Deeply Supervised Flow-Based Generative Models -Supplementary Material-

Inkyu Shin Chenglin Yang Liang-Chieh Chen

ByteDance Seed

A. Appendix

In the appendix, we provide additional information as listed below:

- Appendix A.1 provides additional analysis on feature distance.
- Appendix A.2 provides the design choices of VeRA block.
- Appendix A.3 provides the details of hyperparameters and implementations.
- Appendix A.4 provides the analysis on different number of samplings.
- Appendix A.5 provides the category-level results on GenEval benchmark.
- Appendix A.6 provides additional qualitative results on image generation benchmarks.
- Appendix A.7 provides the details of datasets and metrics used for experiments.
- Appendix A.8 provides the discussion and limitation on our method.

A.1. Additional Analysis on Feature Distance

To further investigate the alignment of velocity features across layers, we analyze the feature distance between all intermediate layers and the final layer. This extends the analysis from Figure 2 in the main paper, where only the distance between the features of key layer and the final layer was measured. By examining the full layer-wise distance trends, we can better understand how intermediate representations evolve toward the final velocity feature. As shown in Figure 1, DeepFlow consistently reduces the feature distance across layers, ensuring a smooth progression toward the final layer representation. Even with deep supervision and the integration of the VeRA block at the key layer (6th), DeepFlow maintains effective feature alignment throughout the network.

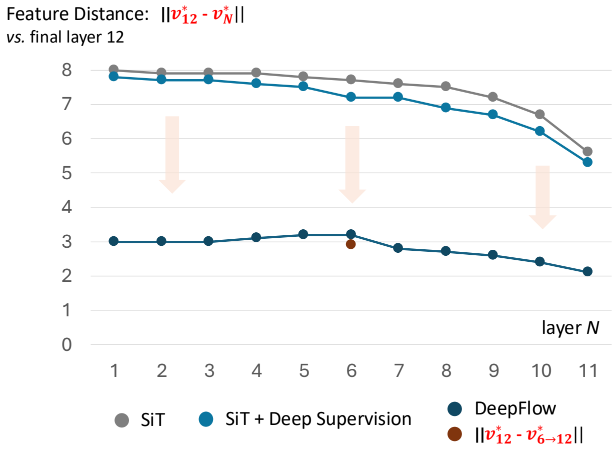
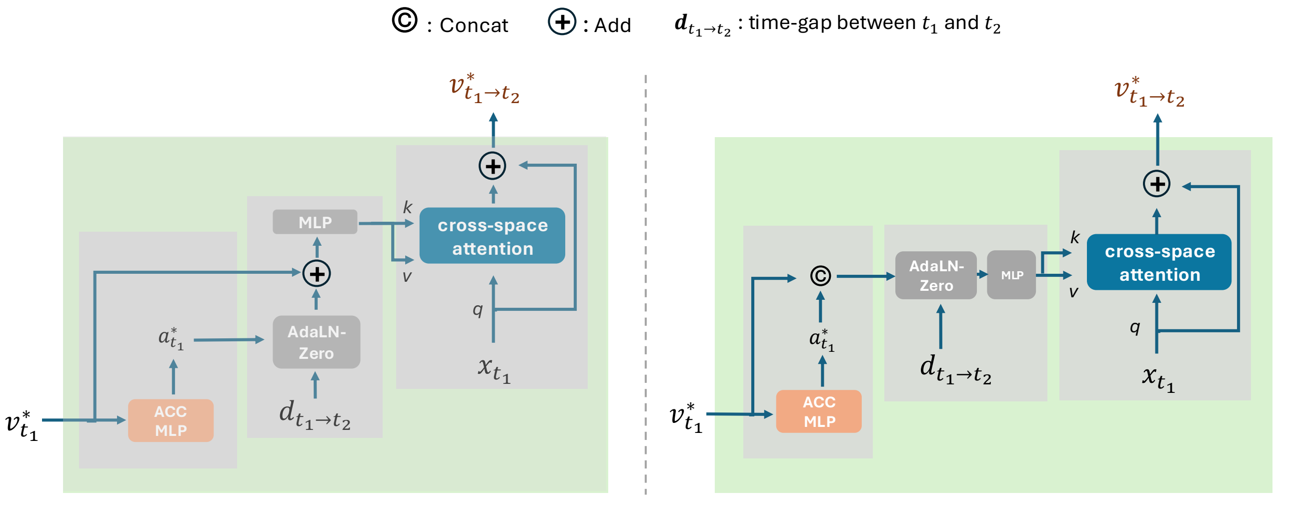


Figure 1. **Velocity Feature Distance between All Layers and Final Layer.** We provide additional analysis on feature distance to quantify the alignment between velocity features at each layer and one in the final layer. The results demonstrate that DeepFlow effectively aligns all intermediate features with the final one, even when deep supervision and the VeRA block are applied to a key layer (6th).

A.2. Design Choices of VeRA Block

We present the design choices for the VeRA block, a core component of our DeepFlow, as illustrated in Figure 2. Both designs leverage acceleration to refine preceding velocity features using an ACC MLP, adaptive layer normalization, and cross-space attention. The design in the left panel is motivated by first-order dynamics using addition of velocity and modulated acceleration. Specifically, $a^*_{t_1}$ from ACC MLP is modulated by $d_{t_1 \rightarrow t_2}$, and added with $v^*_{t_1}$. In base configuration, this approach achieves 29.3 FID, outperforming SiT-B/2 [11] (34.4 FID)—but underperforms compared to



VeRA Block with Addition of Velocity and Acceleration

VeRA Block with Concatenation of Velocity and Acceleration

Figure 2. **Design Choice for VeRA block.** The left panel utilizes addition of velocity and acceleration, while right panel (proposed VeRA block) is differentiated by modulating concatenated feature of velocity and acceleration.

the proposed VeRA block (in the right panel of Figure 2). Although the left design adheres more closely to first-order dynamics, modulating acceleration alone with a time-gap is insufficient to fully adjust the preceding velocity features. In contrast, our proposed VeRA optimizes feature alignment by modulating a concatenation of velocity and acceleration features, which results in superior generation performance (28.1 FID).

A.3. Hyperparameters and Implementations

We provide detailed explanation about hyperparameters and implementations used for DeepFlow in following orders.

- **Image Encoder:** We utilize VAE [8] encoder to pre-compute the latent feature of input as what SiT [11] and REPA [16] did. The checkpoint of VAE encoder is from *stability/sd-vae-ft-ema*, which was pre-trained in Stable Diffusion [14]. Then, we flatten the latent features with patch size of 2.

- **Transformer Blokcs:** We employ same setting of DiT [12] to construct transformer blocks including branches of pre- and post-VeRA block. What we differentiate is we condition them with different time-step during training to train VeRA block with time-gap prior. We set time-gap to be same or under 0.01 as we ablated in main paper.

- **VeRA Block:** As the first core block of VeRA block, ACC MLP consists of 4 linear layers with SiLU [2] activation. Then, adaptive layer normalization with zero-initialization for final linear inputs time-gap to produce scale and shift for concatenated features of velocity and acceleration. For final part, cross-space attention module is performed with layer pre-norm modulated velocity feature space (key and value) and pre-norm spatial feature space (query).

- **Optimizer and Training:** To optimizer baselines and our DeepFlow, we utilize AdamW [10] with constant learning rate of 1e-4, $(\beta_1, \beta_2) = (0.9, 0.999)$ without weight decay and train the models with batch size of 256. For faster training, all of the experiments including DeepFlow and baselines were conducted using Pytorch Accelerate [5] pipeline with mixed-precision (fp16), and A100 GPUs.

- **SSL Alignment:** As demonstrated in our main paper, we employ an SSL encoder for additional feature alignment, following the approach of REPA [16]. Unlike REPA, which aligns a manually selected key layer with the SSL encoder, we incorporate external alignment after the output of each VeRA block in a more unified manner. For instance, in DeepFlow-B/2-2T with SSL alignment, the refined features produced by the VeRA block are further aligned using either DINO$_{v1}$ or DINO$_{v2}$. In DeepFlow-XL/2-3T with SSL alignment, DINO$_{v2}$ is applied twice—once after each VeRA block. Notably, we also experimented with applying SSL

model	SSL align	Overall↑	Single object	Two object	Counting	Colors	Position	Color attr.
SiT-24 [11]	✗	0.2672	0.8312	0.1364	0.2062	0.4069	0.0200	0.0025
	DINOv2	0.3166	0.8969	0.2778	0.2031	0.4495	0.0475	0.0250
DeepFlow-24-3T	✗	0.2957	0.8625	0.1919	0.2156	0.4468	0.0250	0.0325
	DINOv2	0.3458	0.9500	0.3460	0.2406	0.4681	0.0325	0.0375

Table 1. **Zero-Shot Text-to-Image Generation Results on GenEval benchmark.** We trained models with MS-COCO [9], following the training setting of REPA [16] and evaluated them with GenEval [4] benchmark.

model	SSL align	CFG	FID↓	sFID↓	IS↑
DeepFlow-XL/2-3T	✗	1.3	1.98	4.39	256.7
	✗	1.325	1.97	4.39	264.7
	✗	1.35	2.00	4.4	271.6
	DINOv2	1.275	1.78	4.45	263.4
	DINOv2	1.3	1.77	4.44	271.3
	DINOv2	1.325	1.80	4.44	277.7

Table 2. **Optimal CFG [6] Scale Search.** We tested DeepFlow-XL/2-3T (trained with 400 epochs) with different CFG (classifier-free guidance) scales.

alignment twice in the original SiT [11], but this did not lead to any performance improvement.

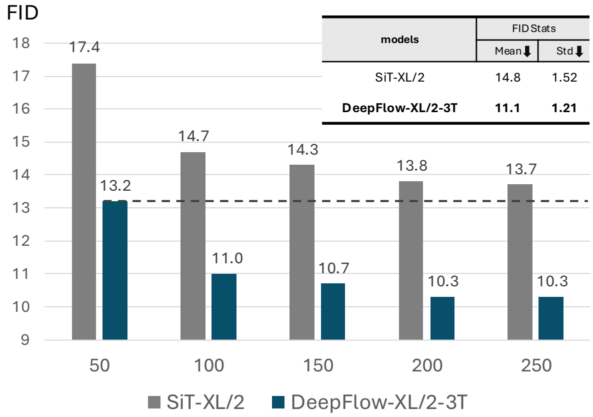
- **Inference (sampling):** In line with SiT [11] and REPA [16], we adopt an SDE sampling strategy and perform 250 steps to ensure a fair comparison. We also search for the optimal classifier-free guidance (CFG) scale during the evaluation of DeepFlow. As shown in Table 2, DeepFlow-XL/2-3T without SSL alignment achieves its best FID performance at a CFG scale of 1.325, whereas DeepFlow-XL/2-3T with SSL alignment reaches optimal performance at a CFG scale of 1.3.

A.4. Sensitivity to Different Number of Samplings

Figure 3 illustrates the performance sensitivity of our DeepFlow model to varying numbers of sampling steps and highlights its robustness compared to SiT [11]. Notably, DeepFlow maintains stable performance across sampling steps ranging from 50 to 250 (with a mean FID of 11.1 and a standard deviation of 1.21), suggesting that it is less sensitive to changes in the number of steps than SiT [11], which exhibits a higher mean FID of 14.8 and a standard deviation of 1.52. Furthermore, DeepFlow surpasses SiT's performance at 250 steps even when using only 50 steps. These results underscore the efficiency of DeepFlow: it not only reduces computational cost by requiring fewer steps, but it also delivers superior overall performance.

A.5. More Detailed Results on GenEval Benchmark

Table 1 presents a zero-shot text-to-image generation comparison between DeepFlow-2/3T and SiT [11], both using 24 transformer layers on GenEval benchmark [4]. Overall, DeepFlow outperforms SiT across most categories, including Single Object, Two Object, and Counting, indicating better handling of object complexity and quantity. DeepFlow also achieves higher scores in color-related tasks (Colors and Color Attributes) and positioning, demonstrating more accurate object placement and color fidelity. Moreover, incorporating SSL alignment (e.g., DINOv2) benefits both models but consistently maintains DeepFlow 's performance advantage.



models	FID Stats	
	Mean⬇	Std⬇
SiT-XL/2	14.8	1.52
DeepFlow-XL/2-3T	**11.1**	**1.21**

Figure 3. **Ablation Study on The Number of Sampling Steps.** We provide additional analysis on performance sensitivity of our DeepFlow-XL/2-3T and SiT-XL/2 to different number of sampling steps including 250, 100, 150, 100, 50 SDE steps.

A.6. Additional Qualitative Results

In this section, we provide an extensive qualitative analysis guided by the following criterion: (i) *Supplementary to Figure 4 in the main paper: Can DeepFlow generate high-quality samples even when trained for substantially fewer epochs?* This question is addressed in Figures 4 and 5, which visualize samples generated by SiT-XL/2 [11] and DeepFlow-XL/2-3T trained across varying epochs. We observe that DeepFlow-XL/2-3T not only yields highly promising results at just 80 epochs but also demonstrates stable convergence in subsequent epochs. (ii) *Can DeepFlow further enhance its generation capability by leveraging classifier-free guidance (CFG) [6]?* We demonstrate the visual effectiveness of

DeepFlow-XL/2-3T with CFG by sampling 256×256 images at a CFG scale of 4.0, as illustrated in Figures 6 to 8. Moreover, we show that the generative performance can be further improved by integrating SSL alignment [16], as shown in Figures 9 to 11. Finally, DeepFlow-XL/2-3T successfully synthesizes high-resolution images (512×512) of superior quality, as demonstrated in Figures 12 to 15. (iii) *Can DeepFlow achieve superior text-to-image generation quality compared to SiT [11]?* Figure 16 visually compares samples generated by MMDiT [3] trained with the SiT objective against those produced by DeepFlow, using identical text prompts. Notably, DeepFlow generates more realistic images that also exhibit higher fidelity to the provided textual descriptions.

A.7. Datasets and Metrics

The datasets we used for training and evaluating DeepFlow are described as follows:

ImageNet-1K: We train and evaluate DeepFlow on ImageNet-1K dataset for class-conditional generation benchmark. This dataset spans 1000 object classes and contains 1,281,167 training images, 50,000 validation images and 100,000 test images. The generation results are evaluated with generation FID using pre-computed statistics and scripts from ADM [1].

License: https://image-net.org/accessagreement

URL: https://www.image-net.org/

MS-COCO: We train and evaluate DeepFlow on MS-COCO dataset for text-to-image generation benchmark. This dataset contains 82,783 images for training, 40,504 images for validation. The generation results are evaluated with generation FID and $\text{FD}_{DINO_{v2}}$ [15].

License: https://cocodataset.org/termsofuse

URL: https://cocodataset.org

GenEval: Baselines and DeepFlow trained on MS-COCO for text-to-image generation are further evaluated on GenEval dataset [4]. It consists of 553 prompts with four images generated per prompt. Generated samples are evaluated according to various criteria (e.g., Single object, Two object, Counting, Colors, Position, Color attribute).

FID *vs.* $\text{FD}_{DINO_{v2}}$ We carefully select evaluation metrics tailored to each benchmark. For the ImageNet benchmark, we use the FID score because the inception model employed for FID was pre-trained on ImageNet, making it a suitable measure for this dataset. Conversely, for the MS-COCO benchmark, which has a distribution different from ImageNet, we also report $\text{FD}_{DINO_{v2}}$ [15]. This metric leverages a DINO_{v2} model pretrained on a more diverse dataset, ensuring a more appropriate evaluation for MS-COCO dataset.

A.8. Discussion & Limitations

While the proposed DeepFlow demonstrates impressive performance and training efficiency in image generation tasks, there remains ample scope for further optimization in future work. First, although our text-to-image results are promising compared to previous flow-based models under fair settings, DeepFlow still underperforms state-of-the-art models (*e.g.*, [7, 3, 13]). Training DeepFlow on large-scale datasets could be a fruitful direction to improve its performance. Second, exploring deeper theoretical insights into DeepFlow would provide a more thorough validation of our approach. We anticipate that our DeepFlow will serve as a general framework for flow-based generative model with this further improvement.

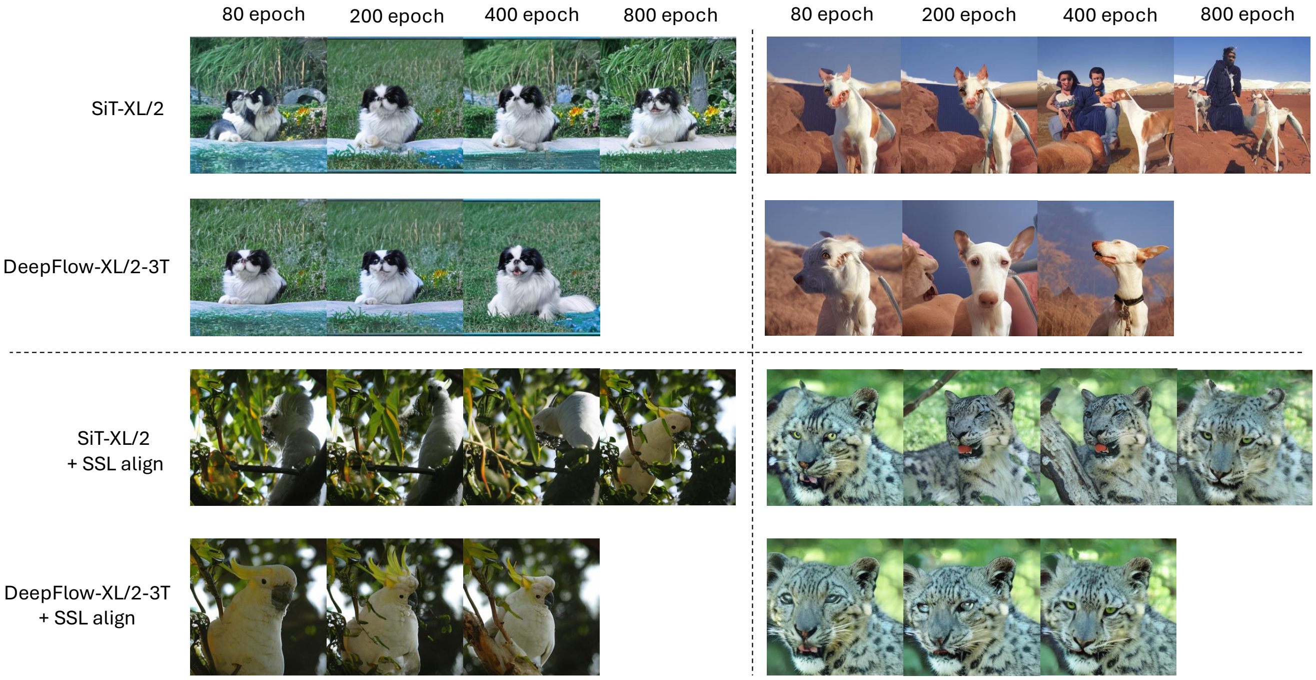


Figure 4. **Qualitative Comparisons with Baseline in Different Epochs (1).**

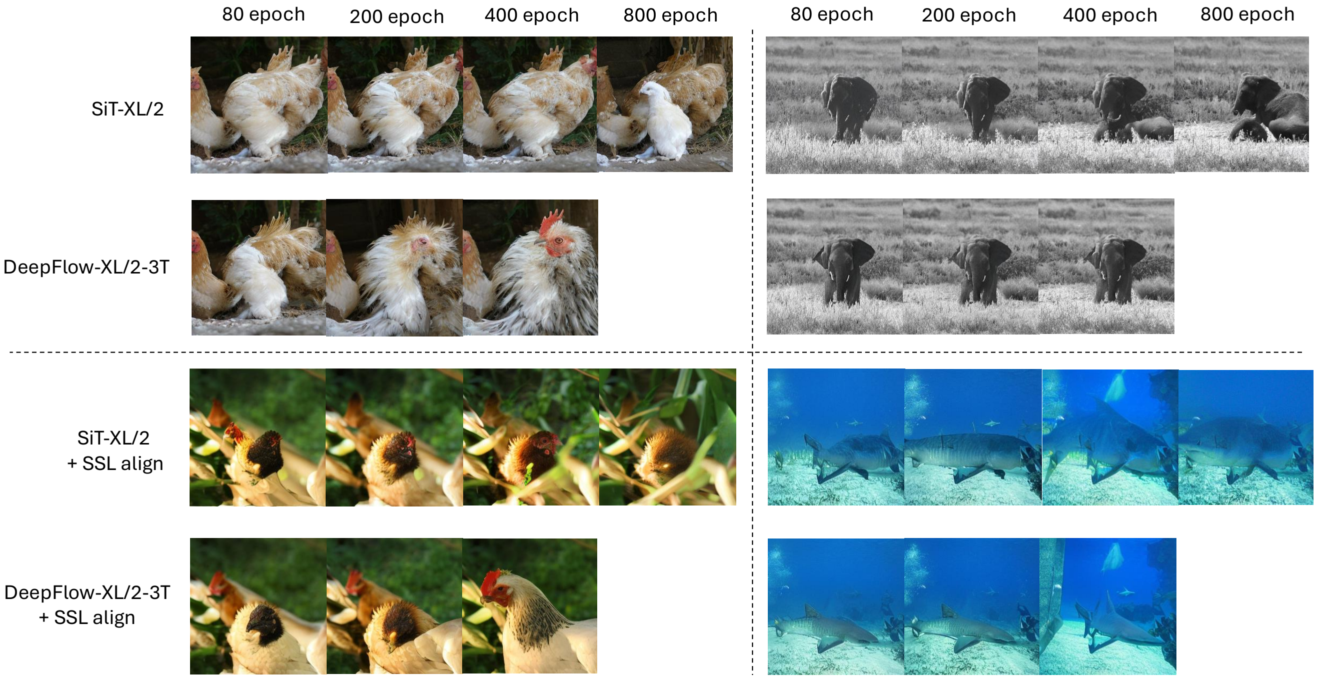


Figure 5. **Qualitative Comparisons with Baseline in Different Epochs (2).**

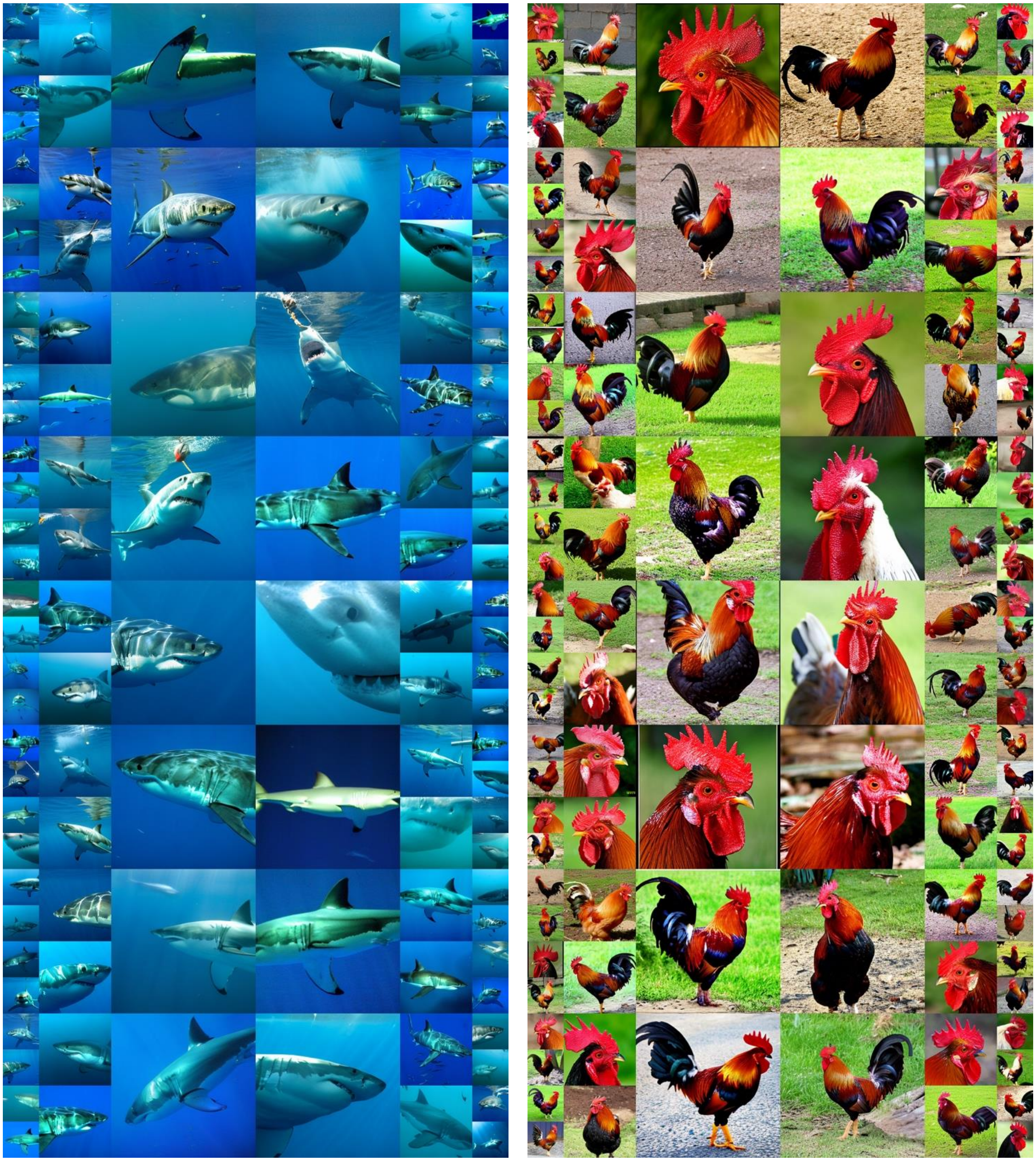


Figure 6. **Uncurated** 256×256 **DeepFlow-XL/2-3T Samples (1).** Classifier-free guidance scale = 4.0.
(Left): Class = "white shark" (2)
(Right): Class = "cock" (7)

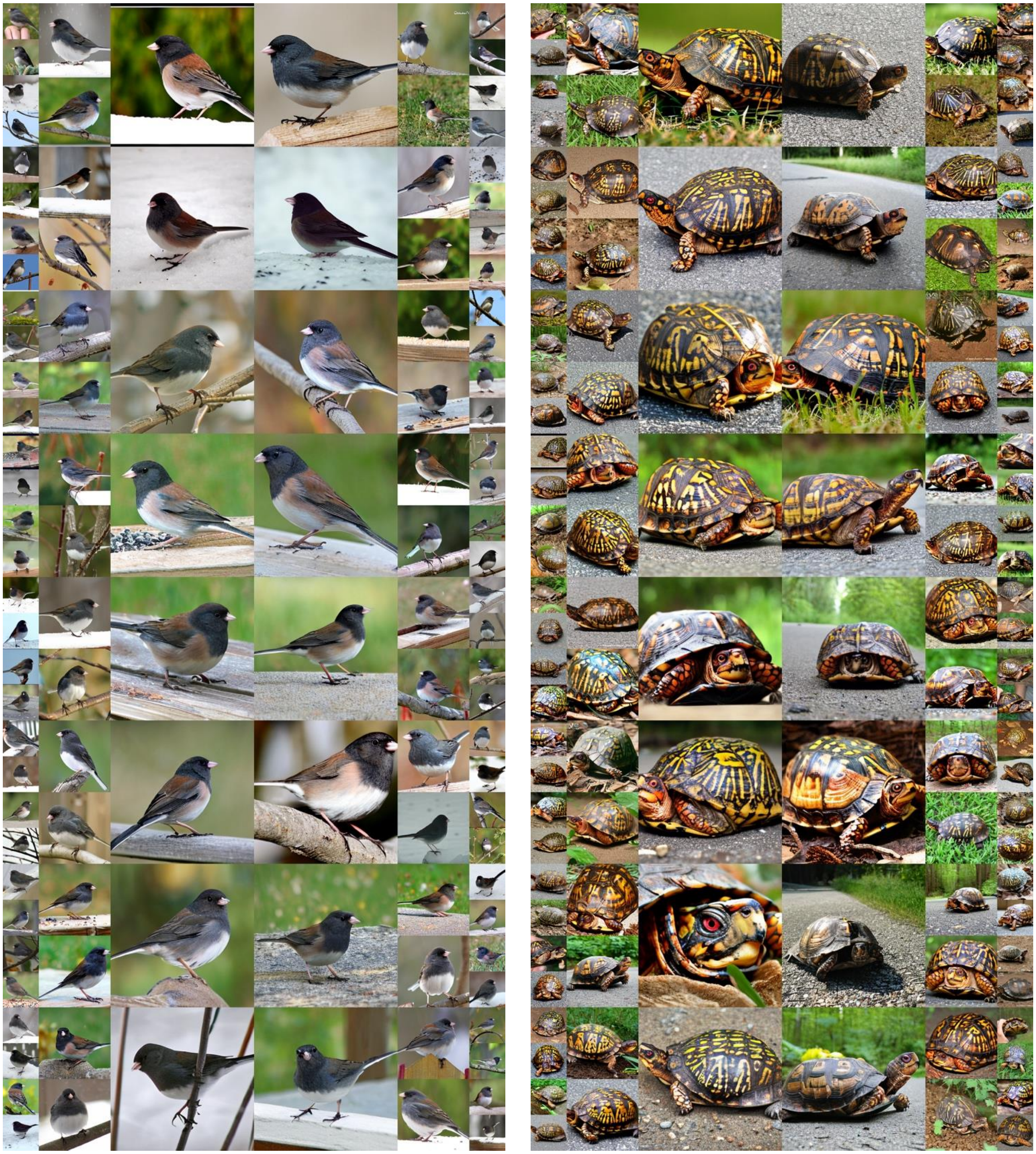


Figure 7. **Uncurated** 256×256 **DeepFlow-XL/2-3T Samples (2).** Classifier-free guidance scale = 4.0.
(Left): Class = “snowbird” (13)
(Right): Class = “box turtle” (37)

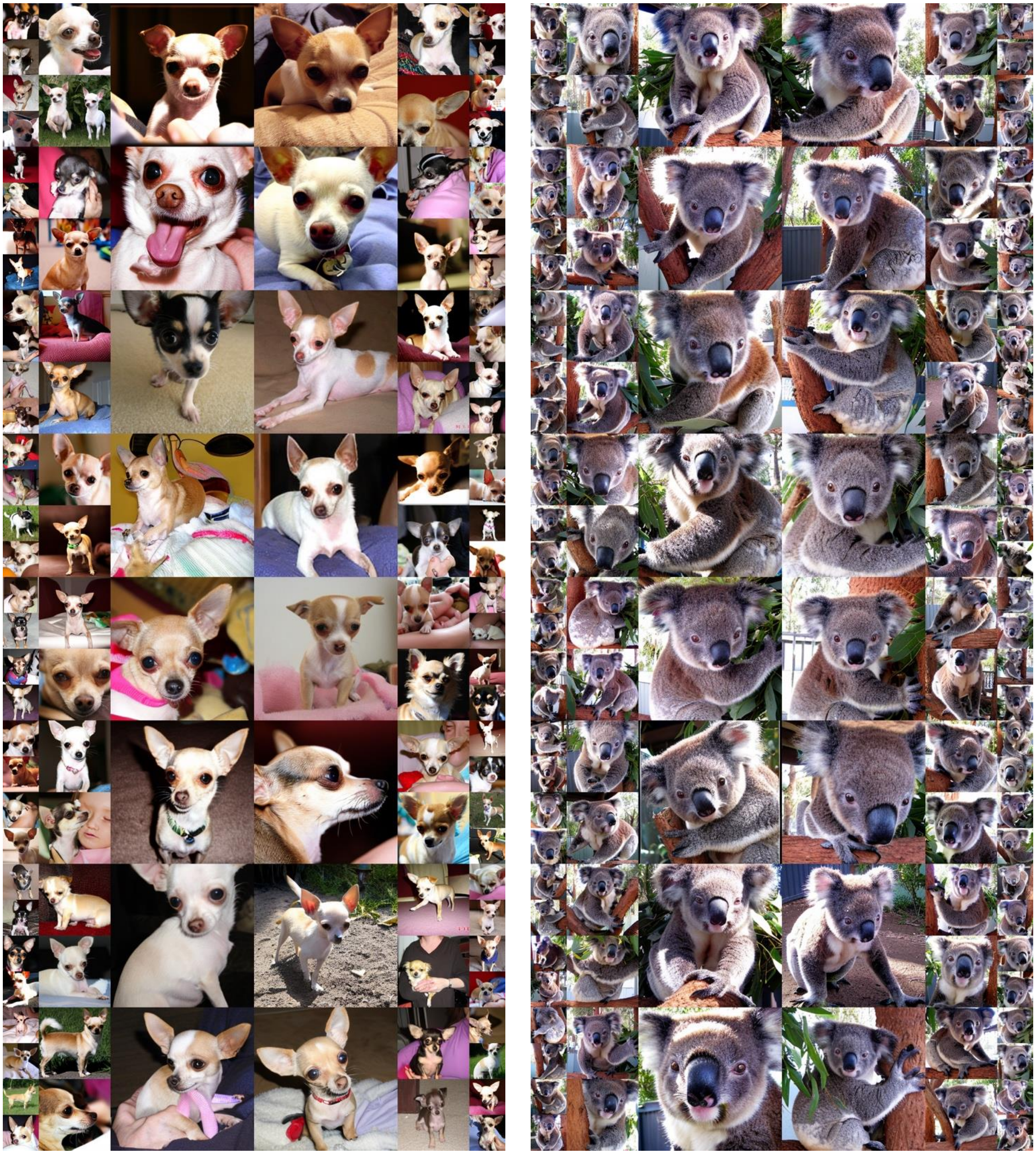


Figure 8. **Uncurated** 256×256 **DeepFlow-XL/2-3T Samples (3).** Classifier-free guidance scale = 4.0.
(Left): Class = “Chihuahua” (151)
(Right): Class = “koala” (105)

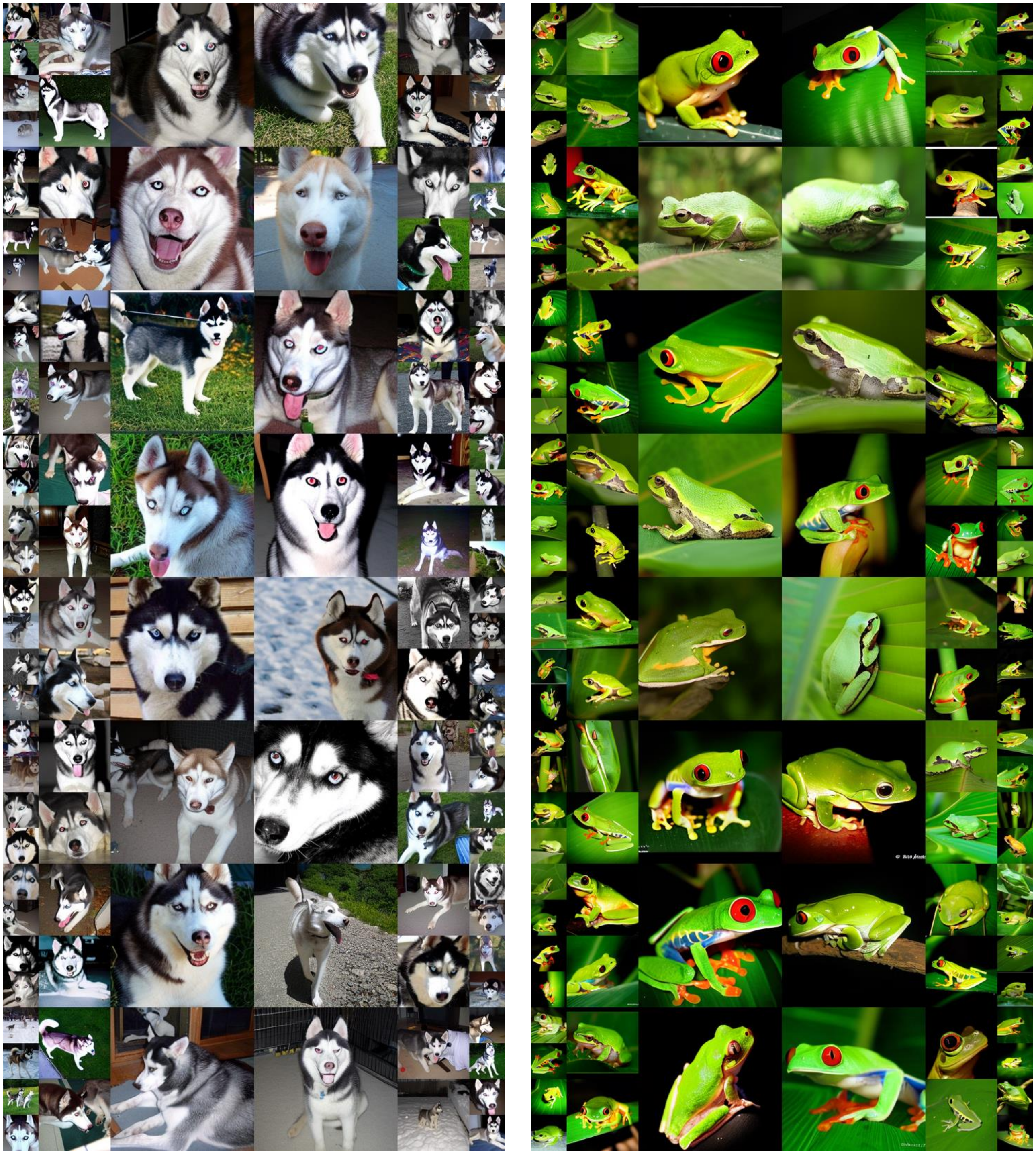


Figure 9. **Uncurated** 256×256 **DeepFlow-XL/2-3T+SSL align** [16] **Samples (1).** Classifier-free guidance scale = 4.0.
(Left): Class = “Siberian husky” (250)
(Right): Class = “tree frog” (31)

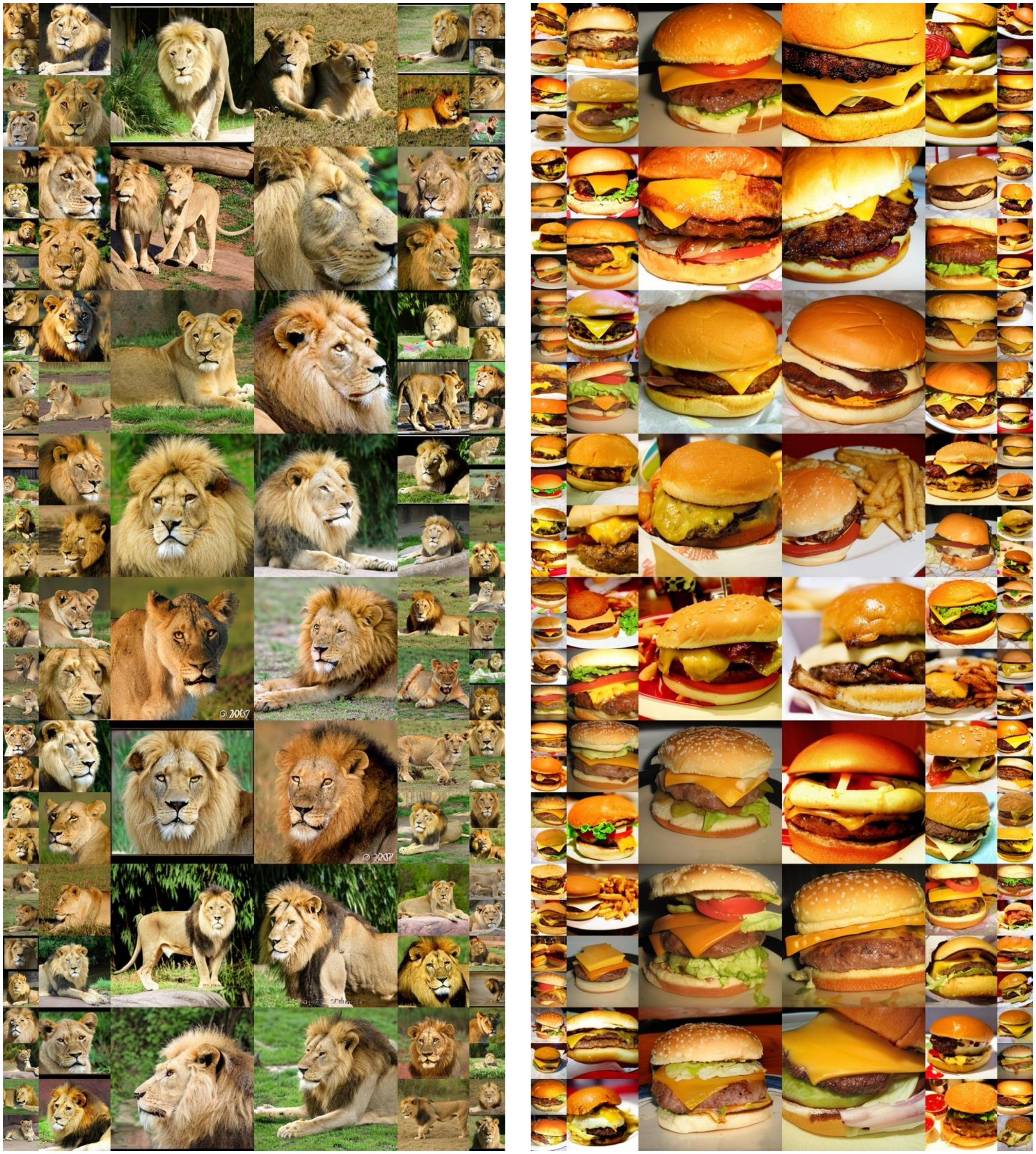


Figure 10. **Uncurated** 256×256 **DeepFlow-XL/2-3T+SSL align** [16] **Samples (2).** Classifier-free guidance scale = 4.0.
(Left): Class = “lion” (291)
(Right): Class = “cheeseburger” (933)

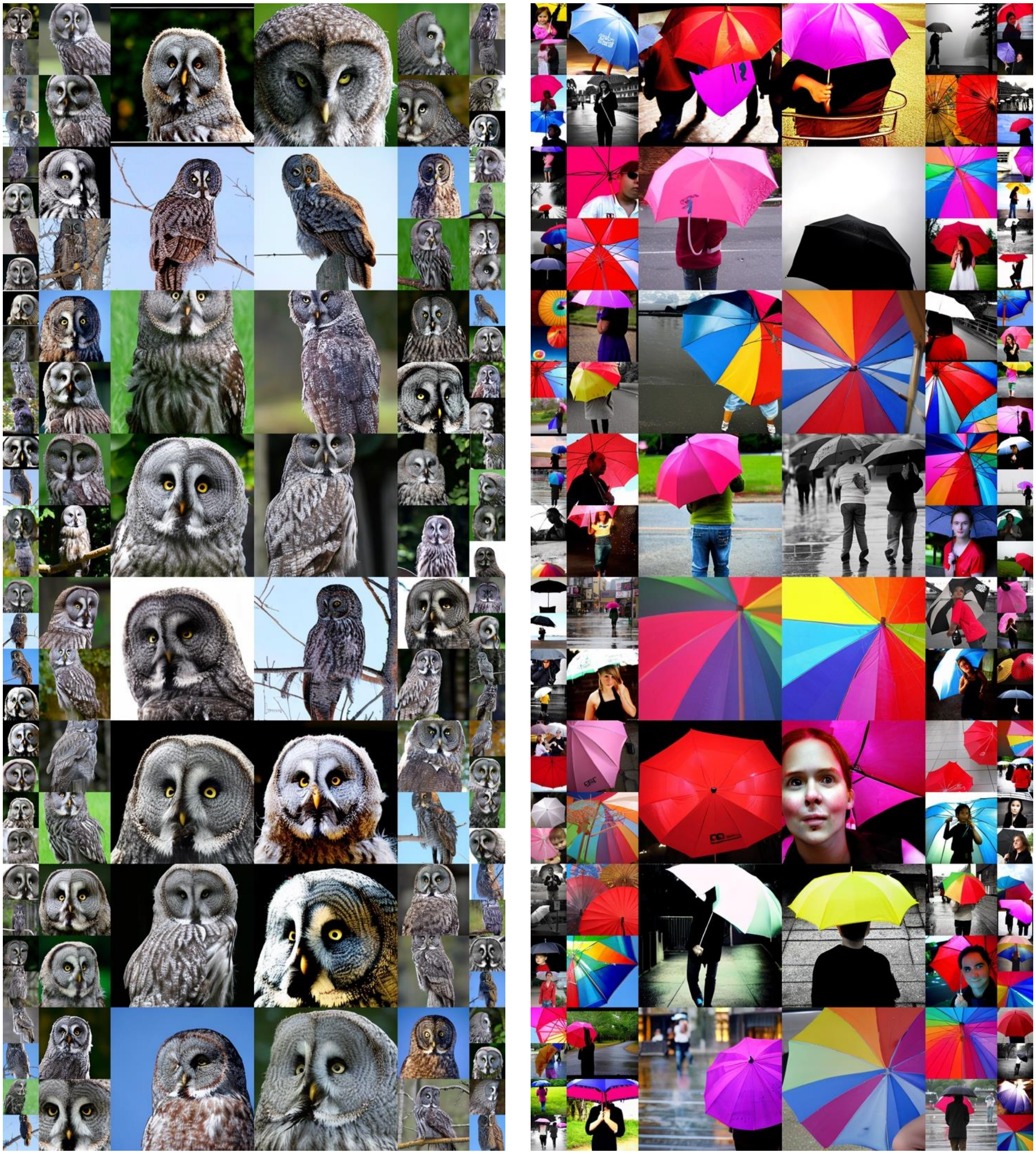


Figure 11. **Uncurated** 256×256 **DeepFlow-XL/2-3T+SSL align** [16] **Samples (3).** Classifier-free guidance scale = 4.0.
(Left): Class = "great grey owl" (24)
(Right): Class = "umbrella" (879)

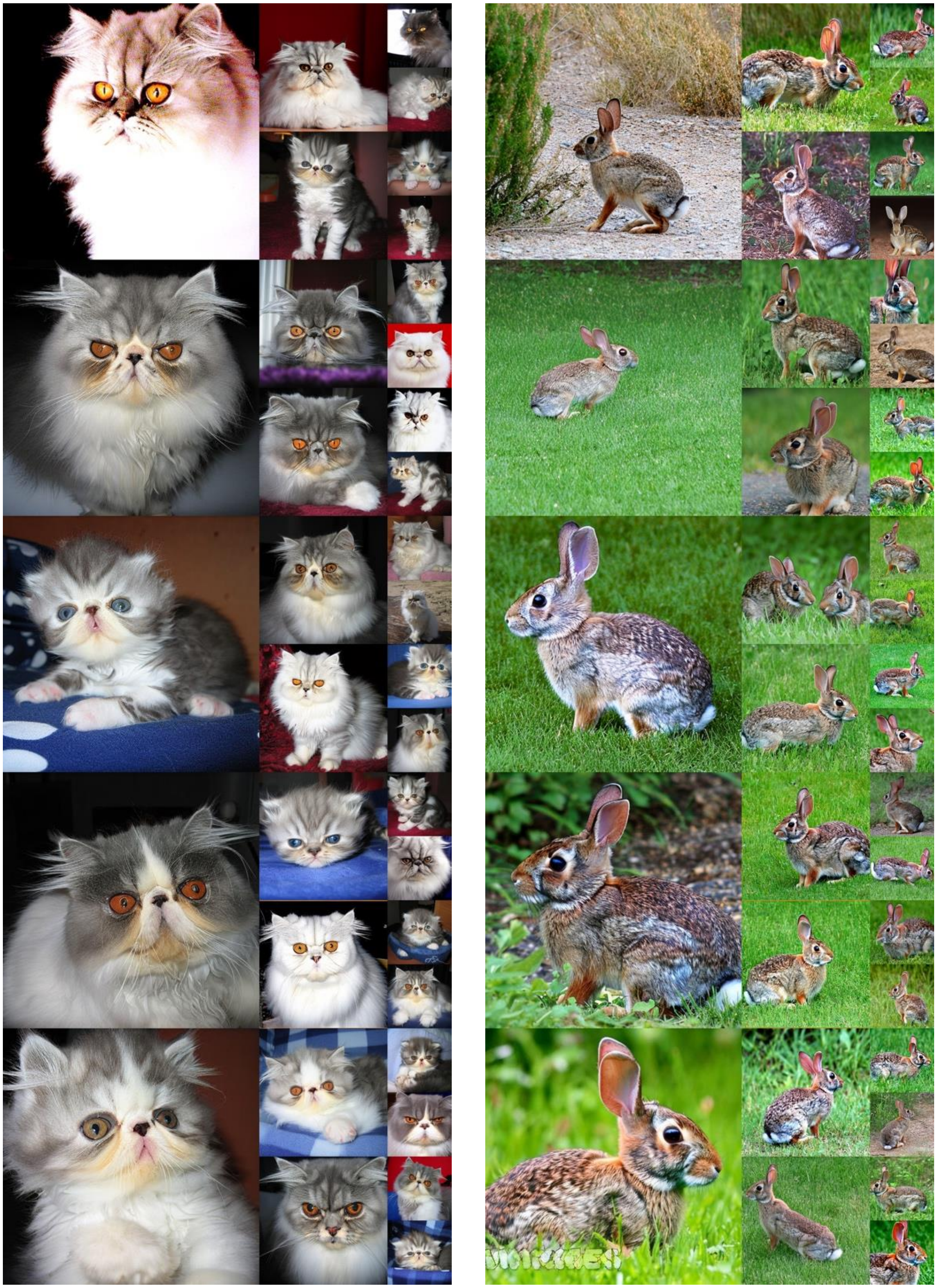


Figure 12. **Uncurated** 512×512 **DeepFlow-XL/2-3T Samples (1).** Classifier-free guidance scale = 4.0.
(Left): Class = "Persian cat" (283)
(Right): Class = "wood rabbit" (330)

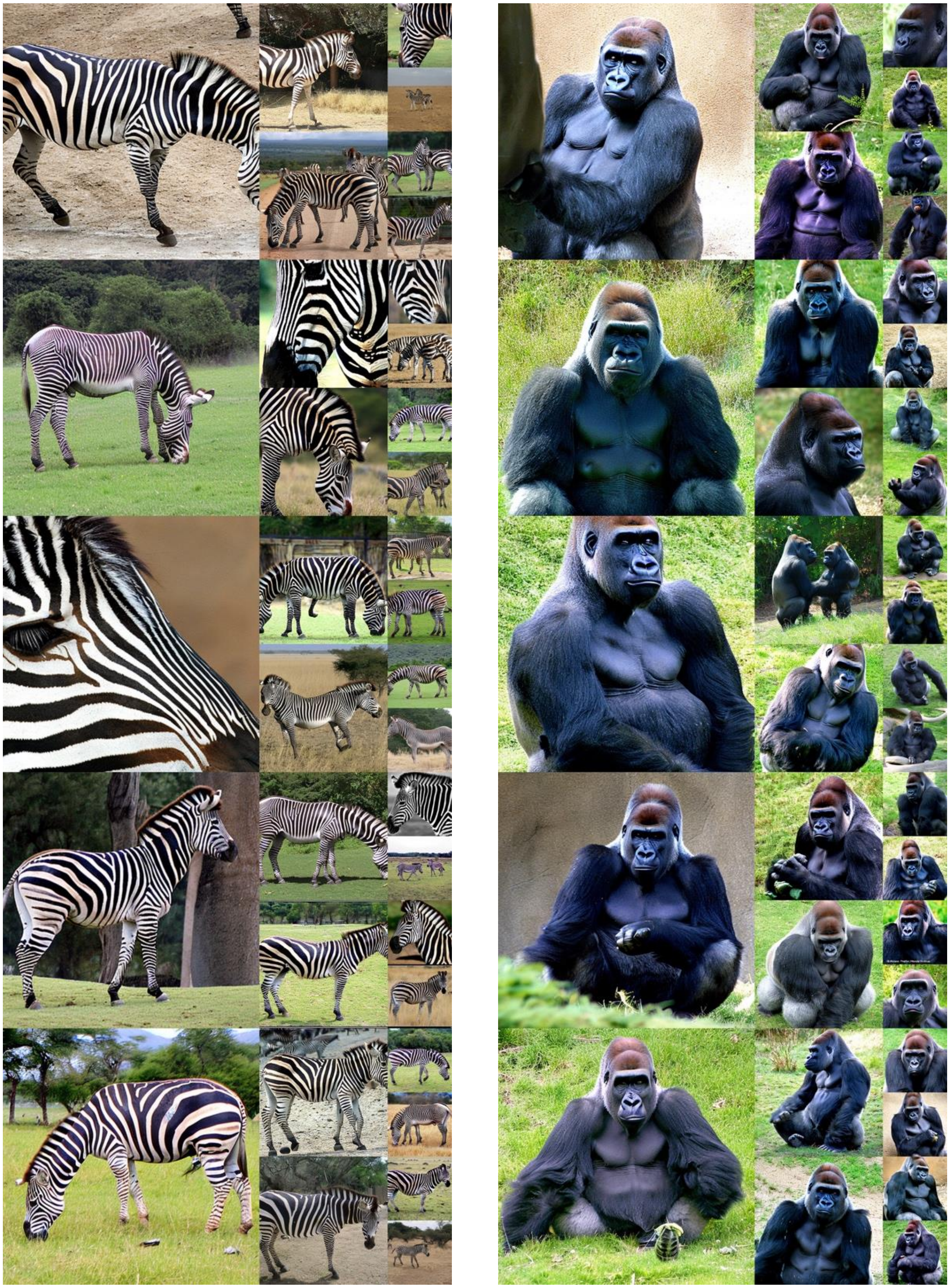


Figure 13. **Uncurated** 512×512 **DeepFlow-XL/2-3T Samples (2).** Classifier-free guidance scale = 4.0.
(Left): Class = “zebra” (340)
(Right): Class = “gorilla” (366)

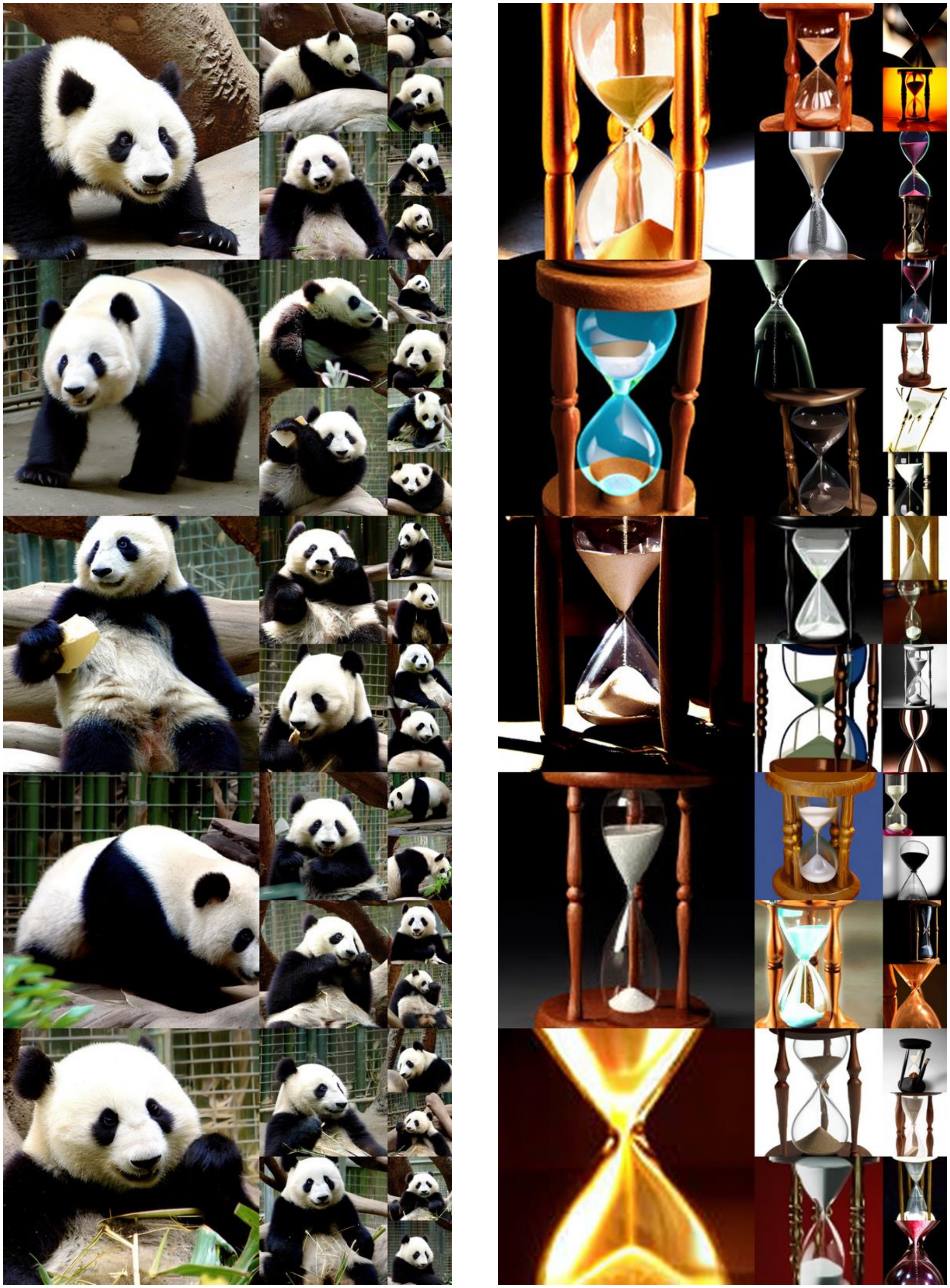


Figure 14. **Uncurated** 512×512 **DeepFlow-XL/2-3T+SSL align** [16] **Samples (1).** Classifier-free guidance scale = 4.0.
(Left): Class = “giant panda” (388)
(Right): Class = “hourglass” (604)

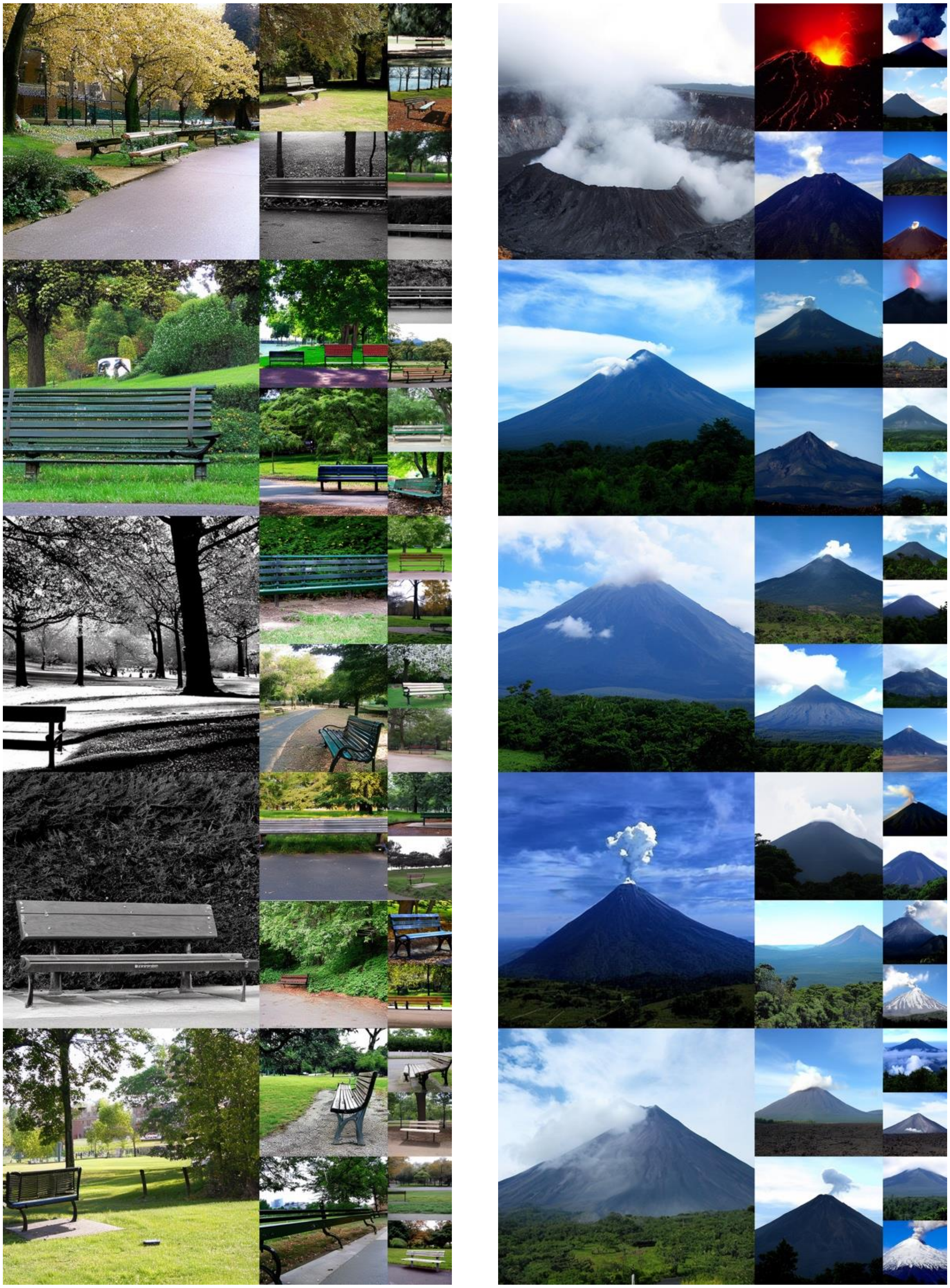


Figure 15. **Uncurated** 512×512 **DeepFlow-XL/2-3T+SSL align** [16] **Samples (2).** Classifier-free guidance scale = 4.0.
(Left): Class = “park bench” (703)
(Right): Class = “volcano” (980)

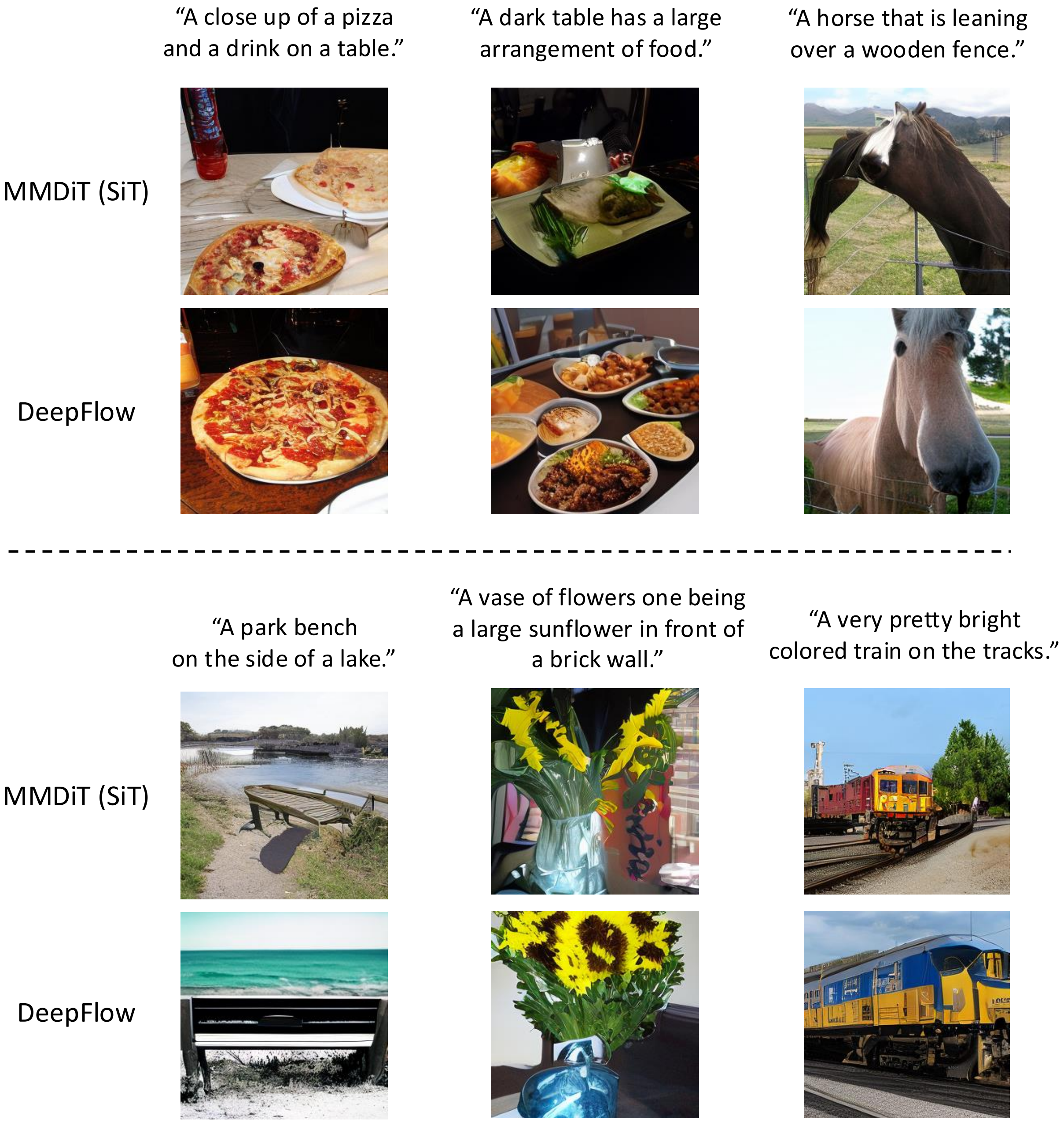

"A close up of a pizza and a drink on a table."
"A dark table has a large arrangement of food."
"A horse that is leaning over a wooden fence."
MMDiT (SiT)
DeepFlow
"A park bench on the side of a lake."
"A vase of flowers one being a large sunflower in front of a brick wall."
"A very pretty bright colored train on the tracks."
MMDiT (SiT)
DeepFlow

Figure 16. **Text-to-Image Generation Results.**

References

[1] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2021. 4

[2] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Journal of Neural Networks (NN)*, 2018. 2

[3] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Proc. of Int'l Conf. on Machine Learning (ICML)*, 2024. 4

[4] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2024. 3, 4

[5] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022. 2

[6] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[7] Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. *arXiv preprint arXiv:2501.07730*, 2025. 4

[8] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2014. 3

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2019. 2

[11] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2024. 1, 2, 3, 4

[12] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2023. 2

[13] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025. 4

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[15] George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2023. 4

[16] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2025. 2, 3, 4, 9, 10, 11, 14, 15