A. UFOGen: You Forward Once Large Scale Text-to-Image Generation via Diffusion GANs

A.1. Deriving the KL objective in Equation 4

In this section, we provide a derivation of obtaining a reconstruction objective from the KL term in Equation 4:

$$\mathrm{KL}(q(x_t|x_{t-1})||p_\theta(x_t|x'_{t-1})). \tag{7}$$

Note that $q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$ is a Gaussian distribution defined by the forward diffusion. For $p_\theta(x_t|x'_{t-1})$, although the distribution on $p_\theta(x'_{t-1})$ is quite complicated (because this depends on the generator model), given a specific x'_{t-1}, it follows the same distribution of forward diffusion: $p_\theta(x_t|x'_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x'_{t-1}, \beta_t\mathbf{I})$. Therefore, Equation 7 is the KL divergence between two Gaussian distributions, which we can computed in closed form. For two multivariate Gaussian distributions with means μ_1, μ_2 and covariance Σ_1, Σ_2, the KL divergence can be expressed as

$$\frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - d + \operatorname{tr}\left\{\Sigma_2^{-1}\Sigma_1\right\} + (\mu_2-\mu_1)^T\,\Sigma_2^{-1}\,(\mu_2-\mu_1)\right].$$

We can easily plug-in the means and variances for $q(x_t|x_{t-1})$ and $p_\theta(x'_t|x'_{t-1})$ into the expression. Note that $\Sigma_1 = \Sigma_2 = \beta_t\mathbf{I}$, so the expression can be simplified to

$$\frac{(1-\beta_t)||x'_{t-1} - x_{t-1}||^2}{2\beta_t} + C, \tag{8}$$

where C is a constant. Therefore, with the outer expectation over $q(x_t)$ in Equation 4, minimizing the KL objective is equivalent to minimizing a weighted reconstruction loss between x'_{t-1} and x_{t-1}, where x_{t-1} is obtained by sampling $x_0 \sim q(x_0)$ and $x_{t-1} \sim q(x_{t-1}|x_0)$; x'_{t-1} is obtained from generating an x'_0 from the generator followed by sampling $x'_{t-1} \sim q(x'_{t-1}|x'_0)$.

Note that in [65], the authors did not leverage the Gaussian distribution's KL-divergence property. Instead, they decomposed the KL-divergence into an entropy component and a cross-entropy component, subsequently simplifying each aspect by empirically estimating the expectation. This simplification effectively converges to the same objective as expressed in Equation 8, albeit with an appended term associated with entropy. The authors of [65] introduced an auxiliary parametric distribution for entropy estimation, which led to an adversarial training objective. Nevertheless, our analysis suggests that this additional term is dispensable, and we have not encountered any practical challenges when omitting it.

A.2. Analysis of the distribution matching at x_0

In this section, we offer a detailed explanation of why training the model with the objective presented in Equation 4 effectively results in matching x_0 and x'_0. The rationale is intuitive: x_{t-1} and x'_{t-1} are both derived from their respective base images, x_0 and x'_0, through independent Gaussian noise corruptions. As a result, when we enforce the alignment of distributions between x_{t-1} and x'_{t-1}, this implicitly encourages a matching of the distributions between x_0 and x'_0 as well. To provide a more rigorous and formal analysis, we proceed as follows.

A.2.1 Adversarial term

We provide an explanation of why the adversarial objective in Equation 4 corresponds to matching the distributions $q(x_0)$ and $p_\theta(x'_0)$. Firstly, note that since $q(x_{t-1}) = \mathbb{E}_{q(x_0)}\left[q(x_{t-1}|x_0)\right]$, where $q(x_{t-1}|x_0)$ is the Gaussian distribution defined by the forward diffusion process. Therefore, $q(x_{t-1})$ can be expressed as a convolution between $q(x_0)$ and a Gaussian kernel:

$$q(x_{t-1}) = q(x_0) * k(z), \quad k(z) = \mathcal{N}(0, (1-\bar{\alpha}_{t-1})\,\mathbf{I}). \tag{9}$$

Similarly, $p_(\theta)(x_{t-1}) = p_\theta(x_0) * k(z)$, where $p_\theta(x_0)$ is the implicit distribution defined by the generator G_θ.

In the following lemma, we show that for a probability divergence D, if $p(x)$ and $q(x)$ are convoluted with the same kernel $k(z)$, then minimizing D on the distributions after the convolution is equivalent to matching the original distributions $p(x)$ and $q(x)$.

Lemma 1 *Let $Y = X + K$, if K is absolutely continuous with density $k(z) > 0, x \in \mathbb{R}$. And a divergence $\boldsymbol{D}(\mathbb{Q}||\mathbb{P})$ is a measure of the difference between distribution $\mathbb{Q}$ and $\mathbb{P}$, where $\boldsymbol{D}(\mathbb{Q}||\mathbb{P}) \geq 0$ and $\boldsymbol{D}(\mathbb{Q}||\mathbb{P}) = 0 \iff \mathbb{Q} = \mathbb{P}$. Then $\boldsymbol{D}(q(y)||p(y)) = 0 \iff q(x) = p(x)$.*

Proof: The probability density of the summation between two variables is the convolution between their probability densities. Thus, we have:

$$\begin{aligned}
&\boldsymbol{D}(q(y)||p(y)) = \boldsymbol{D}(q(x) * k(z)||p(x) * k(z)),\\
&\boldsymbol{D}(q(x) * k(z)||p(x) * k(z)) = 0 \;\; a.e.,\\
&\quad \iff q(x) * k(z) = p(x) * k(z),\\
&\quad \iff \mathcal{F}(q(x) * k(z)) = \mathcal{F}(p(x) * k(z)),\\
&\quad \iff \mathcal{F}(q(x))\mathcal{F}(k(z)) = \mathcal{F}(p(x))\mathcal{F}(k(z)),\\
&\quad \iff q(x) = p(x) \quad a.e.,
\end{aligned}$$

where $\mathcal{F}$ denotes the Fourier Transform, and we utilize the invertibility of the Fourier Transform for the above derivation.

Thus, from **Lemma** 1, we can get $q(x_0) = p_\theta(x_0)$ almost everywhere when $\text{JSD}(q(x_{t-1})||p_\theta(x_{t-1})) = 0$. Notably, while training with the adversarial objective on x_{t-1} inherently aligns the distributions of $q(x_0)$ and $p_{theta}(x'_0)$, it is crucial to acknowledge that we cannot directly employ GAN training on x_0. This is because the additive Gaussian noise, which serves to smooth the distributions, rendering GAN training more stable. Indeed, training GANs on smooth distributions is one of the essential components of all diffusion-GAN hybrid models, as highlighted in [62].

A.2.2 KL term

Here we show that minimizing the reconstruction loss in Equation 8 over the expectation of $q(x_t)$ as in Equation 4 is equivalent to minimizing the reconstruction loss between x_0 and x'_0. According to the sampling scheme of x_{t-1} and x'_{t-1}, we have

$$\mathbb{E}_{q(x_{t-1}),p_\theta(x'_{t-1})}\left[\frac{(1-\beta_t)||x'_{t-1} - x_{t-1}||^2}{2\beta_t}\right] = \mathbb{E}_{q(x_0)q(x_{t-1}|x_0),p_\theta(x'_0)p_\theta(x'_{t-1}|x'_0)}\left[\frac{(1-\beta_t)||x'_{t-1} - x_{t-1}||^2}{2\beta_t}\right]. \tag{10}$$

Since the forward diffusion $q(x_{t-1}|x_0)$ has the Gaussian form [18]

$$q(x_{t-1}|x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\,\mathbf{I}\right) \tag{11}$$

and similar form holds for $p_\theta(x'_{t-1}|x'_0)$, we can rewrite the expectation in Equation 10 over the distribution of simple Gaussian distribution $p(\epsilon) = \mathcal{N}(\epsilon; 0, \mathbf{I})$:

$$\mathbb{E}_{q(x_0)q(x_{t-1}|x_0),p_\theta(x'_0)p_\theta(x'_{t-1}|x'_0)}\left[\frac{(1-\beta_t)||x'_{t-1} - x_{t-1}||^2}{2\beta_t}\right] = \mathbb{E}_{q(x_0),p_\theta(x'_0),p(\epsilon)}\left[\frac{(1-\beta_t)||x'_{t-1} - x_{t-1}||^2}{2\beta_t}\right], \tag{12}$$

where $x'_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}'_0 + (1-\bar{\alpha}_{t-1})\,\epsilon'$ and $x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + (1-\bar{\alpha}_{t-1})\,\epsilon$ are obtained by i.i.d. samples ϵ', ϵ from $p(\epsilon)$. Plug in the expressions to Equation 12, we obtain

$$\begin{aligned}
&\mathbb{E}_{q(x_0),p_\theta(x'_0),p(\epsilon)}\left[\frac{(1-\beta_t)||x'_{t-1} - x_{t-1}||^2}{2\beta_t}\right]\\
&= \mathbb{E}_{q(x_0),p_\theta(x'_0),p(\epsilon)}\left[\frac{(1-\beta_t)||\sqrt{\bar{\alpha}_{t-1}}(x'_0 - x_0) + (1-\bar{\alpha}_{t-1})\,(\epsilon' - \epsilon)||^2}{2\beta_t}\right]\\
&= \mathbb{E}_{q(x_0),p_\theta(x'_0)}\left[\frac{(1-\beta_t)\bar{\alpha}_{t-1}||x'_0 - x_0||^2}{2\beta_t}\right] + C,
\end{aligned}$$

where C is a constant independent of the model. Therefore, we claim the equivalence of the reconstruction objective and the matching between x_0 and x'_0.

However, it's essential to emphasize that the matching between x_0 and x'_0 is performed with an expectation over Gaussian noises. In practical terms, this approach can introduce significant variance during the sampling of x_{t-1} and x'_{t-1}. This variance, in turn, may result in a less robust learning signal when it comes to aligning the distributions at clean data. As detailed in Section 4.1, we propose a refinement to address this issue. Specifically, we advocate for the direct enforcement of reconstruction between x_0 and x'_0. This modification introduces explicit distribution matching at the level of clean data, enhancing the model's robustness and effectiveness.

A.3. Experimental Details

For all the experiments, we initialize the parameters of both the generator and discriminator with the pre-trained Stable Diffusion (SD) 1.5 checkpoint. In consequence, we follow SD 1.5 to use the same VAE for image encoding/decoding and the frozen text encoder of CLIP ViT-L/14 for text conditioning. Note that both the generator and discriminator operates on latent space. In other words, the generator generates the latent variables and the discriminator distinguishes the fake and true (noisy) latent variables.

Important Hyper-parameters One important hyper-parameter is the *denoising step size* during training, which is the gap between $t-1$ and t. Note that in Section 4.1, we mentioned that the model is trained with multiple denoising steps, while it enables one-step inference. Throughout the experiments, we train the models using denoising step size 250, given the 1000-step discrete time scheduler of SD. Specifically, during training, we sample t randomly from 1 to 1000, and the time step for $t-1$ is $max(0, t-250)$. We conduct ablation studies on this hyper-parameter in Section A.4.

Another important hyper-parameter is λ_{KL}, the weighting coefficient for reconstruction term in the objective in Equation 6. We set $\lambda_{KL} = 1.0$ throughout the experiments. We found the results insensitive to slight variations of this coefficient.

Common Hyper-parameters We train our models on the LAION Aesthetic 6+ dataset. For the generator, we use AdamW optimizer [36] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$; for the discriminator, we use AdamW optimizer with $\beta_1 = 0.0$ and $\beta_2 = 0.999$. We adopt learning rate warm-up in the first 1000 steps, with peak learning rate $1e-4$ for both the discriminator and the generator. For training the generator, we apply gradient norm clipping with value 1.0 for generator only. We use batch size 1024. For the generator, we apply EMA with coefficient 0.999. We observe quick convergence, typically in $<$ 50k steps.

A.4. Additional Results of Ablation Studies

In this section, we provide additional results for ablation studies, which are briefly covered in the main text due to the constraints of space. In Appendix A.4.1, we provide qualitative results corresponds to the ablation study conducted in Section 5.2. In Appendix A.4.2, we conduct an additional ablation experiment on the denoising step size during training.

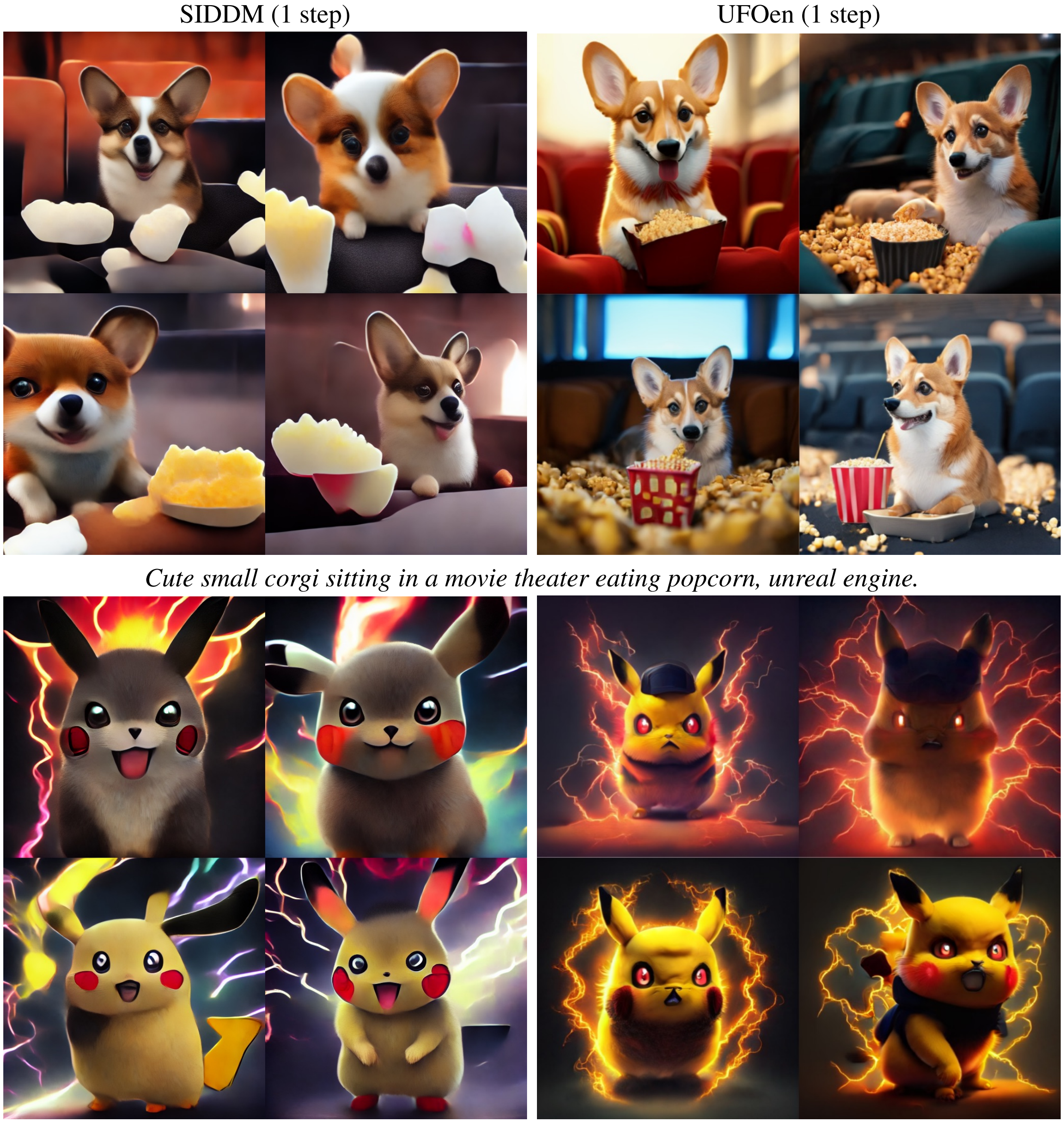
A.4.1 Qualitative Results for Table 4

We provide qualitative examples to contrast between the single-step sample generated by SIDDM [65] and our proposed UFOGen. Results are shown in Table 7 and 8. We observe that when sampling from SIDDM in only one-step, the samples are blurry and over-smoothed, while UFOGen can produce sharp samples in single step. The observation strongly supports the effectiveness of our introduced modifications to the training objective.

A.4.2 Ablation on Denoising Step-size

One important hyper-parameter of training UFOGen is the denoising step size, which is the gap between t and $t-1$ during training. Note that although UFOGen can produce samples in one step, the training requires a meaningful denoising step size to compute the adversarial loss on noisy observations. Our model is based on Stable Diffusion, which adopts a discrete time scheduler with 1000 steps. Previous diffusion GAN hybrid models [62, 65] divides the denoising process into 2 to 4 steps. We explore denoising step size 125, 250, 500 and 1000, which corresponds to divide the denoising process to 8, 4, 2, and 1 steps. Note that during training, we sample t uniformly in $[1, 1000)$, and when the sampled t is smaller than the denoising step size, we set $t-1$ to be 0. In other words, a denoising step size 1000 corresponds to always setting $t-1=0$ and hence the adversarial loss is computed on clean data x_0.

Quantitative results of the ablation study is presented in Table 9. We observe that a denoising step size 1000 fails, suggesting that training with the adversarial loss on noisy data is critical for stabilizing the diffusion-GAN training. This observation was made on earlier work [62, 65] as well. We also observe that denoising step size 250 is the sweet spot, which is also aligned with the empirical observations of [62, 65]. We conjecture that the reason for the performance degrade when reducing the denoising step size is that the discriminator does not have enough capacity to discriminate on many distinct noise levels.



Cute small corgi sitting in a movie theater eating popcorn, unreal engine.

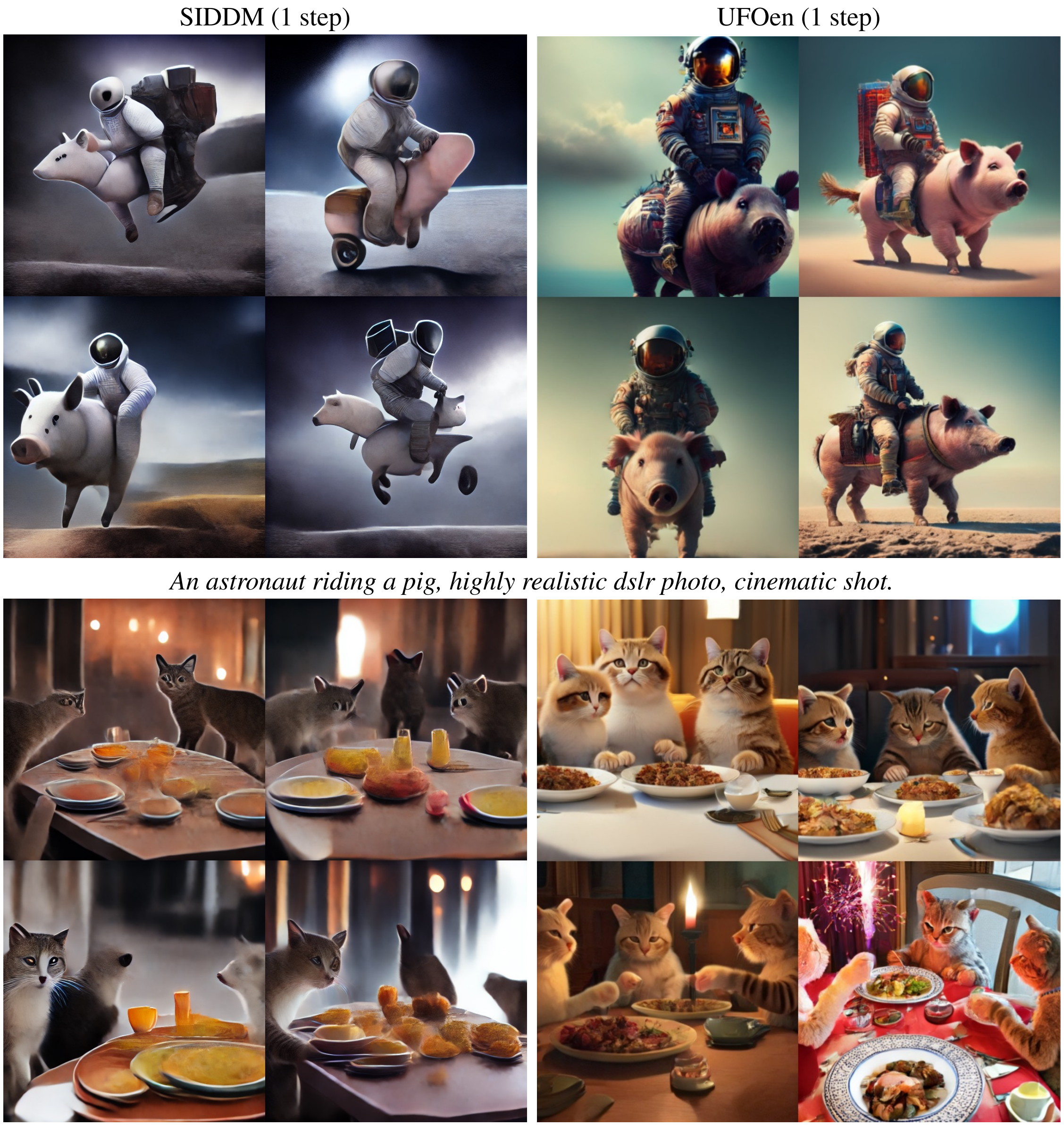
A Pikachu with an angry expression and red eyes, with lightning around it, hyper realistic style.

Table 7. Qualitative results for the ablation study that compares one-step samples from SIDDM and UFOGen.

A.5. Additional Results for Qualitative Comparisons

A.5.1 Failure of Single-step LCM

Consistency models try to learn the consistency mapping that maps every point on the PF-ODE trajectory to its boundary value, *i.e.*, x_0 [59], and therefore ideally consistency models should generate samples in one single step. However, in practice, due to the complexity of the ODE trajectory, one-step generation for consistency models is not feasible, and some iterative refinements are necessary. Notably, Latent consistency models (LCM) [39] distilled the Stable Diffusion model into a consistency model, and we observe that single-step sampling fail to generate reasonable textures. We demonstrate the single-step samples from LCM in figure A.5.1. Due to LCM's ineffectiveness of single-step sampling, we only qualitatively compare our model to 2-step and 4-step LCM.



An astronaut riding a pig, highly realistic dslr photo, cinematic shot.

Three cats having dinner at a table at new years eve, cinematic shot, 8k.

Table 8. Qualitative results for the ablation study that compares one-step samples from SIDDM and UFOGen.

Denoising Step-size	FID-5k	CLIP
1000	32.92	0.288
500	23.2	0.314
250	22.5	0.311
125	24.7	0.305

Table 9. Ablation study comparing the denoising step size during training. For all training denoising step sizes, we generate the samples in one step.

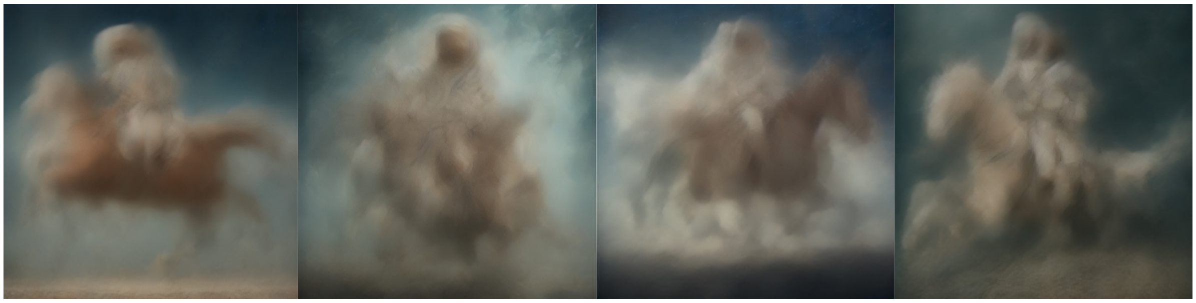


Figure 4. Single-step samples from LCM [39] with prompt "Photo of an astronaut riding a horse".

A.5.2 Extended Results of Table 2

In consideration of space constraints in the main text, our initial qualitative comparison of UFOGen with competing methods for few-step generation in Table 2 employs a single image per prompt. It is essential to note that this approach introduces some variability due to the inherent randomness in image generation. To provide a more comprehensive and objective evaluation, we extend our comparison in this section by presenting four images generated by each method for every prompt. This expanded set of prompts includes those featured in Table 2, along with additional prompts. The results of this in-depth comparison are illustrated across Table 10 to 17, consistently highlighting UFOGen's advantageous performance in generating sharp and visually appealing images within an ultra-low number of steps when compared to competing methods.

Concurrent to our paper submission, the authors of LCM [39] released updated LCM models trained with more resources. The models are claimed to be stronger than the initially released LCM model, which is used in our qualitative evaluation. For fairness in the comparison, we obtain some qualitative samples of the updated LCM model that shares the SD 1.5 backbone with us[3], and show them in Table 18 and 19. We observe that while the new LCM model generates better samples than initial LCM model does, our single-step UFOGen is still highly competitive against 4-step LCM and significantly better than 2-step LCM.

A.6. Additional Qualitative Samples from UFOGen

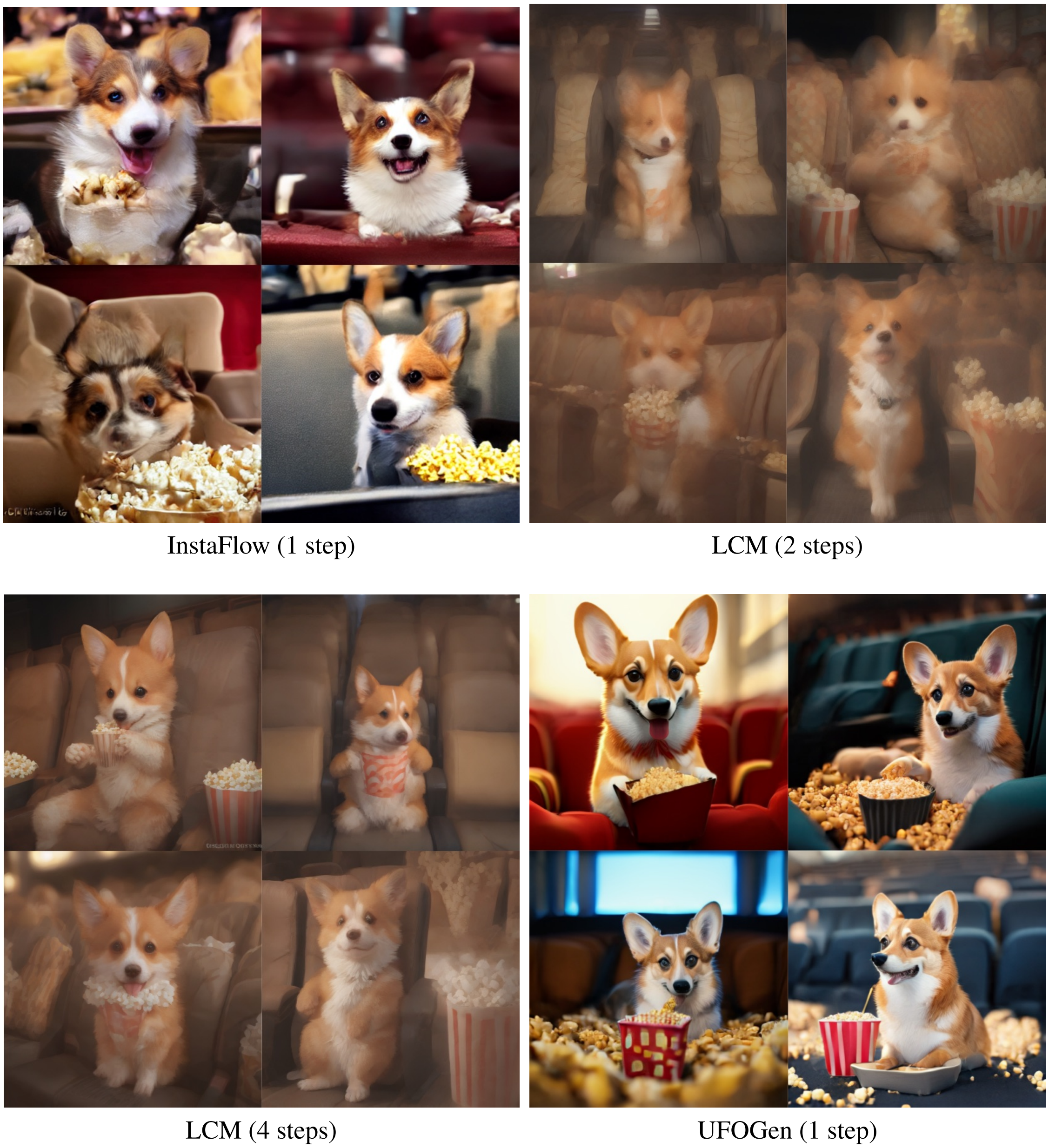
In this section, we present supplementary samples generated by UFOGen models, showcasing the diversity of results in Table 20, 21 and 22. Through an examination of these additional samples, we deduce that UFOGen exhibits the ability to generate high-quality and diverse outputs that align coherently with prompts spanning various styles (such as painting, photo-realistic, anime) and contents (including objects, landscapes, animals, humans, etc.). Notably, our model demonstrates a promising capability to produce visually compelling images with remarkable quality within just a single sampling step.

In Table 23, we present some failure cases of UFOGen. We observe that UFOGen suffers from missing objects, attribute leakage and counting, which are common issues of SD based models, as discussed in [8, 11].

A.7. Additional Results of UFOGen's Applications

In this section, we provide extended results of UFOGen's applications, including the image-to-image generation in Figure 5 and controllable generation in Figure 6.

[3] https://huggingface.co/latent-consistency/lcm-lora-sdv1-5



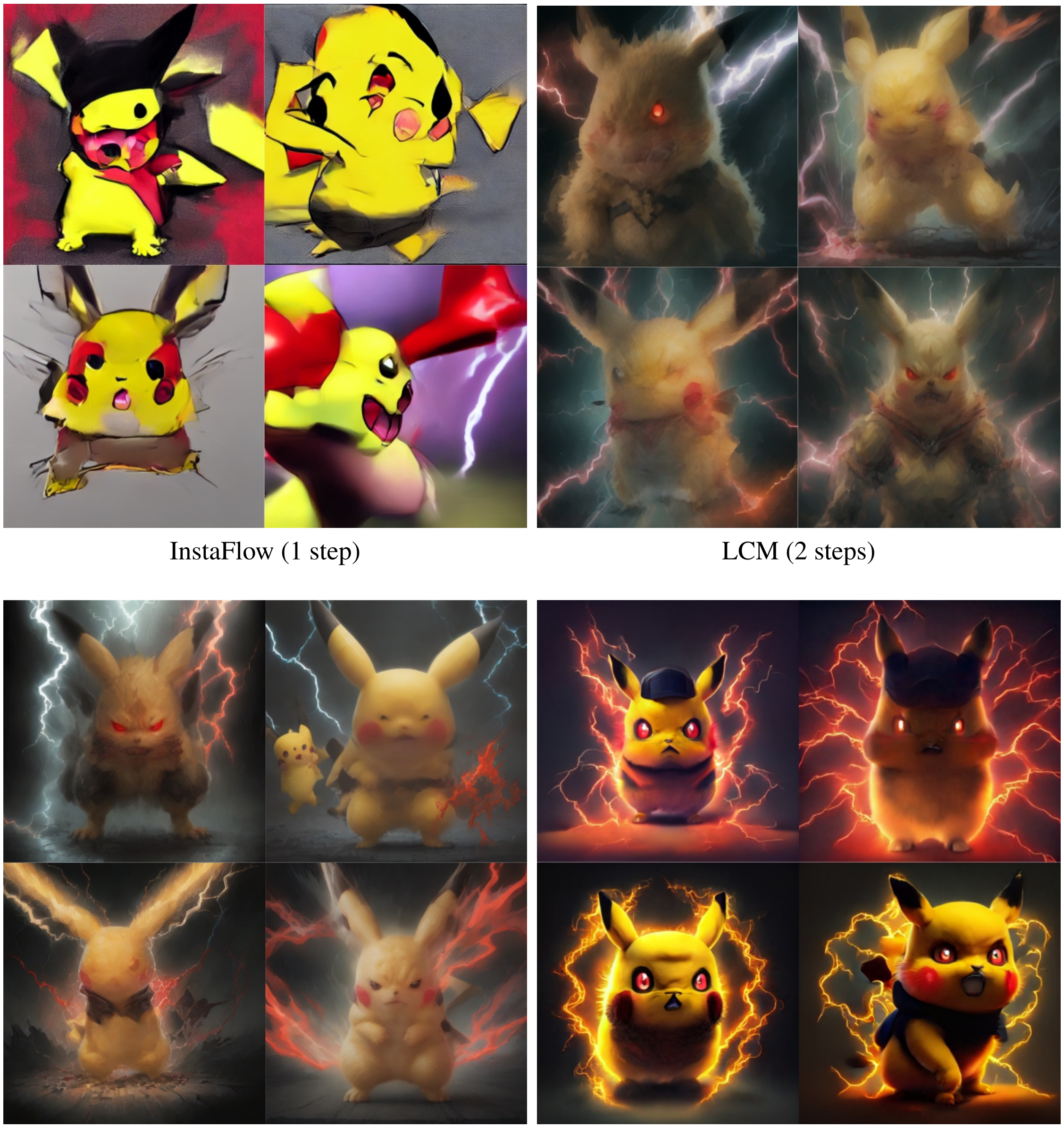
InstaFlow (1 step)

LCM (2 steps)

LCM (4 steps)

UFOGen (1 step)

Table 10. Prompt: *Cute small corgi sitting in a movie theater eating popcorn, unreal engine.*



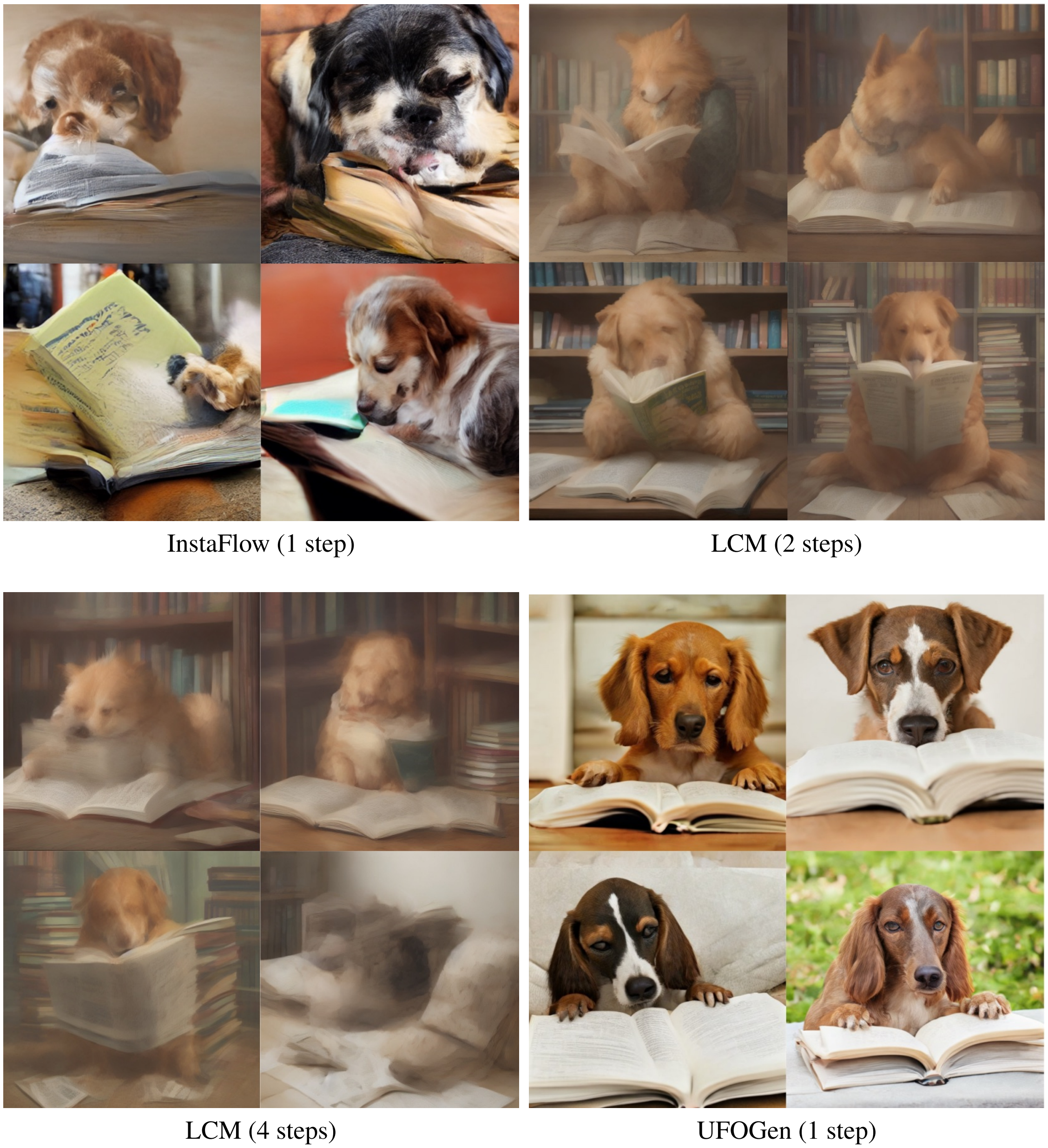
InstaFlow (1 step)

LCM (2 steps)

LCM (4 steps)

UFOGen (1 step)

Table 11. Prompt: *A Pikachu with an angry expression and red eyes, with lightning around it, hyper realistic style.*



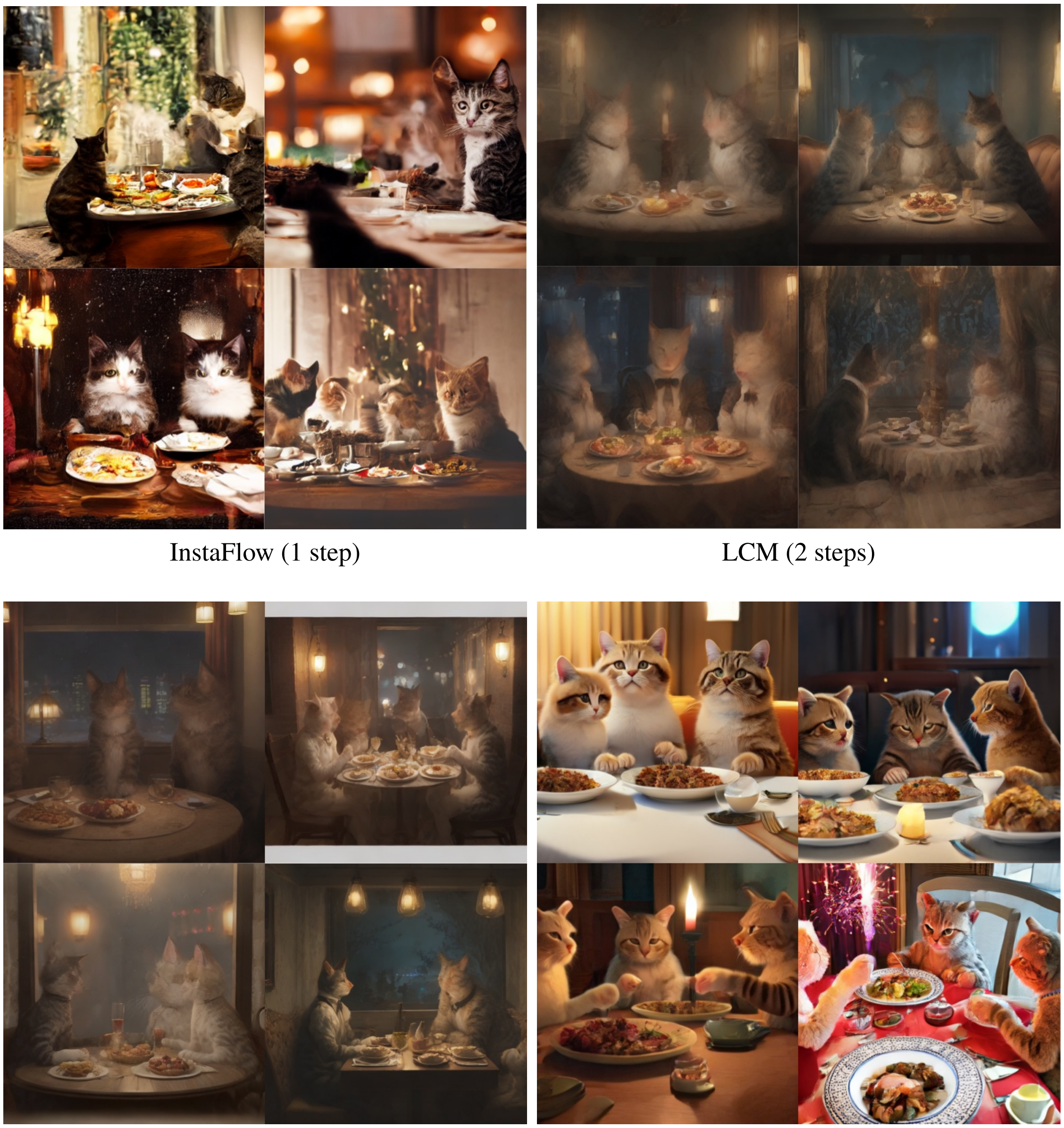
InstaFlow (1 step)

LCM (2 steps)

LCM (4 steps)

UFOGen (1 step)

Table 12. Prompt: *A dog is reading a thick book.*



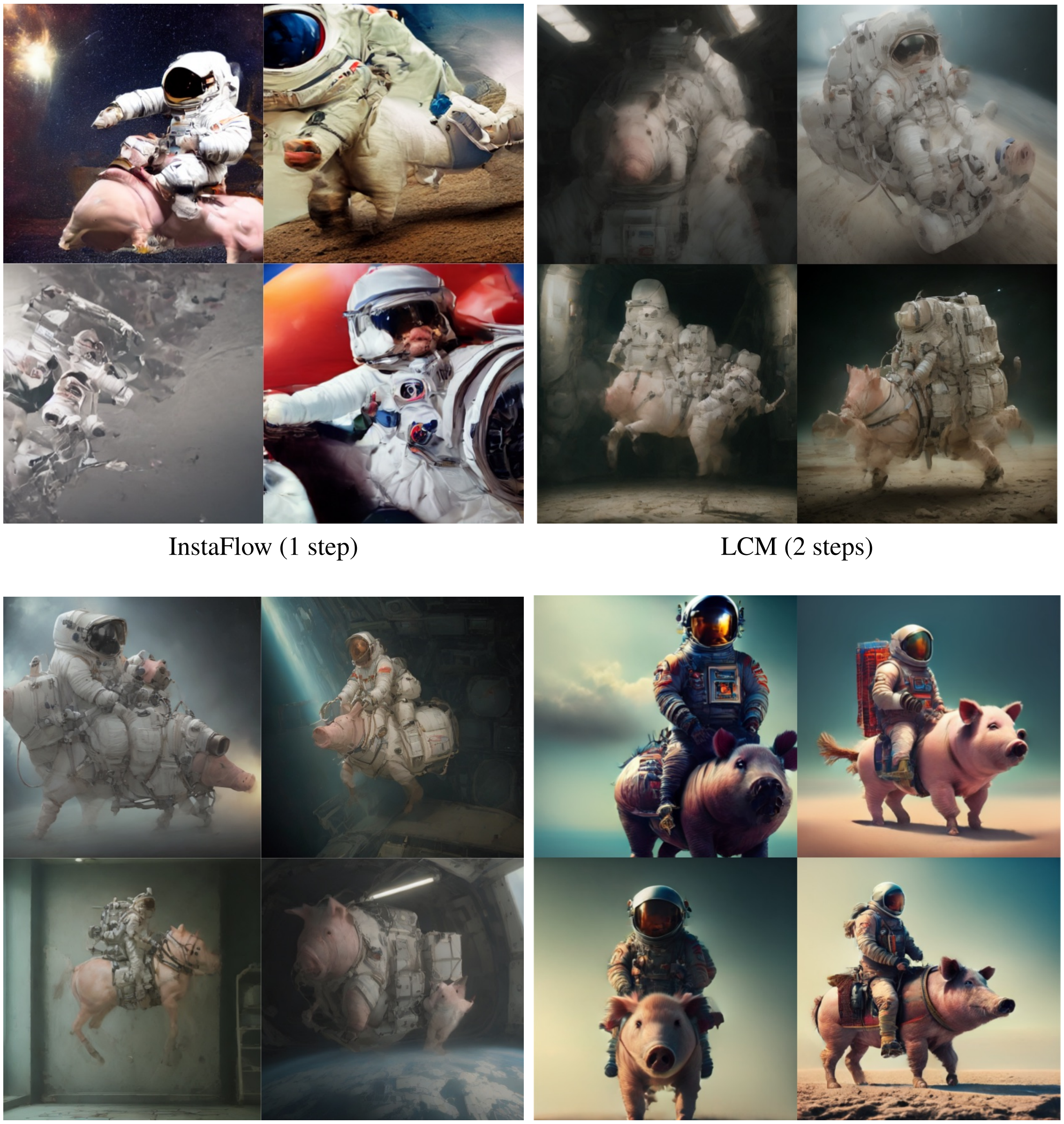
InstaFlow (1 step)

LCM (2 steps)

LCM (4 steps)

UFOGen (1 step)

Table 13. Prompt: *Three cats having dinner at a table at new years eve, cinematic shot, 8k.*



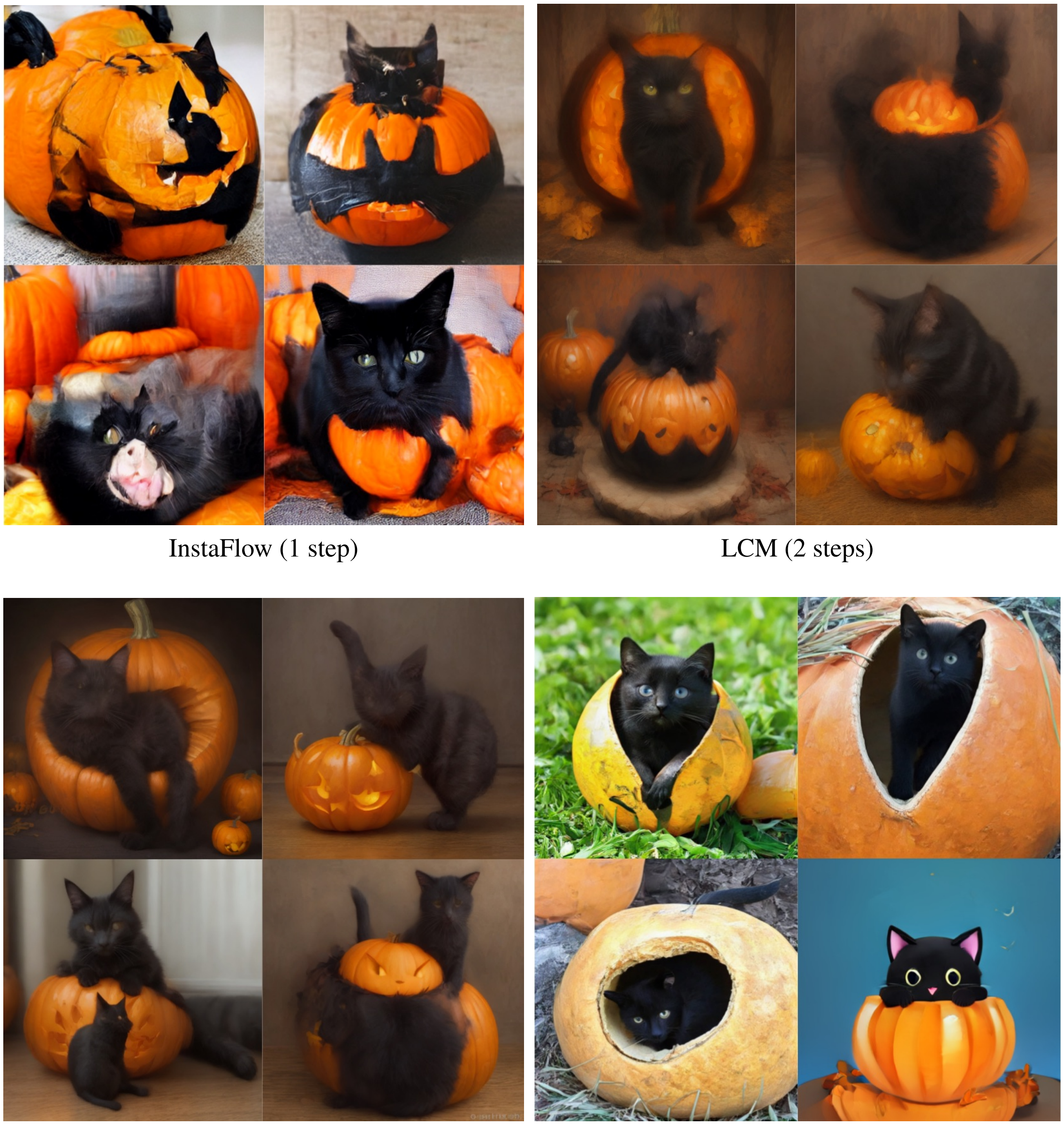
InstaFlow (1 step)

LCM (2 steps)

LCM (4 steps)

UFOGen (1 step)

Table 14. Prompt: *An astronaut riding a pig, highly realistic dslr photo, cinematic shot.*



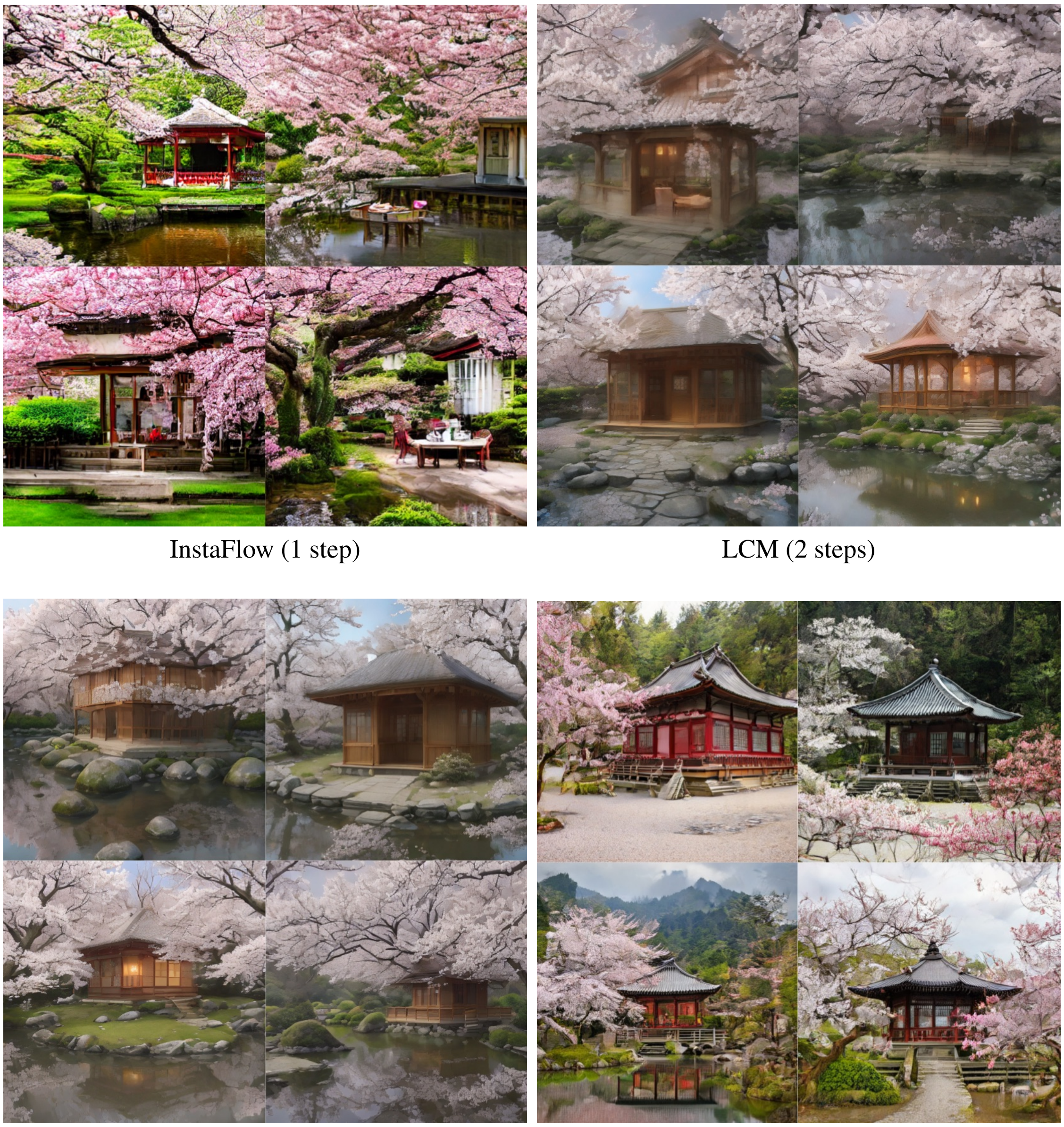
InstaFlow (1 step)

LCM (2 steps)

LCM (4 steps)

UFOGen (1 step)

Table 15. Prompt: *A cute black cat inside of a pumpkin.*



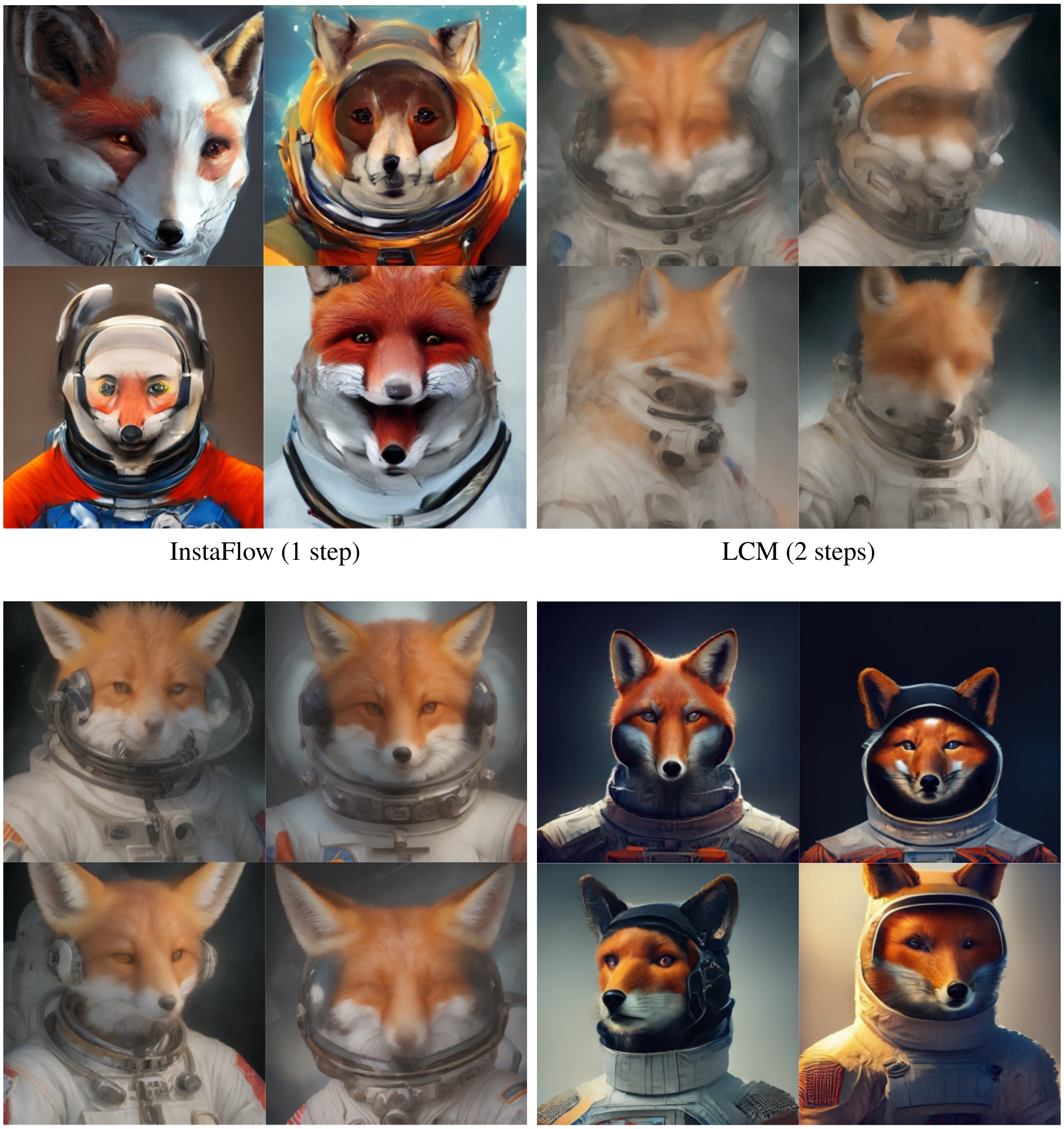
InstaFlow (1 step)

LCM (2 steps)

LCM (4 steps)

UFOGen (1 step)

Table 16. Prompt: *A traditional tea house in a tranquil garden with blooming cherry blossom trees.*



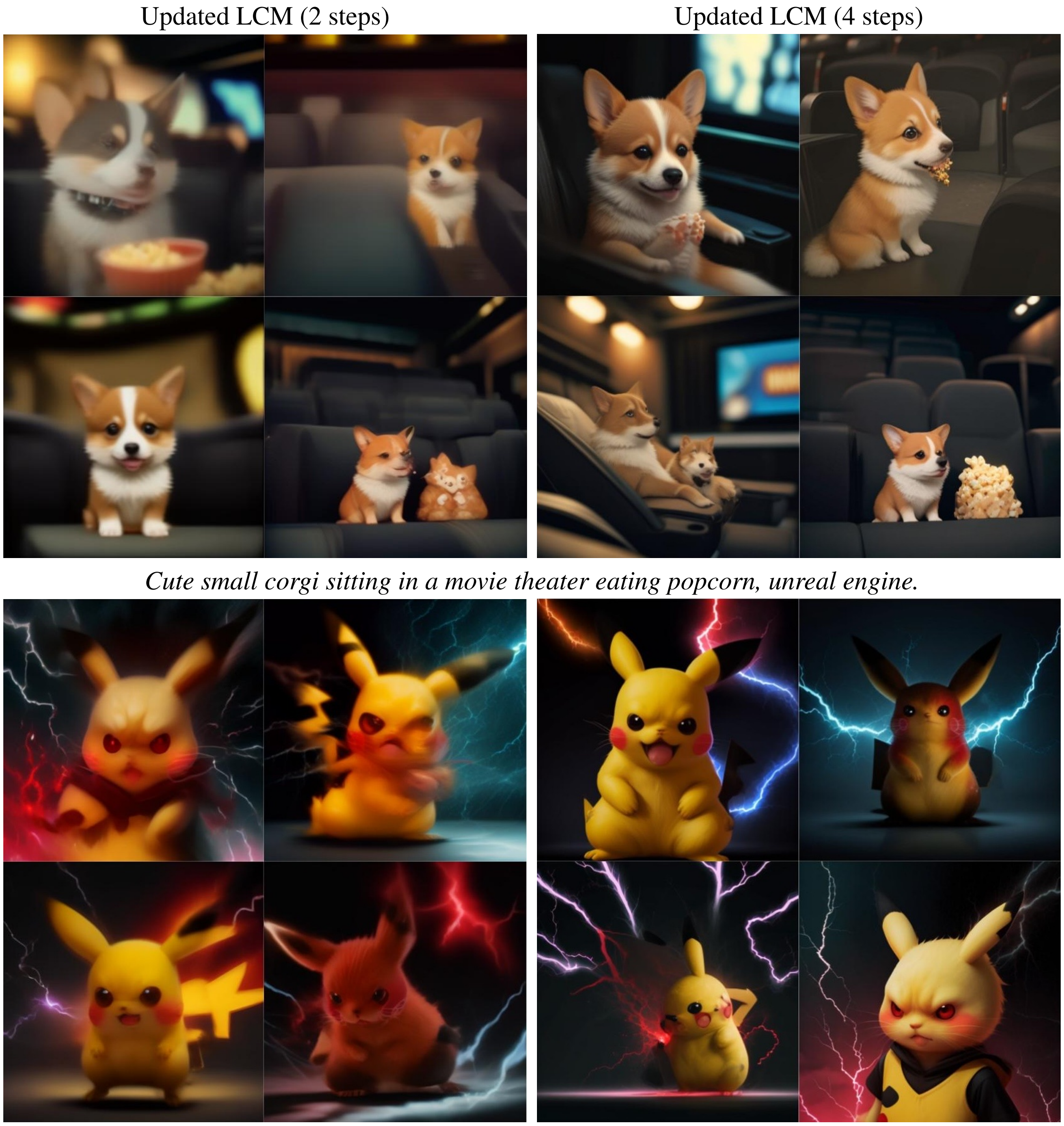
InstaFlow (1 step)

LCM (2 steps)

LCM (4 steps)

UFOGen (1 step)

Table 17. Prompt: *Hyperrealistic photo of a fox astronaut, perfect face, artstation.*



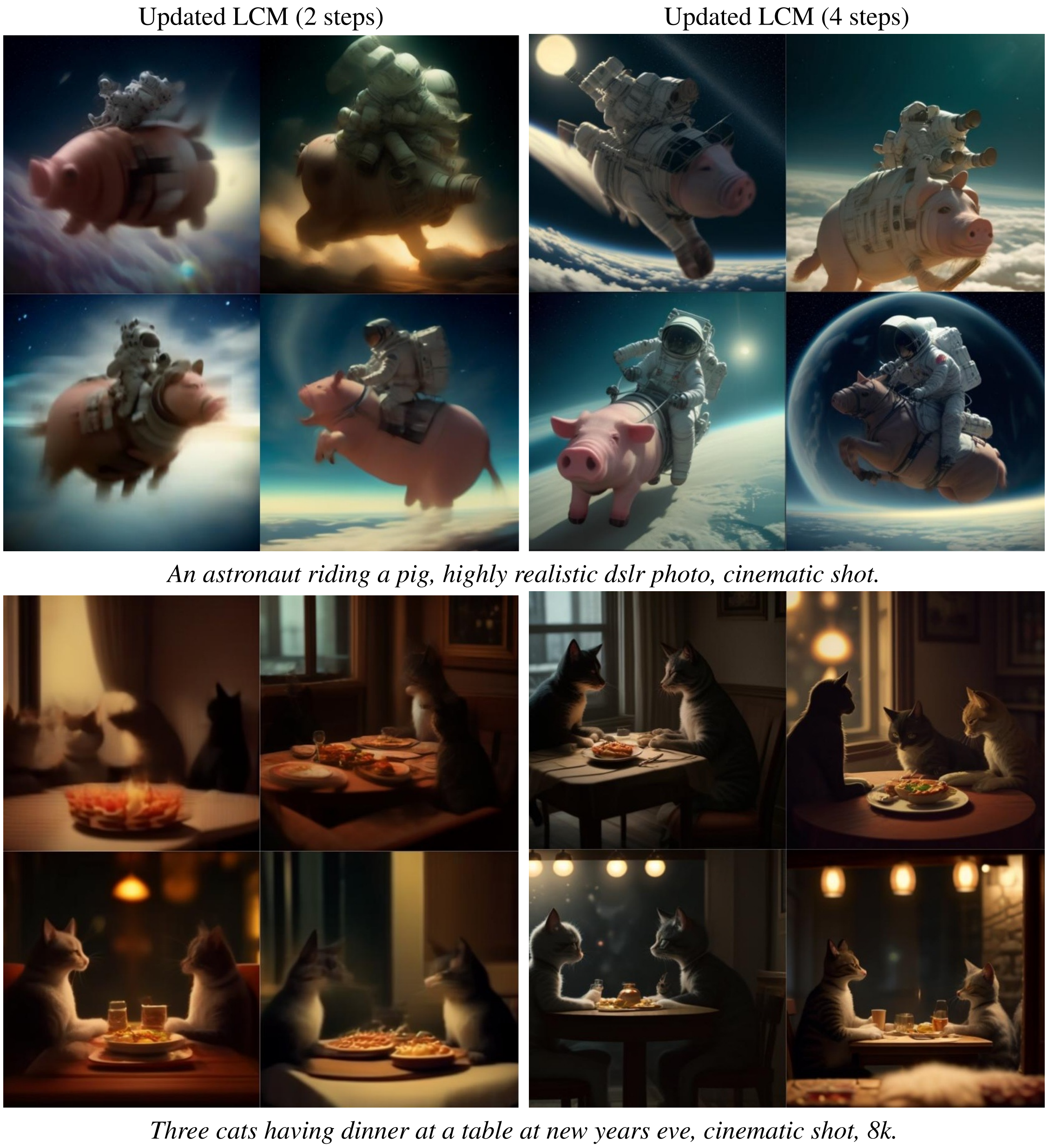
Updated LCM (2 steps)

Updated LCM (4 steps)

Cute small corgi sitting in a movie theater eating popcorn, unreal engine.

A Pikachu with an angry expression and red eyes, with lightning around it, hyper realistic style.

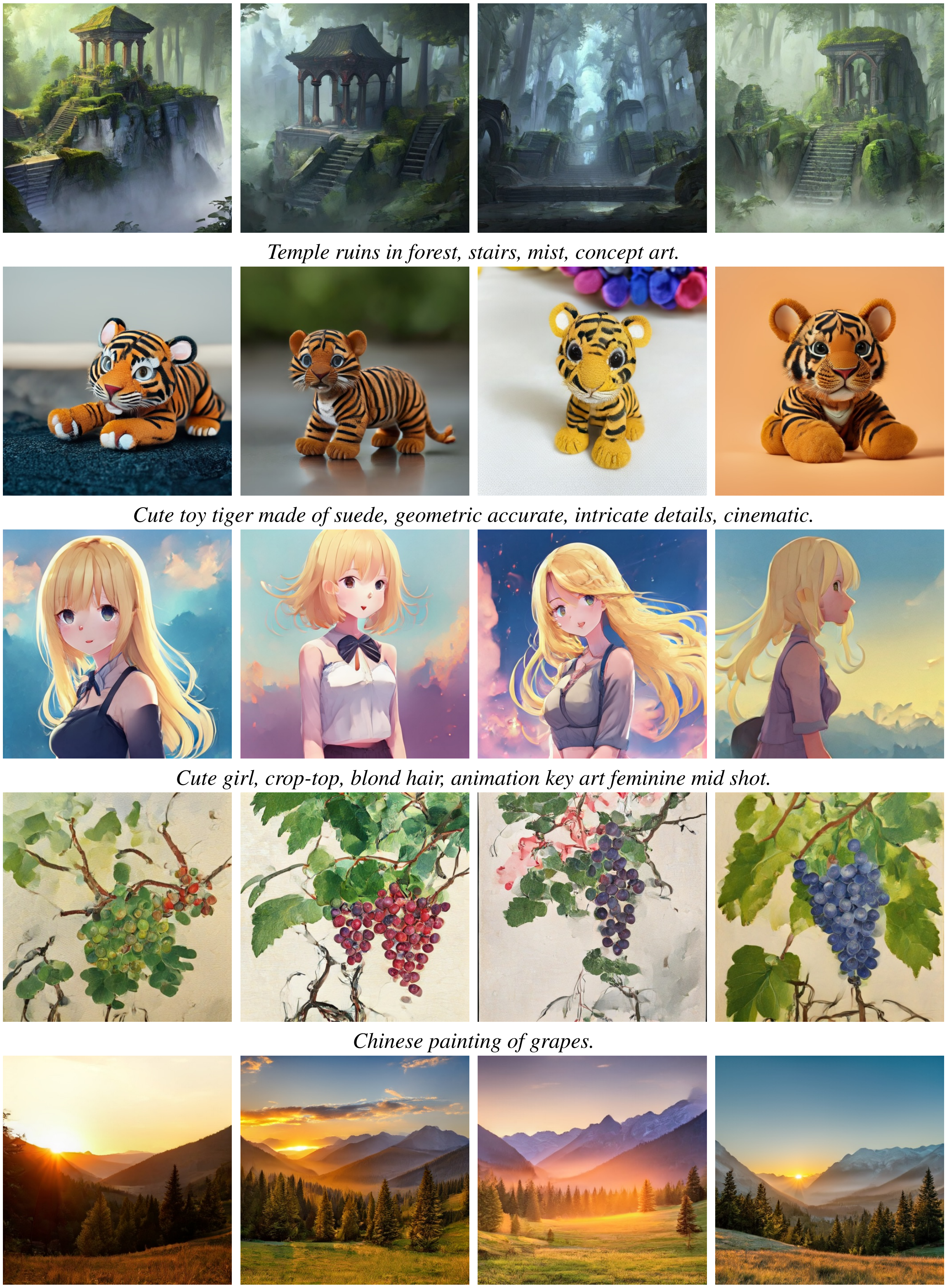
Table 18. Qualitative results of updated LCM model.



An astronaut riding a pig, highly realistic dslr photo, cinematic shot.

Three cats having dinner at a table at new years eve, cinematic shot, 8k.

Table 19. Qualitative results of updated LCM model.



Temple ruins in forest, stairs, mist, concept art.

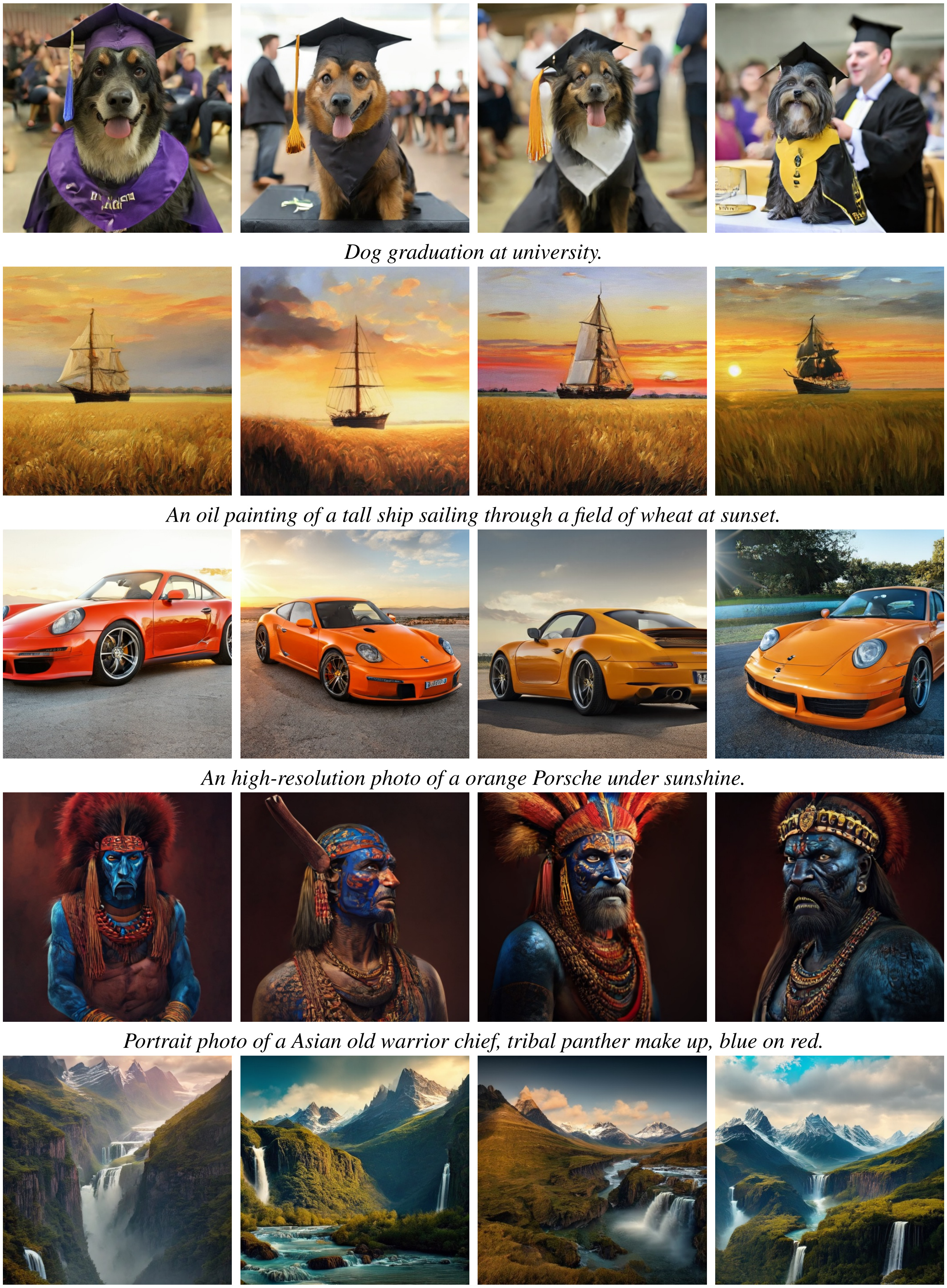
Cute toy tiger made of suede, geometric accurate, intricate details, cinematic.

Cute girl, crop-top, blond hair, animation key art feminine mid shot.

Chinese painting of grapes.

Sunset in a valley with trees and mountains.

Table 20. Additional qualitative results of UFOGen. Zoom-in for better viewing.



Dog graduation at university.

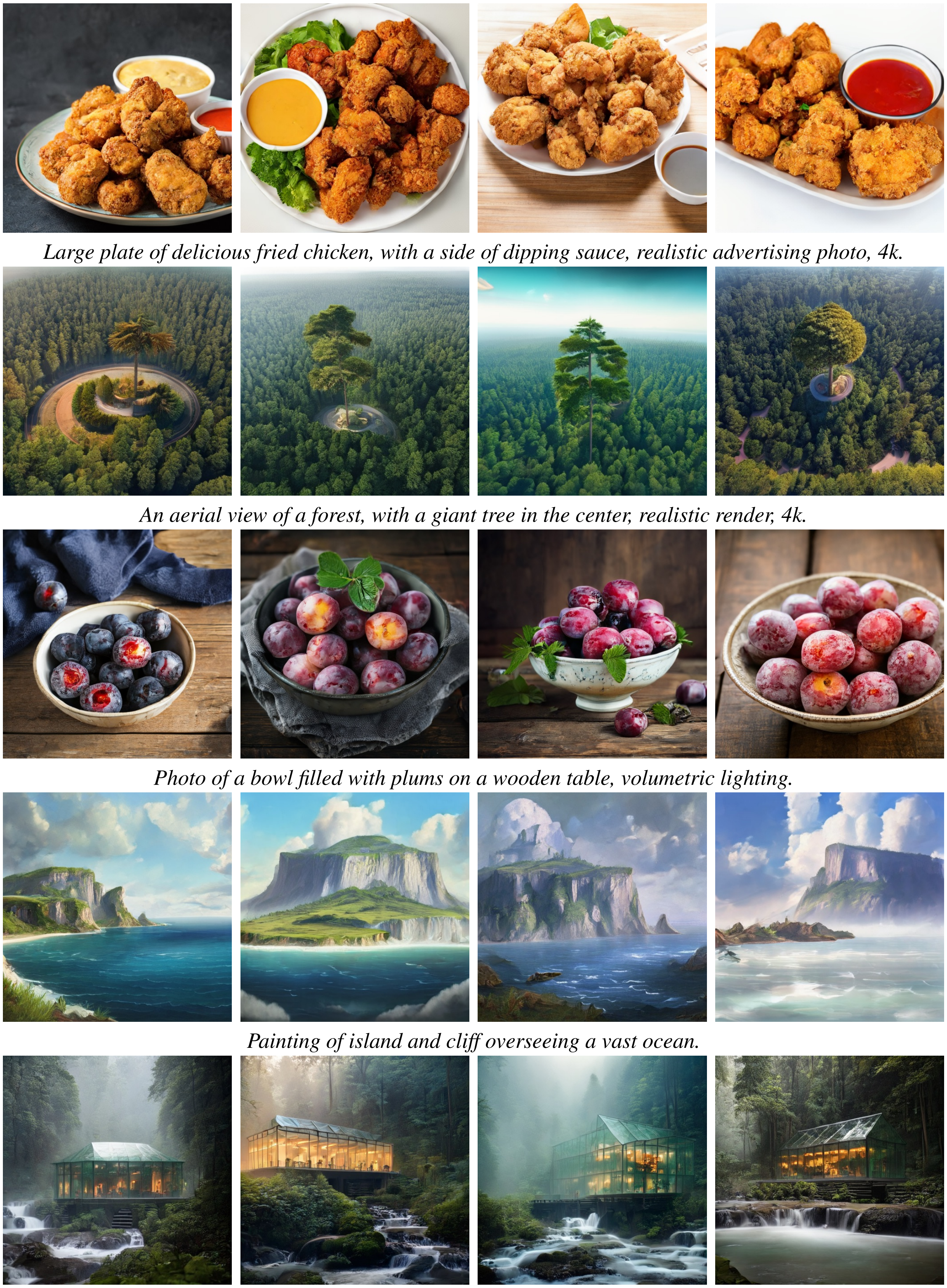
An oil painting of a tall ship sailing through a field of wheat at sunset.

An high-resolution photo of a orange Porsche under sunshine.

Portrait photo of a Asian old warrior chief, tribal panther make up, blue on red.

A close-up photo of a intricate beautiful natural landscape of mountains and waterfalls.

Table 21. Additional qualitative results of UFOGen. Zoom-in for better viewing.



Large plate of delicious fried chicken, with a side of dipping sauce, realistic advertising photo, 4k.

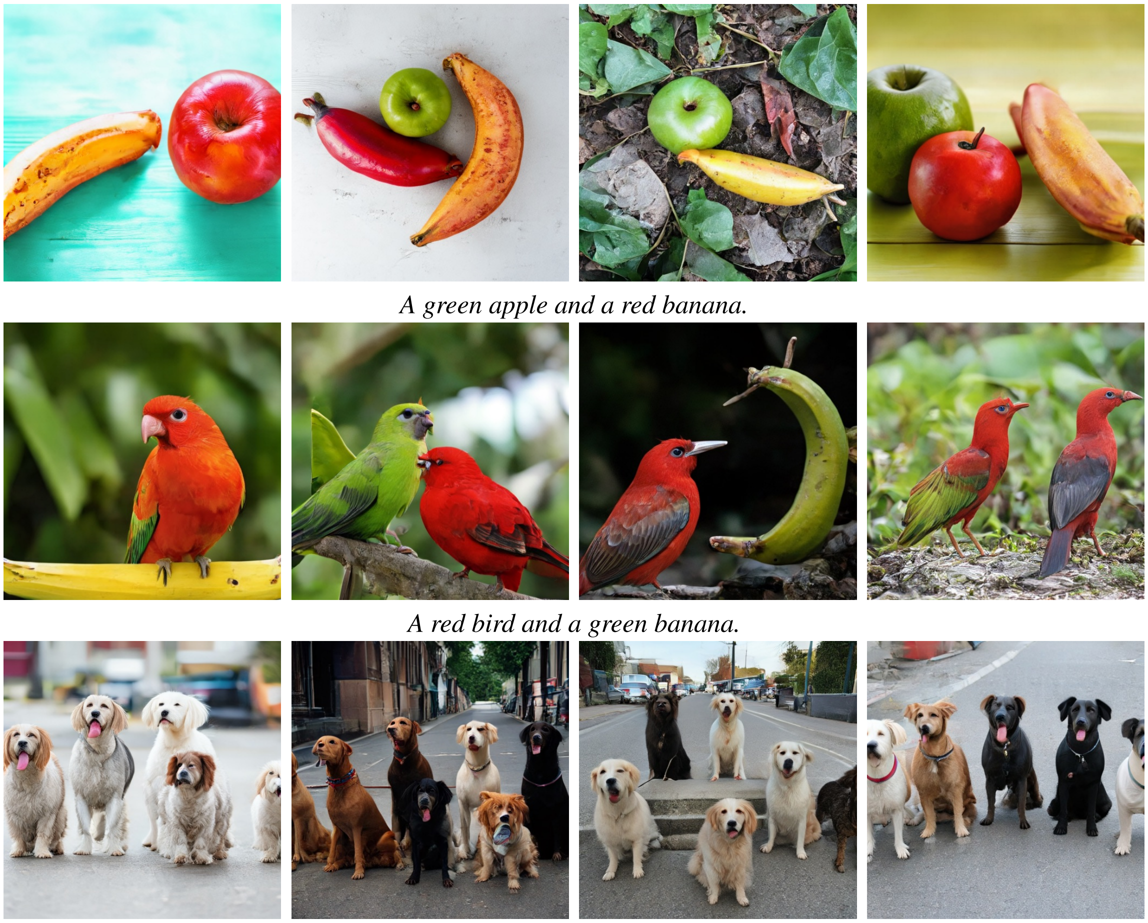
An aerial view of a forest, with a giant tree in the center, realistic render, 4k.

Photo of a bowl filled with plums on a wooden table, volumetric lighting.

Painting of island and cliff overseeing a vast ocean.

Photo of a modern glass house in the jungle, small stream flowing, mist, atmospheric.

Table 22. Additional qualitative results of UFOGen. Zoom-in for better viewing.

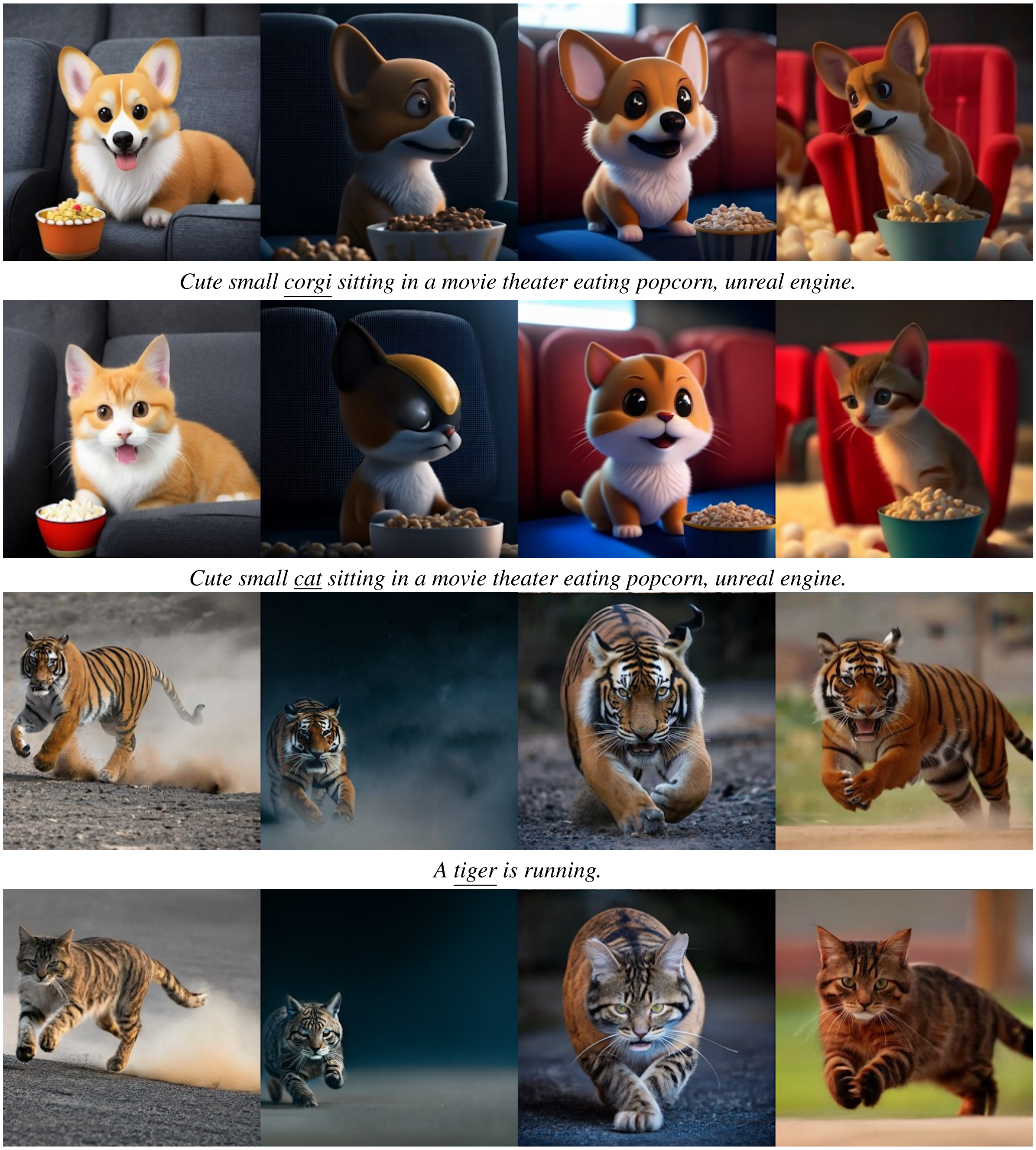


A green apple and a red banana.

A red bird and a green banana.

Four dogs on the street.

Table 23. Failure cases of UFOGen.



Cute small corgi sitting in a movie theater eating popcorn, unreal engine.

Cute small cat sitting in a movie theater eating popcorn, unreal engine.

A tiger is running.

A cat is running.

Figure 5. Extended results of image-to-image generation from UFOGen. For each group, we edit the images by adding noise and sightly modifying the prompt.

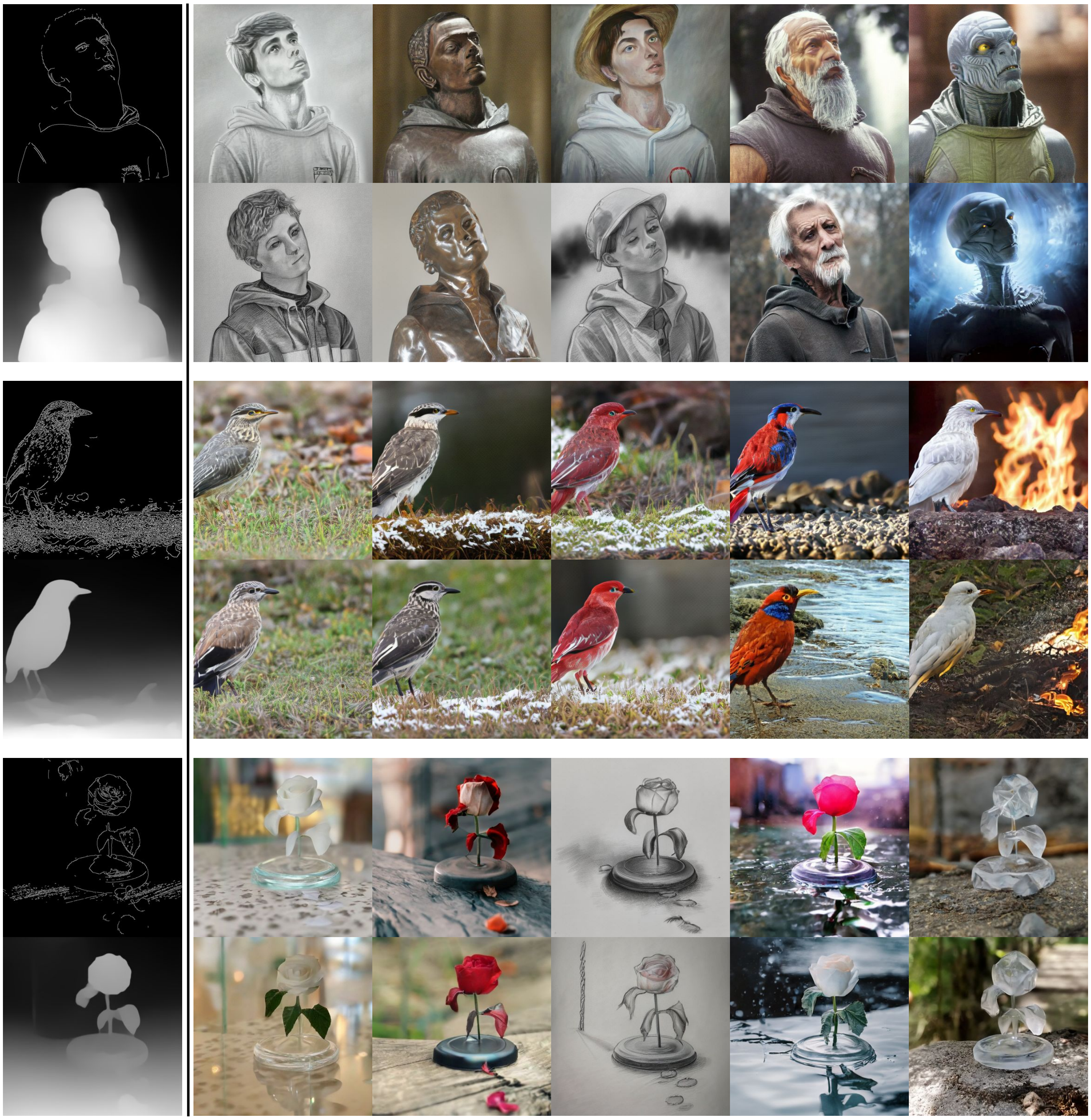


Figure 6. Extended results of controllable generation from UFOGen. For each group of canny edge and depth map images, we use same prompts per column.