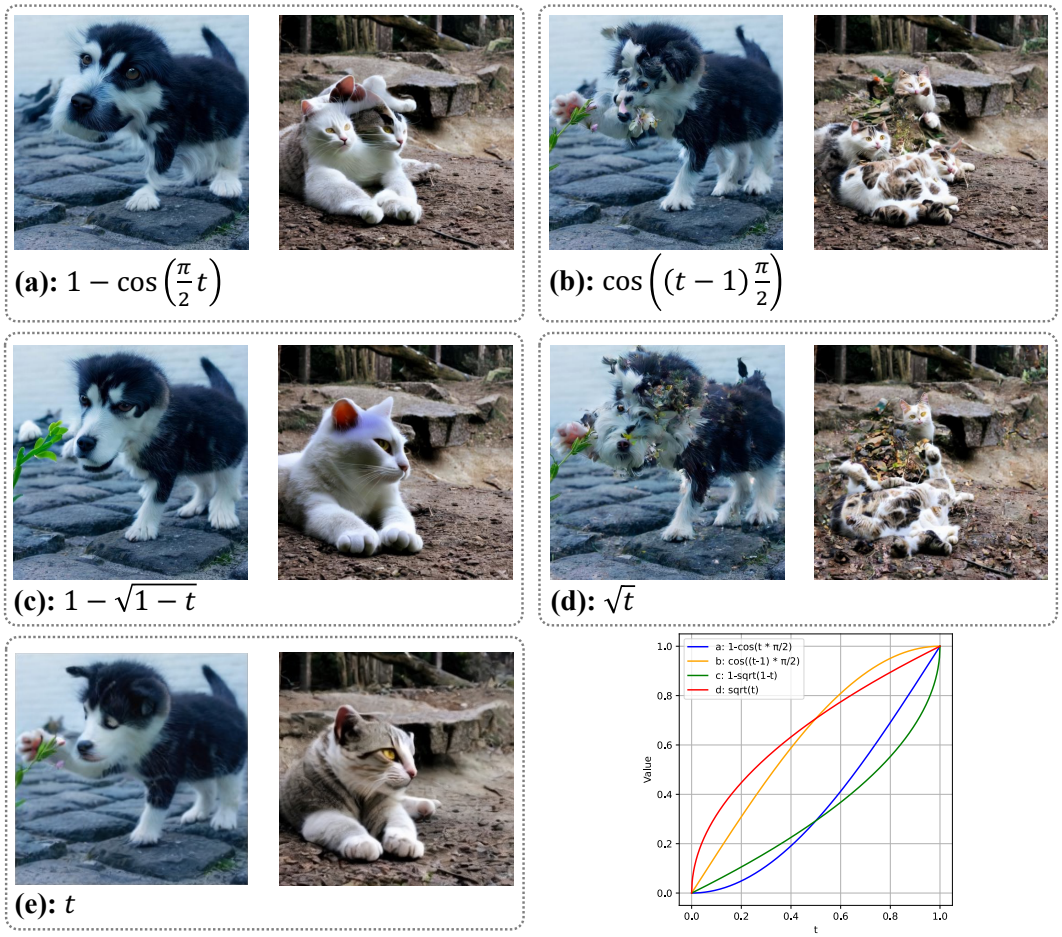

Figure 12. **Comparison with different masking schedule.** (a): $1 - \cos(t \cdot \pi/2)$, (b): $\cos((t-1) \cdot \pi/2)$,(c): $1 - \sqrt{1-t}$, (d): $\sqrt{t}$.

	Structure	CLIP Similarity	
Mask Schedule	Distance$_{\times 10^3}$ ↓	Whole ↑	Edited ↑
(a): $1 - \cos(t \cdot \pi/2)$	7.54	23.48	20.96
(b): $\cos((t-1) \cdot \pi/2)$	25.39	23.56	21.24
(c): $1 - \sqrt{1-t}$	5.11	22.99	20.50
(d): $\sqrt{t}$	26.35	23.59	21.36
(e): t	11.34	23.79	21.23

Table 9. **Comparison with different masking schedule.** (a): $1 - \cos(t \cdot \pi/2)$, (b): $\cos((t-1) \cdot \pi/2)$,(c): $1 - \sqrt{1-t}$, (d): $\sqrt{t}$.

Figure 13. **Comparison with different noise injection functions.**

Figure 14. **Inversion reconstruction comparison with different lambda schedule.**

Figure 15. **Comparison between inclusive and random masks.**

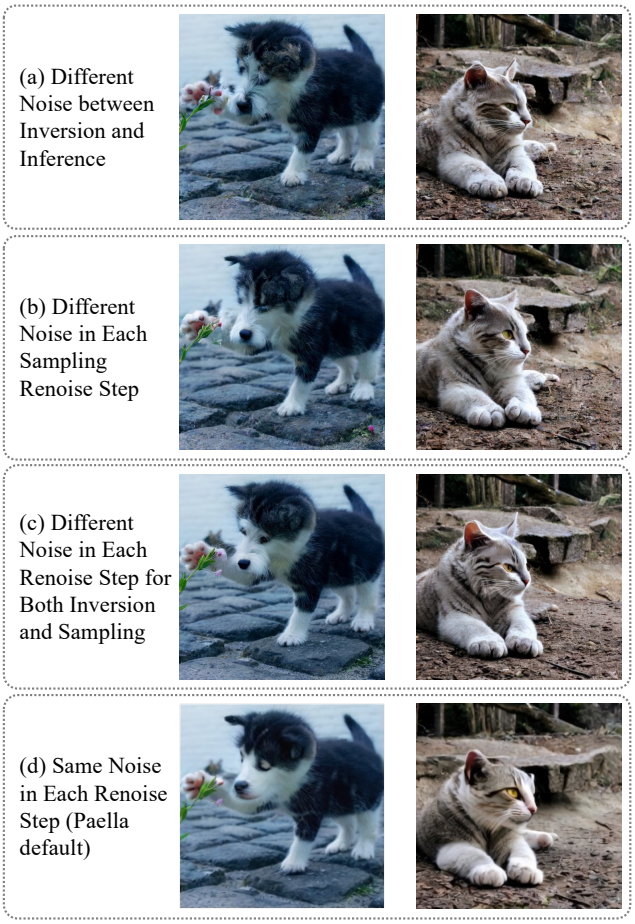

Figure 16. **Comparison with different noise token schedule.** Here we show visualization results of using different noise tokens in inversion and inference, using different noise tokens in each renoising step of the sampling process, using different noise tokens in each renoising step of both inversion and sampling process, and ours by using the same tokens in both inversion and inference.