A Closer Look at Time Steps is Worthy of Triple Speed-Up for Diffusion Model Training

Supplementary Material

A. More Detail of Experiments

In this section, we introduce detailed experiment settings, datasets, and architectures.

A.1. Architecture and Training Recipe.

We utilize Unet and DiT as our base architecture in the diffusion model. pre-trained VAE which loads checkpoints from huggingface is employed to be latent encoder. Following Unet implementation from LDM and DIT from official implementation, we provide the architecture detail in Tab. 6. We provide our basic training recipe and evaluation setting with specific details in Tab. 7.

A.2. Datasets

CIFAR-10. CIFAR-10 datasets consist of 32×32 size colored natural images divided into categories. It uses $50,000$ in images for training and EDM evaluation suite in image generation.

MetFaces is an image dataset of human faces extracted from works of art. It consists of 1336 high-quality PNG images at 1024×1024 resolution. We download it at 256 resolution from kaggle.

FFHQ is a high-quality image dataset of human faces, contains 70,000 images. We download it at 256x256 resolution from kaggle.

ImageNet-1K is the subset of the ImageNet-21K dataset with $1,000$ categories. It contains $1,281,167$ training images and $50,000$ validation images.

MSCOCO is a large-scale text-image pair dataset. It contains 118K training text-image pairs and 5K validation images. We download it from official website.

FaceForensics is a video dataset consisting of more than 500,000 frames containing faces from 1004 videos that can be used to study image or video forgeries.

A.3. Detail of MDT + SpeeD Experiment

MDT utilizes an asymmetric diffusion transformer architecture, which is composed of three main components: an encoder, a side interpolater, and a decoder. During training, a subset of the latent embedding patches is randomly masked using Gaussian noise with a masking ratio. Then, the remaining latent embedding, along with the full latent embedding is input into the diffusion model.

Following the official implementation of MDT, We utilize DiT-S/2 and MDT-S/2 as our base architecture, whose total block number both is 12 and the number of decoder layers in MDT is 2. We employ the AdamW [39] optimizer with constant learning rate 1e-4 using 256 batch size without weight decay on class-conditional ImageNet with an image resolution of 256^2. We perform training on the class-conditional ImageNet dataset with images of resolution 256x256. The diffusion models are trained for a total of 1000K iterations, utilizing a mask ratio of 0.3.

A.4. Detail of FDM + SpeeD Experiment

FDM add the momentum to the forward diffusion process with a scale that control the weight of momentum for faster convergence to the target distribution. Following official implementation, we train diffusion models of EDM and FDM. We retrain these official network architecture which is U-Net with positional time embedding with dropout rate 0.13 in training. We adopt Adam optimizer with learning rate 1e-3 and batch size 512 to train each model by a total of 200 million images of 32^2 CIFAR-10 dataset. During training, we adopt a learning rate ramp-up duration of 10 Mimgs and set the EMA half-life as 0.5 Mimgs. For evaluation, EMA models generate 50K images using EDM sampler based on Heun's 2^{nd} order method [61].

A.5. Text-to-Image Experiment Detail

In text to image task, diffusion models synthesize images with textual prompts. For understanding textual prompts, text-to-image models need semantic text encoders to encode language text tokens into text embedding. We incorporate a pre-trained CLIP language encoder, which processes text with a maximum token length of 77. DiT-XL/2 is employed as our base diffusion architecture. We employ AdamW optimizer with a constant learning rate 1e-4 without weight decay. We train text-to-image diffusion models for 400K training iterations on MS-COCO training dataset and evaluate the FID and CLIP score on MS-COCO validation dataset. To enhance the quality of conditional image synthesis, we implement classifier-free guidance with 1.5 scale factor.

A.6. Theoretical Analysis

A.6.1. Notations

In this section, we will introduce the main auxiliary notations and the quantities that need to be used. The range of schedule hyper-parameter group $\{\beta_t\}_{t\in[T]}$ turns out to be $t = 1$ to $t = T$. For analytical convenience, we define β_0 as $\beta_0 := \beta_1 - \Delta_\beta/T$.

architecture	input size	input channels	patch size	model depth	hidden size	attention heads
U-Net	32×32	3	-	8	128	1
DiT-XL/2	32×32	4	2	28	1152	16
DiT-S/2	32×32	4	2	12	384	6

Table 6. Architecture detail of Unet and DiT on MetFaces, FFHQ, and ImageNet.

	MetFaces 256×256	FFHQ 256×256	ImageNet-1K 256×256
latent size	$32 \times 32 \times 3$	$32 \times 32 \times 3$	$32 \times 32 \times 3$
class-conditional			✓
diffusion steps	1000	1000	1000
noise schedule	linear	linear	linear
batch size	256	256	256
training iterations	50K	100K	400K
optimizer	AdamW	AadamW	AdamW
learning rate	1e-4	1e-4	1e-4
weight decay	0	0	0
sample algorithm	DDPM	DDPM	DDPM
number steps in sample	250	250	250
number sample in evaluation	$10,000$	$10,000$	$10,000$

Table 7. Our basic training recipe based on MetFaces, FFHQ, ImageNet datasets

Another auxiliary notation is forward ratio ρ_t, which is defined as $\rho_t = t/T$. Forward ratio provide an total number free notation for general diffusion process descriptions.

Based on the two auxiliary notations β_0 and ρ_t, the expression of β_t with respect to the forward process ratio is $\beta_t = \beta_0 + \Delta_\beta \rho_t$.

The relationship between α_t and β_t is recalled and re-written as follows: $\alpha_t = 1 - \beta_t = 1 - \beta_0 - \Delta_\beta \rho_t$. $\bar{\alpha}_t$ the multiplication of α_t is re-written as $\bar{\alpha}_t = \Pi_{s=1}^{t}(1 - \beta_0 - \Delta_\beta \rho_s)$.

Perturbed samples' distribution: $x_t|x_0 \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I})$

A.6.2. Auxiliary Lemma and Core Theorem

Lemma 1 (Bounded α by β)**.** *In DDPM [20], using a simple equivariant series $\{\beta_t\}_{t\in[T]}$ to simplify the complex cumulative products $\{\bar{\alpha}_t\}_{t\in[T]}$, we obtain the following auxiliary upper bound of $alpha_t$.*

$$\bar{\alpha}_t \leq \exp\{-(\beta_0 t + \frac{\Delta_\beta t^2}{2T})\}$$

A.6.3. Propositions

Proposition A.1 (Jensen's inequality)**.** *If f is convex, we have:*

$$\mathbf{E}_X f(X) \geq f(\mathbf{E}_X X).$$

A variant of the general one shown above:

$$||\sum_{i\in[N]} x_i||^2 \leq N \sum_{i\in[N]} ||x_i||^2.$$

Proposition A.2 (triangle inequality)**.** *The triangle inequality is shown as follows, where $||\cdot||$ is a norm and A, B is the quantity in the corresponding norm space:*

$$||A + B|| \leq ||A|| + ||B||$$

.

Proposition A.3 (matrix norm compatibility)**.** *The matrix norm compatibility, $A \in \mathbb{R}^{a\times b}, B \in \mathbb{R}^{b\times c}, v \in \mathbb{R}^b$:*

$$||AB||_m \leq ||A||_m ||B||_m$$
$$||Av||_m \leq ||A||_m ||v||.$$

Proposition A.4 (Peter Paul inequality)**.**

$$2\langle x, y\rangle \leq \frac{1}{\epsilon}||x||^2 + \epsilon||y||^2$$

.

A.6.4. Proof of Lemma 1

Proof. To proof the auxiliary Lemma 1, we re-arrange the notation of $\bar{\alpha}_t$ as shown in Section A.6.1, and we have the

Table 8. The ingredients of generalized curves $\dot{\Delta}$ and $\dot{\Sigma}$ schedules about mainstream SDE designs, including VP, VE [59], EDM [28].

Schedules	s	σ^2	$\dot{s}$	$\dot{\sigma}$
VP	$\exp\{-\frac{1}{4}\Delta_\beta t^2 - \frac{1}{2}\beta_0 t\}$	$\exp\{\frac{1}{2}\Delta_\beta t^2 + \beta_0 t\} - 1$	$-\frac{\sigma\dot{\sigma}}{(1+\sigma^2)^{3/2}}$	$\frac{(1+\sigma^2)(\Delta_\beta t+\beta_0)}{2\sigma}$
VE	1	t	0	1
EDM	1	t^2	0	$2t$

following upper bound:

$$\begin{aligned}\log\bar{\alpha}_t &= \sum_{s=1}^{t}\log(1-\beta_0-\Delta_\beta\rho_s)\\ &\le t\log(\frac{1}{t}\sum_{s=1}^{t}(1-\beta_0-\Delta_\beta\rho_s))\\ &= t\log(1-\beta_0-\Delta_\beta\frac{1}{t}\sum_{s=1}^{t}\frac{s}{T})\\ &= t\log(1-\beta_0-\Delta_\beta\frac{t+1}{2T})\\ &\le -(\beta_0 t+\frac{\Delta_\beta(t+1)t}{2T}),\end{aligned}$$

where the two inequalities are by the concavity of $\log$ function and the inequality: $\log(1+x)\le x$. Taking exponents on both sides simultaneously, we have:

$$\bar{\alpha}_t \le \exp\{-(\beta_0 t+\frac{\Delta_\beta t^2}{2T})\}.$$

□

A.6.5. Proof of Theorem 1

Before the proof of the theorem, we note that the samples $x_t|x_0 \sim \mathcal{N}(\mu_t,\sigma_t)$ have the following bounds with Lemma 1:

- Reformulate the expression of $\sqrt{\bar{\alpha}}$, we have the mean vector μ_t's components $\dot{\mu}_t$ bounded by $\dot{x}_0$ the corresponding components of data x_0 as follows:

$$\dot{\mu}_t = \sqrt{\bar{\alpha}_t}\dot{x}_0 \le \exp\{-\frac{1}{2}(\beta_0 t+\frac{\Delta_\beta t^2}{2T})\}\dot{x}_0,$$

- Reformulate the expression of $\bar{\alpha}$, we have a partial order relation on the cone about covariance matrix of $x_t|x_0$ as follows:

$$\sigma_t = (1-\bar{\alpha}_t)\mathbf{I} \succeq (1-\exp\{-(\beta_0 t+\frac{\Delta_\beta t^2}{2T})\})\mathbf{I}.$$

Proof. The process increment at given t^{th} time step is $\delta_t = x_{t+1}-x_t$. δ_t is a Gaussian process as follows:

$$\delta_t \sim \mathcal{N}(\underbrace{(\sqrt{\alpha_{t+1}}-1)\sqrt{\bar{\alpha}_t}x_0}_{\phi_t}, \underbrace{[2-\bar{\alpha}_t(1+\alpha_{t+1})]\mathbf{I}}_{\Psi_t})$$

The theorem's key motivation is that the label is noisy, and noisy magnitude is measured by mean vector's norm $||\phi_t||$ and covariance matrix Ψ_t.

The upper bounds of mean vectors' norm and the partial order of covariance matrix at different time step t are shown as follows:

$$\begin{aligned}||\phi_t||^2 &\le (\sqrt{\alpha_{t+1}}-1)^2\bar{\alpha}_t||\mathbb{E}x_0||^2\\ &\le (1-\alpha_{t+1})\bar{\alpha}_t||\mathbb{E}x_0||^2\\ &\le \underbrace{(\beta_0+\Delta_\beta\rho_{t+1})}_{\beta_{t+1}}\exp\{-\underbrace{(\beta_0+\frac{\Delta_\beta t}{2T})}_{\beta_{t/2}}t\}||\mathbb{E}x_0||^2\\ &\le \beta_{\max}\exp\{-\underbrace{(\beta_0+\frac{\Delta_\beta t}{2T})}_{\beta_{t/2}}t\}||\mathbb{E}x_0||^2\end{aligned}$$

where the inequalities are by Lemma 1, $(1-x)^2 \le (1-x^2) = (1-x)(1+x)$, when $x\in[0,1]$, and $\beta_{t+1}\le\beta_{\max}$

$$\begin{aligned}\Psi_t &= [2(1-\bar{\alpha}_t)+\bar{\alpha}_t(\beta_0+\Delta_\beta\rho_{t+1})]\mathbf{I}\\ &\succeq 2(1-\exp\{-\underbrace{(\beta_0+\frac{\Delta_\beta t}{2T})}_{\beta_{t/2}}t\})\mathbf{I}+\bar{\alpha}_t\beta_{t+1}\mathbf{I}\\ &\succeq 2(1-\exp\{-\underbrace{(\beta_0+\frac{\Delta_\beta t}{2T})}_{\beta_{t/2}}t\})\mathbf{I}\end{aligned}$$

where the inequalities are by Lemma 1 and $\bar{\alpha}_t\beta_{t+1}\mathbf{I}\succeq\mathbf{0}$. The residual term is

$$\bar{\alpha}_t\beta_{t+1} = \beta_{t+1}\Pi_{s=1}^{t}(1-\beta_s) \ge \exp\{\log\beta_{t+1}+t\log(1-\beta_t)\}$$

□

B. More Experiment Results

Efficiency comparisons. In Fig. 8, besides the Min-SNR and CLTS, we show the efficiency comparison with P2 and Log-Normal methods. One can find that our method consistently accelerates the diffusion training in large margins.

Super resolution with SpeeD. We employ SpeeD to super-resolution image generation on 512 × 512 MetFaces compared with vanilla DiT. We train DiT-XL/2 for 100K training iterations and compare the FID score at 50K, 100K training iterations.

Table 9. Super resolution.

Method	50K	100K
DiT-XL/2	77.9	35.4
SpeeD	48.7	10.6

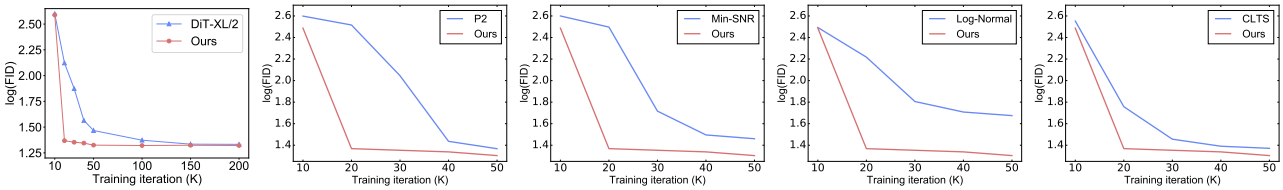

Figure 8. More efficiency comparison on MetFaces.

The batch size is 32 for saving the GPU memory. As shown in 9, SpeeD obtain better performance than vanilla DiT at same training iterations on 512^2 MetFaces dataset. It indicates that SpeeD can achieve training acceleration on super-resolution tasks.

B.1. Additional Experiments

Experimetns on DiT-S/8. Further comparison on DiT of smaller scales are reported in Tab. 10. The datasets include ImageNet-1K and Celeb-A.

#steps	DiT-S/8	P2	Min-SNR	Log-normal	CLTS	ours
ImageNet-1K (FID ↓)						
10K	399.9	398.6	398.4	399.7	399.6	400.8
20K	380.0	365.0	368.0	387.5	381.9	379.2
40K	200.0	207.6	208.6	365.9	231.6	**191.5**
Celeb-A (FID ↓)						
10K	408.7	412.5	408.7	410.4	408.0	407.8
20K	386.7	366.8	386.7	394.0	386.4	377.9
40K	271.6	271.1	271.6	293.0	258.8	**254.9**
Celeb-A (IS ↑)						
10K	1.50	1.49	1.50	1.50	1.50	1.50
20K	1.49	1.48	1.49	1.50	1.49	1.63
40K	3.29	3.70	3.29	2.55	3.46	**3.87**

Table 10. Comparison on ImageNet-1K and Celeb-A. FID and IS are reported. The baseline is measured on DiT-S/8, with global batchsize of 16. Other settings are default.

More baslines. Comparison between SpeeD and BS [78] and B-TTDM [79] are shown in Tab. 11.

Detailed ablation study. FID-10K on 10K/20K/40K/50K iterations are provided in Tab. 12 the model is DiT-S/8 with batchsize of 16.

B.1.1. Detailed Training Process

The detailed training process on FFHQ through 100K iterations are shown in Tab. 13.

C. More Related Works

We discuss other works related to SpeeD, including Text to Image and Video generation. Another point to mention is that we learn from InfoBatch [48] in writing.

Text to image generation with diffusion models Text-to-image generation has emerged as a hotly contested and rapidly evolving field in recent years, with an explosion of related industrial products springing up [2, 6, 11, 51, 54].

#steps	BS [78]	B-TTDM [79]	ours
100K	155.9	157.4	**155.2**
200K	152.2	**150.9**	151.7
400K	141.8	140.5	**139.2**

Table 11. New baselines to be added. The settings follow BS. (FID)

#steps	10K	20K	40K	50K
DiT-S/8	399.9	380.0	200.0	–
$\lambda = 0.5$	400.7	376.7	207.4	202.1
$\lambda = 0.6$	400.8	379.2	191.5	191.2
$\lambda = 0.8$	400.3	379.2	203.2	200.5
$\tau = 600$	401.0	382.6	210.1	198.5
$\tau = 700$	400.8	379.2	191.5	191.2
$\tau = 800$	399.5	380.9	200.1	200.0
$k = 1$	400.4	388.0	214.5	202.4
$k = 2$	400.8	379.2	191.5	191.2
$k = 10$	400.4	380.5	231.7	206.7

Table 12. Detailed ablation across training steps. FID-10K is reported. DiT-S/8 serves as the baseline. Global batchsize is 16.

Convert textual descriptions into corresponding visual content, models not only learn to synthesize image content but also ensuring alignment with the accompanying textual descriptions. To better align images with textual prompt guidance, previous work has primarily focused on enhancements in several schemes including strengthening the capacity of text encoder [49, 50] improving the condition plugin module in diffusion model [71], improving data quality [2].

Video generation with diffusion models. As diffusion models achieve tremendous success in image generation, video generation has also experienced significant breakthroughs, marking the field's evolution and growth. Inspired by image diffusion, pioneering works such as RVD [68] and VDM [22] explore video generation using diffusion methods. Utilizing temporal attention and latent modeling mechanisms, video diffusion has advanced in terms of generation quality, controllability, and efficiency [15, 18, 21, 56, 63, 65, 70, 81]. Notably, Stable Video Diffusion [3] and Sora [5] achieve some of the most appealing results in the field.

Other diffusion acceleration works To achieve better results with fewer NFE steps, Consistency Models [60] and Consistency Trajectory Models [29] employ consistency loss and novel training methods. Rectified Flow [36], followed by Instaflow [37], introduces a new perspective to

iterations (K)	10	20	30	40	50	60	70	80	90	100
DiT-XL/2	356.1	335.3	165.2	35.8	12.9	11.9	10.5	9.6	8.7	7.8
SpeeD	322.1	320.0	91.8	19.8	9.9	7.6	7.1	6.6	6.2	5.8

Table 13. Details about training to 100K on FFHQ.

obtain straight ODE paths with enhanced noise schedule and improved prediction targets, together with the reflow operation. DyDiT [73] incorporates dynamic neural networks [16, 72, 73] into diffusion models, achieving significant acceleration.

D. Visualization

Visualizations of the generated images. The figures above illustrate the quality of images generated by our method across various datasets, including CIFAR-10, FFHQ, MetFaces, and ImageNet-1K. In Fig. 9, the generated images from the CIFAR-10 dataset display distinct and recognizable objects, even for challenging categories. Fig. 10 presents generated images from the FFHQ dataset, showcasing diverse and realistic human faces with varying expressions and features. Fig. 11 exhibits images from the MetFaces dataset, depicting detailed and lifelike representations of artistic portraits. Finally, Fig. 12 includes images from the ImageNet-1K dataset, featuring a wide range of objects and scenes with excellent accuracy and visual fidelity. These results emphasize the superior performance of our method in generating high-quality images across different datasets, indicating its potential for broader applications in image synthesis and computer vision tasks.

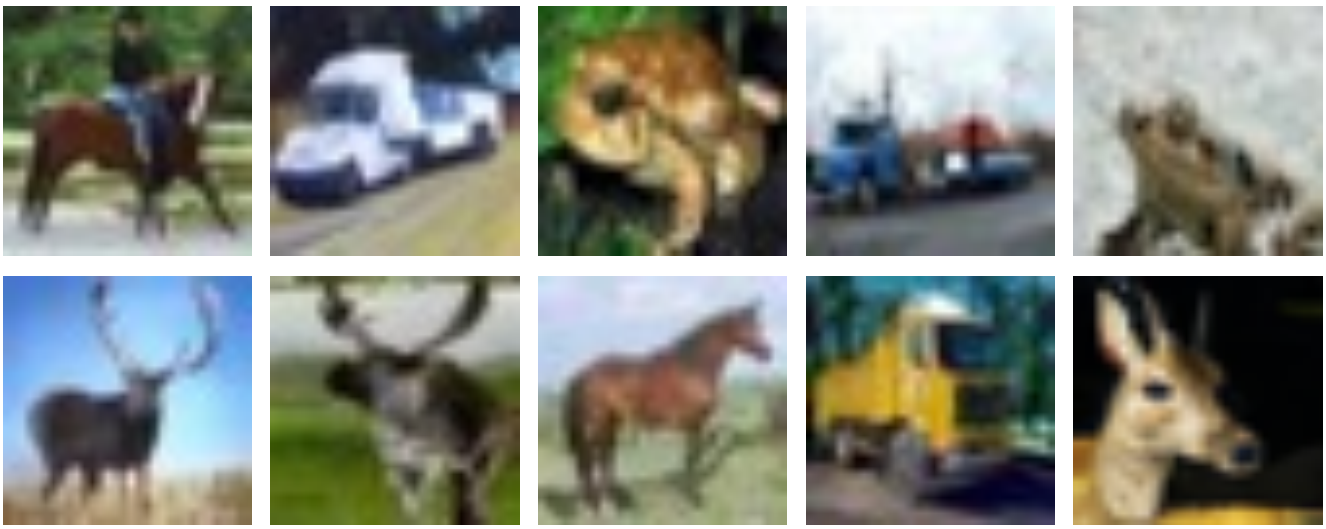

Figure 9. Generated images of CIFAR-10.

Figure 10. Generated images of FFHQ.

Figure 11. Generated images of MetFaces.

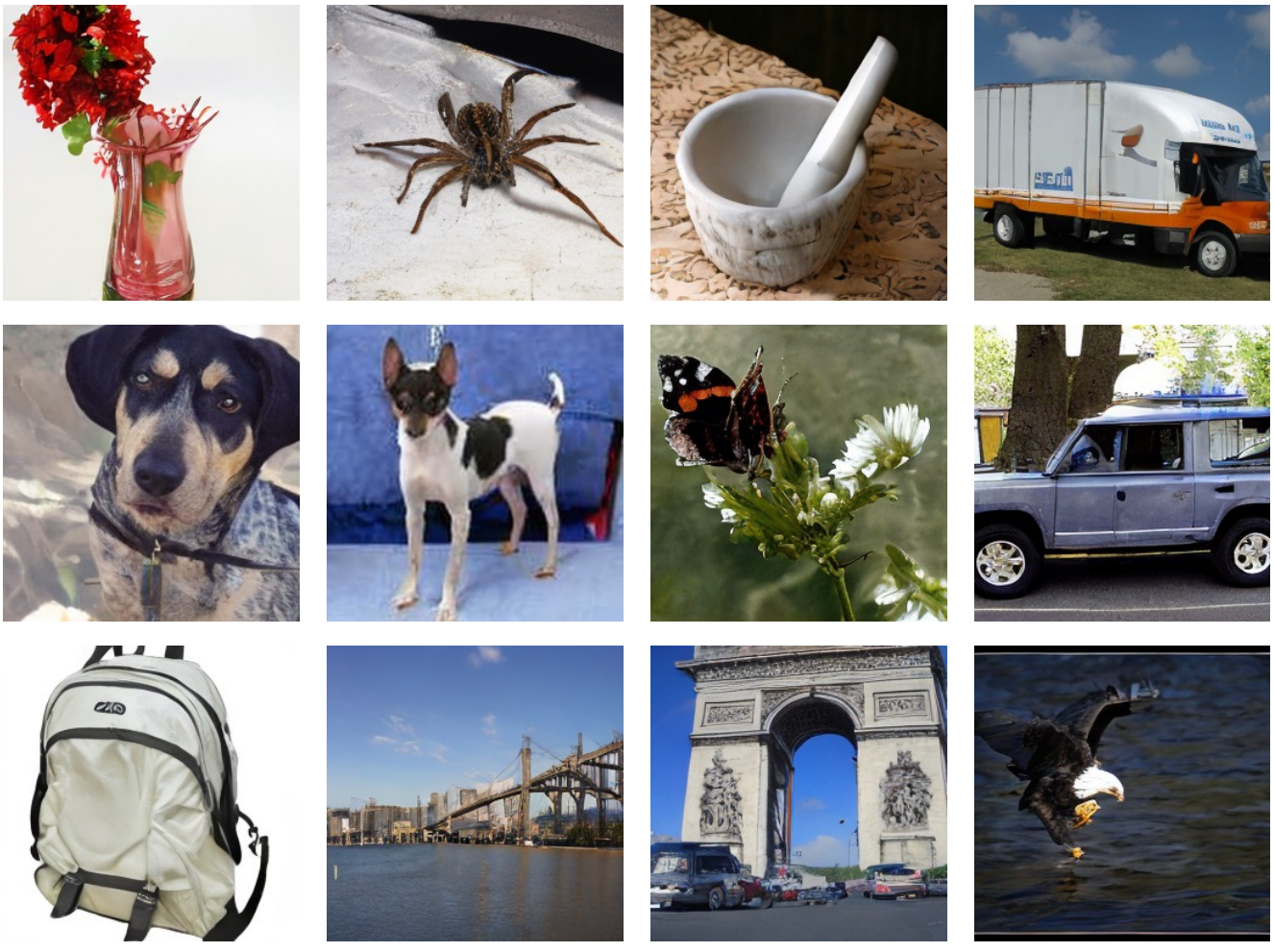

Figure 12. Generated images of ImageNet-1K.