
Appendix of Improving Progressive Generation with Decomposable Flow Matching

Anonymous Author(s)

Affiliation

Address

email

1 Contents

2	A Framework Details	1
3	A.1 DFM Implementation Details	1
4	A.2 DFM Training Details	2
5	A.3 DFM Dataset Details	2
6	A.4 Inference Details	2
7	A.5 Choosing Training and Inference Hyperparameter	2
8	B Baseline Details.	3
9	C Additional Evaluation Results and Details	3
10	D Autoencoders without Scale Equivariance	4
11	E Large Scale Finetuning	4
12	E.1 FLUX DFM Finetuning	5
13	F Failed Experiments	5
14	G Limitations	6

15 A Framework Details

16 A.1 DFM Implementation Details

17 We base our architecture on a modified version of DiT [7]. Specifically, we normalize the data across
18 scales by applying scale-wise pre- and postconditioning following the scheme of [4], which ensures
19 that each input and output distribution has unit variance in expectation. Additionally, we replace ViT
20 frequency-based positional embeddings applied in DiT [7] with 2D rotary positional embeddings
21 (RoPE) [10] due to its wide adaptation in recent models [5, 11]. Furthermore, we condition the model
22 on the stage being generated by adding an embedding of the current stage index to the modulation
23 signal. Finally, we drop the class label conditioning 10% of the time to enable classifier-free guidance.

To adapt the DiT [7] for video generation, we patchify the latent frames independently and apply tokenwise concatenation of the resulting tokens [11]. Furthermore, we replace the 2D RoPE with 3D RoPE to adapt the position information to the video input. Finally, our ablations on Kinetics-700 reveal that unlike the FLUX autoencoder (See Table. 1 (c)), Cogvideo latents benefit from standardizing the input to have unit variance before the application of the diffusion process. Therefore, we apply such standardization to all of our video experiments.

A.2 DFM Training Details

We train using the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$) and a base learning rate of 0.0001 with 10k linear warmup steps, weight decay of 0.01, and total batchsize of 256. The main experiments are trained with DiT-XL/2 on ImageNet-1K for 512px and 1024px, respectively, for 500k and 350k steps. For the main video experiments on Kinetics-700, we train for 200k steps. Ablations are trained with DiT-B/2 for 600k steps using the same hyperparameters as the main experiments.

ImageNet-1K 512px experiments are trained on a single node containing 8 H100 GPUs, whereas ImageNet-1K 1024px and Kinetics-700 512px experiments used 2 nodes of the same type. Ablations were trained on a single node.

A.3 DFM Dataset Details

Our main image experiments are trained and evaluated on ImageNet-1K [2] which has a research-only, non-commercial license. Our video experiments are trained and evaluated on Kinetics-700 [1], which is available under the Creative Commons Attribution 4.0 (CC BY 4.0) license.

A.4 Inference Details

For all of our experiments, unless otherwise specified, we use an Euler ODE sampler with 40 steps and a linear timestep scheduler.

A.5 Choosing Training and Inference Hyperparameter

DFM introduces several training and inference hyperparameters. In practice, a default configuration of such hyperparameters generalizes well across different datasets and settings. In the following, we discuss such optimal configurations and summarize the observed effects of varying the main hyperparameters.

- **Number of generation stages:** Using 2 stages works well for most experiments, where the first stage has a resolution of 256px, while the second stage has a resolution equal to the final resolution (see Table. 1). We find that a base resolution of 256px for the first stage contains an ideal amount of structural detail to support generation of the successive ones while not containing excessive amounts of fine-grained details.
- **First stage sampling probability p_t^0 :** We find that 0.9 generalizes well across different model sizes. The probability determines the amount of model capacity spent on structural detail modeling, so smaller models may benefit from higher values (see Table. 1)
- **Stage noise sampling parameters:** At training time, we use a logit normal distribution with parameter ($location=0$, $scale=1.0$) for the stage currently sampled for training, and ($location=1.5$, $scale=1.0$) for the previous stage. Larger location values for previous stages allow the model to leverage more structural details as conditioning during second stage generation, but expose the model more to train-inference mismatches if the first stage results are not generated with sufficient quality. Thus, larger location values are more beneficial for larger models (see Table. 1).
- **Sampling stage threshold τ :** 0.9 generalizes well across different settings (see Figure 1). As the parameter determines the amount of information in the first stage that the second stage can leverage as conditioning, in the presence of a high-quality first stage prediction, larger thresholds are desirable. While the optimal value varies depending on cfg, model, and dataset, we find the suggested value to be a reliable default.
- **Input standardization:** Before applying noise, we standardize each level in the decomposed input representation to have unit-variance for Kinetics-700 video experiments using the

CogVideo autoencoder, while this behavior is disabled for ImageNet-1K image generation experiments with the FLUX autoencoder. Standardization has an impact on loss magnitude for each stage, thus it acts similarly to the first stage sampling probability p_t^0 in balancing the amount of model capacity allocated to modeling structural details or fine details. We found that the optimal parameters can vary depending on the autoencoder representation and we suggest ablating over this setting when adopting a new autoencoder.

B Baseline Details.

This section, includes details about the baselines training and inference. First, the base architecture is described, then specific details about the Cascaded and Pyramidal Flow baselines are detailed.

Baselines architecture. All baselines use the hyperparameters and architecture of [7], with the modification to incorporate rotary positional embeddings (RoPE) [10] and Network Preconditioning [4]. Inference is performed with 40 steps using an Euler sampler and a linear sampling schedule.

Cascaded baseline. We initially train the stage 1 model following the spatial baseline for $\approx 80\%$ of the total training steps. To match the training and inference compute of the other baselines, we adopt a patch size of 1×1 , yielding the same number of tokens as the other baselines. Subsequently, we finetune the stage 1 model to obtain the stage 2 model, which upsamples the first stage output. For the stage 2 model, we introduce a dedicated pacification layer for the conditioning input from stage 1 and concatenate its output tokens with those in stage 2. We add a small amount of noise to the conditioning input during training to reduce exposure bias. The amount of noise is sampled from a logit normal distribution with scale parameters of 1.0 and location of 1.5. For inference, we perform a grid search over the best inference parameters and found that equally dividing the inference steps between stage 1 and stage 2 (*i. e.* 20 inference steps for each stage) and using a noise level of 0.025 yields the best results. Therefore, we use these settings for our evaluations.

Pyramidal Flow baseline. We follow Jin *et al* [3] in training the Pyramidal Flow baseline. For the three-stages experiments, we allocate twice the batch size for the second stage compared with the first and second stage following [3] and use a gamma parameter of 0.33. In the two-stages experiments, we allocate an equal batch size between the first and second stages. During inference, we use an equal number of inference stapes for each stage.

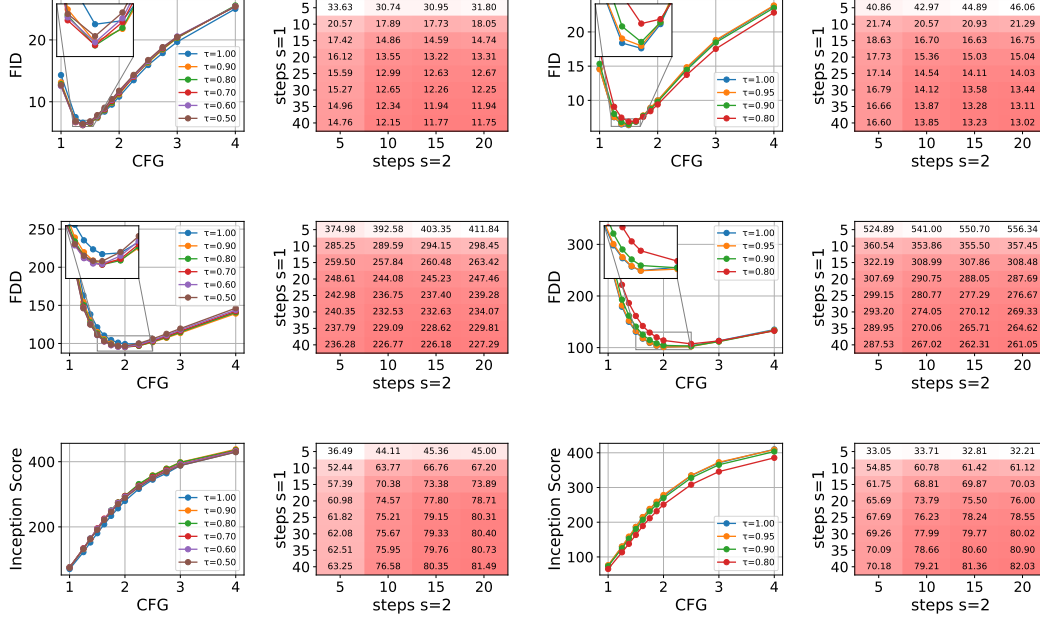
C Additional Evaluation Results and Details

Sampling parameters ablation We report quantitative results ablating the role of different sampling hyperparameters in Figure 1 and show qualitative results in Figure 13 and Figure 14 ablating, respectively, sampling steps allocated to each stage, and different cfg and sampling threshold configurations. Increasing the number of steps allocated to the first stage results in improved image structure, while increasing the number of steps allocated to the second stage results in reduced gains which are observable in areas with complex textures such as foliage, grass, or animal fur. The sampling threshold has a reduced impact on generation quality and is most visible at low cfg values.

Comparison to baselines Convergence behavior for DFM and baselines is shown in Figure 3, Figure 4, and Figure 5, respectively, for ImageNet-1K [2] 512px, ImageNet-1K [2] 1024px, and Kinetics-700 [1]. In addition, we show qualitative comparison on non-curated samples for DFM against baselines on ImageNet-1K [2] 512px (see Figure 6), ImageNet-1K [2] 1024px (see Figure 7), and Kinetics-700 [1] (see Figure 8, Figure 9, and Figure 10). Additional qualitative results and videos are provided in the *Website*.

FLUX-DFM qualitative comparison with FLUX-FT. We provide additional qualitative comparison of FLUX finetuned with DFM (FLUX-DFM) against FLUX with standard finetuning applied (FLUX-FT) on 1024px text-to-image generation in Figure 11 and Figure 12.

Qualitative results. We provide qualitative results on ImageNet-1K [2] 512px of selected classes in Figure 15, Figure 16, and Figure 17. Additionally, we provide fully uncured samples in Figure 18, Figure 19, and Figure 20.



(a) ImageNet-1K [2] 512px

(b) ImageNet-1K [2] 1024px

Figure 1: FID_{10K}, FDD_{10K} and IS ablating sampling configuration, threshold, and per-stage steps on DiT-XL/2 trained with DFM on ImageNet-1K [2].

121 D Autoencoders without Scale Equivariance

122 DFM explicitly decouples low- and high-frequency content into two successive stages. The first
 123 stage diffuses coarse, low-frequency information. Its output then conditions the generation of high-
 124 frequency details in the second stage. Recent work shows that diffusion models follow this spectral
 125 autoregressive pattern implicitly [6, 8, 9] and proposed scale-equivariant autoencoders to improve
 126 this separation, making the produced latents easier to diffuse [9]. Although DFM is agnostic to the
 127 choice of autoencoder, we conduct our experiments with the autoencoders of [9] finetuned for scale
 128 equivariance to ensure desirable spectral properties both for baselines and DFM.

129 In this section, we explore the behavior of DFM on non-regularized autoencoders. Table 1 reports
 130 a comparison between Flow Matching and DFM on the original FLUX autoencoder and scale-
 131 equivariant FLUX autoencoder for a DiT-B architecture trained for 400k steps on ImageNet-1K [2].
 132 When the scale-equivariant FLUX autoencoder is replaced with the original variant, Flow Matching
 133 suffers a substantial drop in performance, whereas DFM preserves a similar performance, thanks to
 134 its explicit spectral decomposition. In both cases, DFM outperforms Flow Matching.

135 E Large Scale Finetuning

136 This section explores the usage of DFM for fine-tuning large-scale models. Since the stage 1 input
 137 resembles a low-resolution version of the original input, we aim to preserve the pretrained model’s
 138 behavior as much as possible. DFM adds a per-stage patchification layer, timestep embedding, and
 139 output head. Accordingly, we reuse the pretrained patchification layer and timestep embedding for
 140 stage 1, while zero-initialising the corresponding stage 2 modules. Before training, the framework
 141 therefore replicates the pretrained model’s low-resolution predictions. After training, all weights
 142 (including patchification layers) are adapted to generate consistent stage 1 and stage 2 outputs.
 143 To adjust the model patchification layers weight to stage 1, which uses a smaller patch size, we
 144 upsample stage 1 input by a factor of 2 before adding noise, halving the effective patch size. We then
 145 downsample the model’s output to match stage 1 resolution.

E.1 FLUX DFM Finetuning

We choose to fine-tune FLUX-DEV [5] due to its strong performance. To avoid biases introduced by cfg distillation, the distillation-guidance factor is fixed at 1.0 throughout training, and all fine-tuning is carried on an internal dataset. We follow [9] to regularize the the FLUX autoencoder, yielding a *scale-equivariant* autoencoder.

Then, we perform full-finetuning of FLUX for 24k steps to adjust it to the new autoencoder, producing FLUX-SE. During the initial 4k training steps, we freeze all layers except the patchfication layers. Then, we finetune FLUX-SE for 32k steps to obtain FLUX-DFM. As a baseline, we finetune FLUX-SE for the same 32k steps using standard full-finetuning. Finetuning uses 1024-px and 512-px images at variable aspect ratios. We use Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$) and a base learning rate of 0.00001 with 2k linear warmup steps, weight decay of 0.01, and total batch size of 192. We drop the text conditioning 10% of the time to enable classifier-free guidance.

Please note that since FLUX-DEV is distilled and post-trained on highly aesthetic images, its distribution differs from that of our internal data. Therefore, a direct comparison with the original FLUX-DEV would not be informative. Rather, we measure speed in learning the new training distribution with DFM compared with standard full-finetuning at an equal training cost.

Inference. We evaluate using a test split of 10k prompts from our internal dataset. We use 40 sampling steps and Euler ODE solver, cfg of 3.0, and a distillation guidance factor of 1.0 (equivalent to not applying cfg).

F Failed Experiments

DCT-Space DFM We apply DFM in the DCT space, where the input visual modality is represented in the frequency domain rather than in the spatial domain. We obtain the DCT decomposition by first dividing the visual input into 2D or 3D blocks of pixels of size 4 or 8 and applying DCT to each of them. Different stages are formulated as the progressive modeling of DCT components of increased frequency. We find, however, that DiT models provide reduced performance when working in the frequency rather than the spatial domain, a finding we speculate may originate from an increased difficulty of leveraging token similarities in attention operations. While experiments showed that DCT can be used as a way of producing the input decomposition, input to the DiT model should be converted to the spatial domain for optimal performance, and the Laplacian decomposition is preferred due to its simplicity.

Alternative parameter specialization methods As parameter specialization showed the capability of increasing the model’s performance (see Table. 1 (d)), we investigate alternative ways of specializing model parameters per each stage. As an alternative avenue, we investigate the recent Tokenformer [12] architecture due to its capacity to progressively integrate new parameters during training. In particular, we instantiate a shared set of parameters and a set of specialized parameters for each stage. The shared set of parameters is always active, while the specialized parameters are activated only when corresponding to the currently active stage. We find the Tokenformer [12] architecture to underperform the regular DiT design when given the same amount of computation, thus we abort this experiment. To reduce the number of parameters required for specialization, we wrap each linear layer in different LoRA wrappers, one for each stage, and activate the respective wrapper when the corresponding stage is selected at training or inference. LoRA showed improved results over the baseline when utilizing high LoRA ranks which resulted in similar parameter counts to full parameter specialization.

Expert Models We train separate or *expert* models for each stage rather than a joint model. We then evaluate performance by mixing expert models trained for different numbers of steps. We find that performance improves steadily with more training of the first stage and improves marginally with more training with longer training of the second stage model, supporting the intuition that modeling of structural detail has a larger importance to sample quality. To reduce the computational burden of training separate models for each stage, we explore finetuning the second stage model starting from the first stage model with positive results. Despite positive results, the idea is not explored further due to the increased complexity of maintaining separate models for each stage. We note that, in principle, it is possible to obtain expert models by finetuning a jointly trained model separately for each stage with full or LoRA finetuning. We leave the exploration of this possibility as future work.



Figure 2: (left) Selected samples highlighting failure cases generated by our framework trained on ImageNet-1K [2] 512px on DiT-XL for 1.3M iterations. When generating high-frequency details such as fine structures, vegetation or cluttered environments, DFM may produce artifacts. (right) Artifacts can be mitigated by increasing the number of sampling steps for the second stage.

Table 1: DFM with original (orig) and scale-equivariant (SE) FLUX autoencoder. Experiments are performed on DiT-B and trained on ImageNet-1K [2] 512px for 400k steps.

Method	FID _{50K} ↓	FDD _{50K} ↓	IS ↑
Flow Matching (FLUX-ae-orig)	54.50	855.5	21.4
DFM (FLUX-ae-orig)	34.00	630.1	34.2
Flow Matching (FLUX-ae-SE)	43.16	728.85	26.6
DFM (FLUX-ae-SE)	32.89	626.5	35.0

G Limitations

DFM improves modeling of visual inputs by decomposing them into different components. The framework’s hyperparameters control the amount of model capacity dedicated to modeling each of the components. As an example, a larger probability of sampling stage 0 (p_t^0) during training (see Table. 1 (a)) results in a larger emphasis on structural details. As shown in Figure 2, selected samples containing large amounts of high-frequency components such as vegetation, fur, thin structures, or cluttered environments may exhibit artifacts in such regions which manifest as a flattened appearance. Increasing the number of sampling steps for the second stage (see Figure 2 (right) and fig. 13) mitigates such artifacts. By acting on training sampling probabilities for each stage, and distribution of sampling steps between different stages, the framework allows for balance between structural and fine details modeling quality.

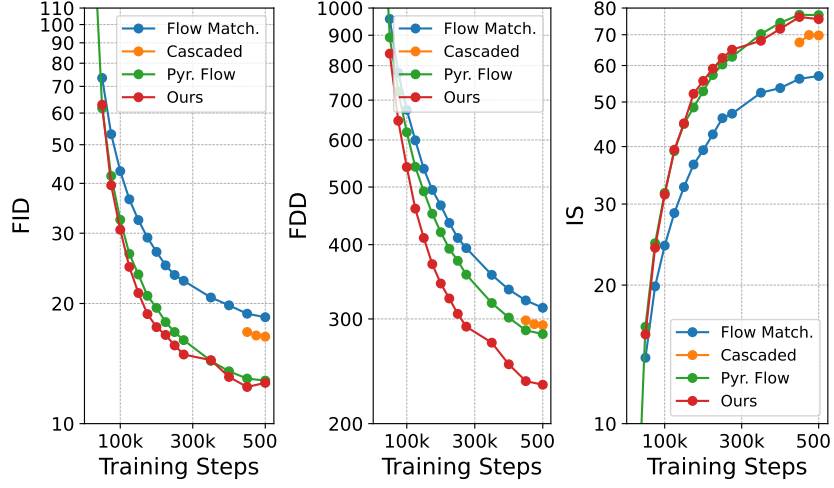


Figure 3: Convergence curves for different datasets and metrics comparing our framework to baselines on ImageNet-1K [2] 512px.

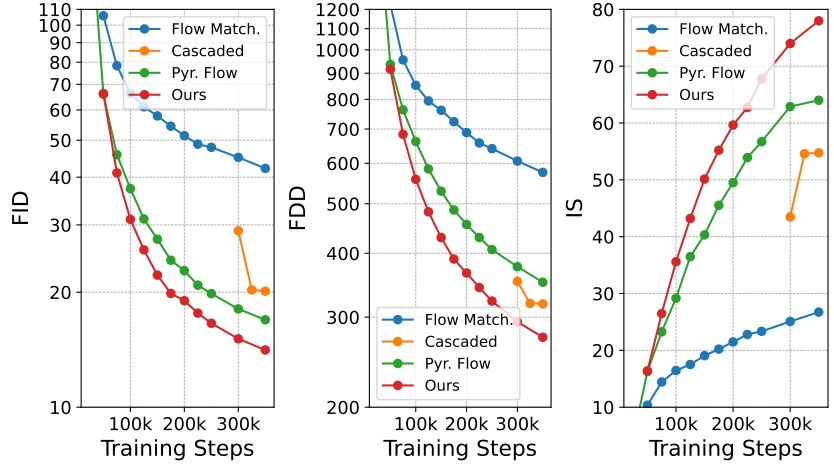


Figure 4: Convergence curves for different datasets and metrics comparing our framework to baselines on ImageNet-1K [2] 1024px.

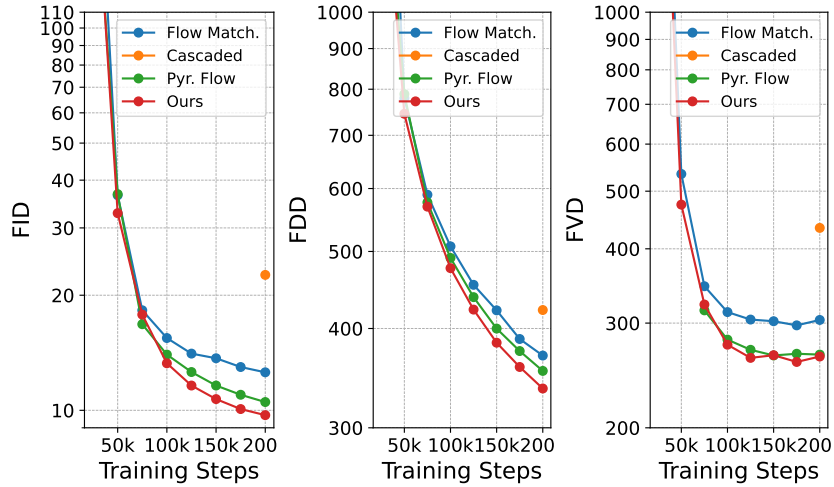


Figure 5: Convergence curves for different datasets and metrics comparing our framework to baselines on Kinetics-700 [1] 512px.



Figure 6: Comparison of DFM against baselines on DiT-XL trained on ImageNet-1K 512px [2] for 500k steps. Samples are fully uncensored and generated with cfg 3.0.



Figure 7: Comparison of DFM against baselines on DiT-XL trained on ImageNet-1K 1024px [2] for 350k steps. Samples are fully uncured and generated with cfg 3.0.

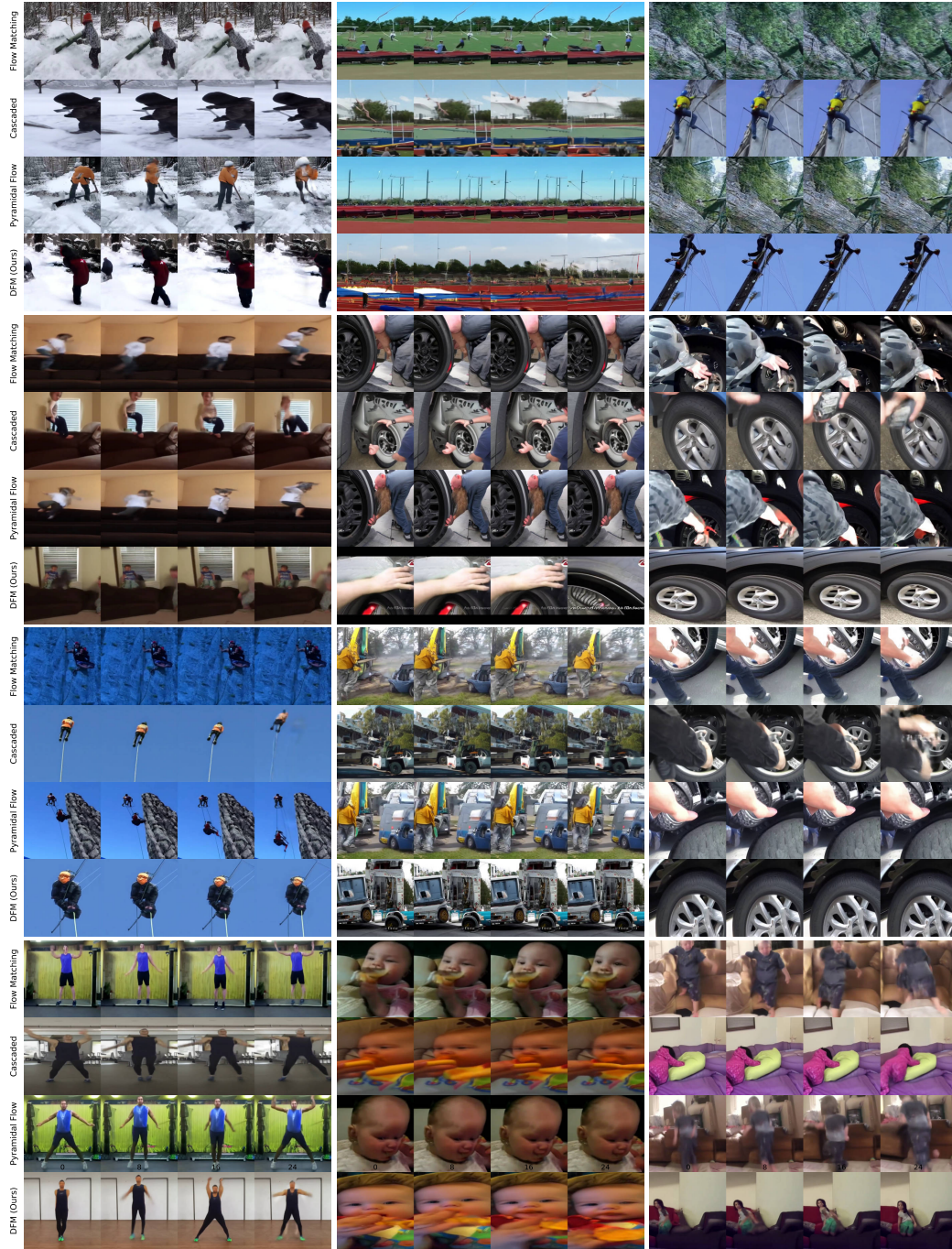


Figure 8: Comparison of DFM against baselines on DiT-XL trained on Kinetics-700 [1] 512px for 200k steps. Samples are fully uncensored and generated with cfg 3.0.

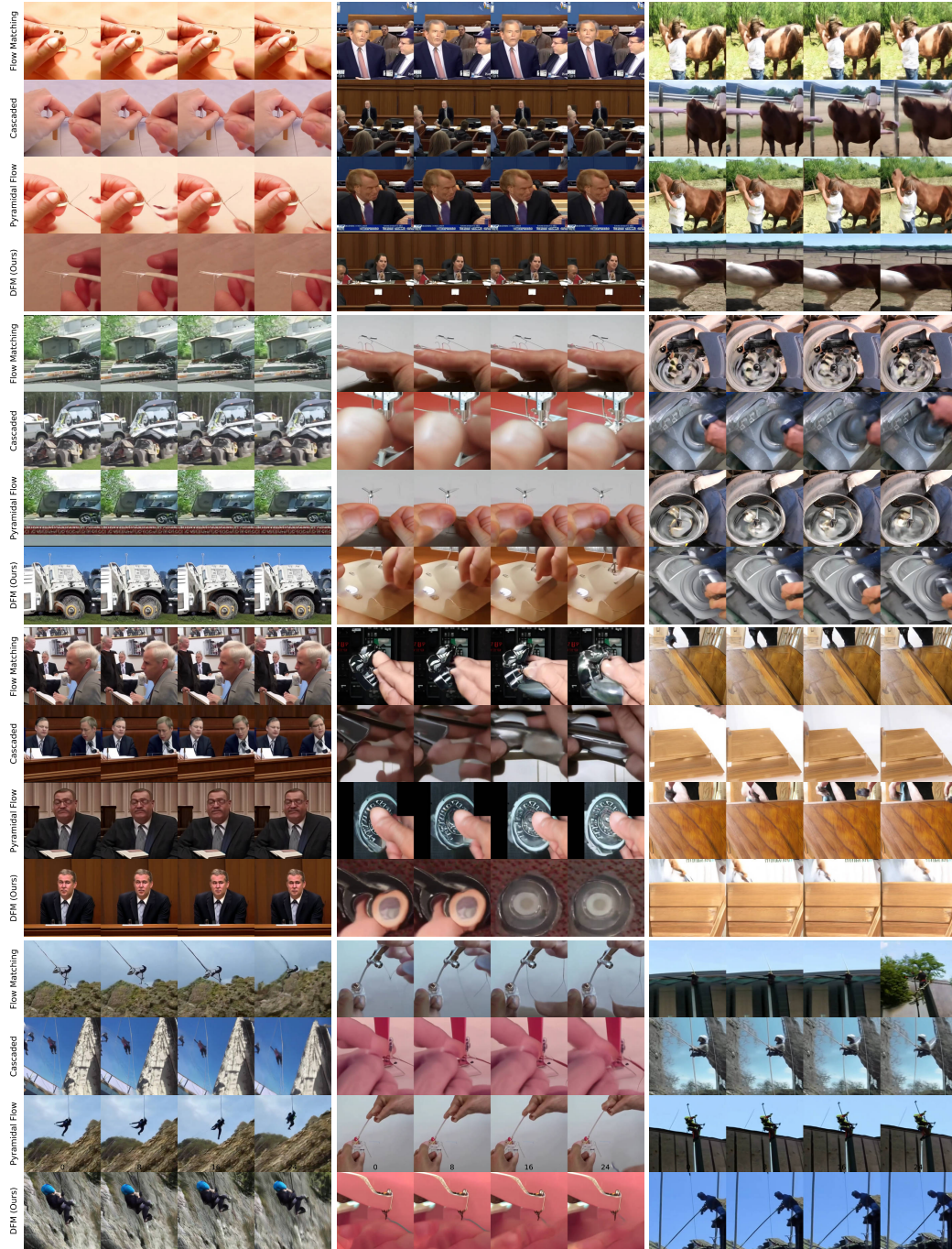


Figure 9: Comparison of DFM against baselines on DiT-XL trained on Kinetics-700 [1] 512px for 200k steps. Samples are fully uncured and generated with cfg 3.0.

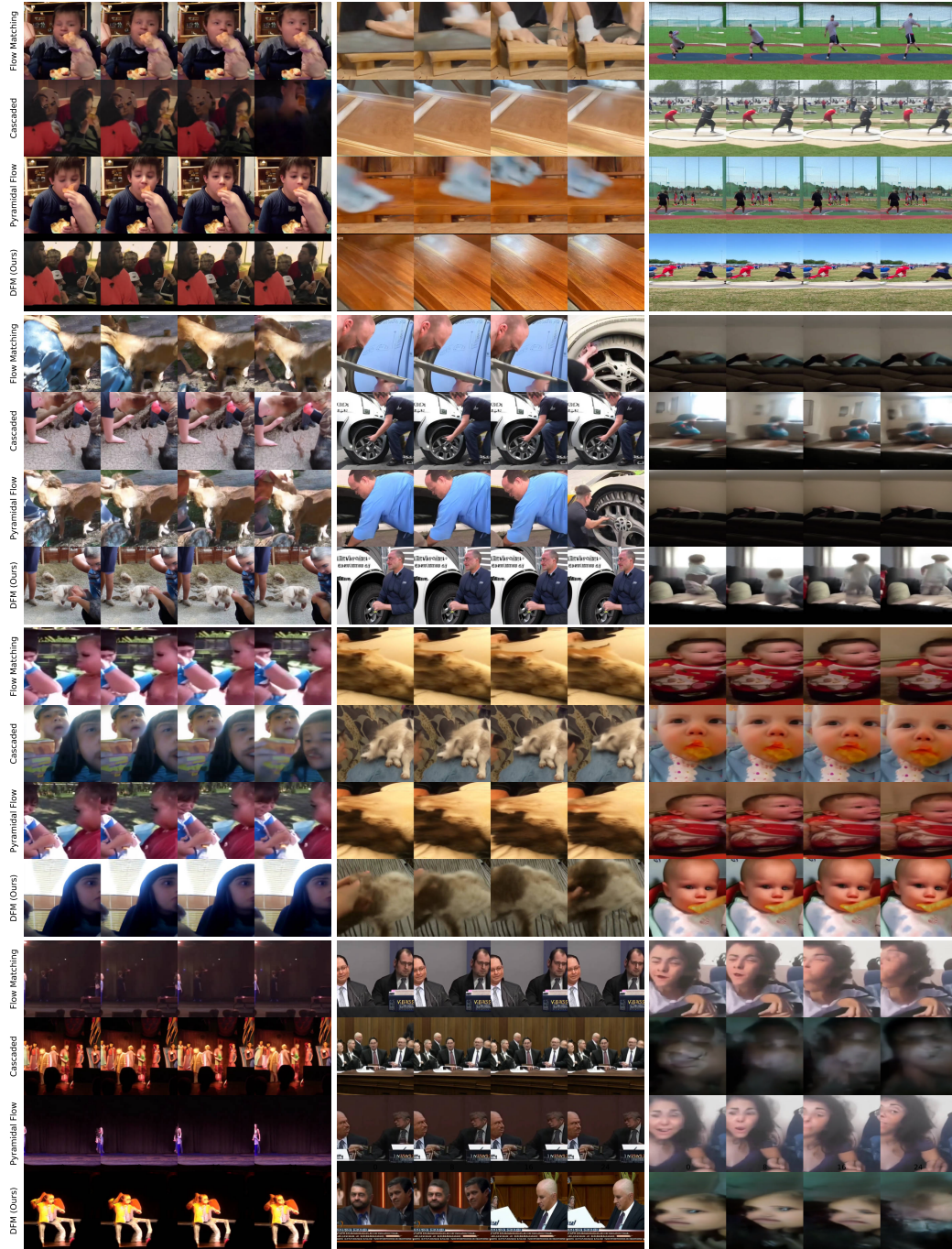


Figure 10: Comparison of DFM against baselines on DiT-XL trained on Kinetics-700 [1] 512px for 200k steps. Samples are fully uncensored and generated with cfg 3.0.

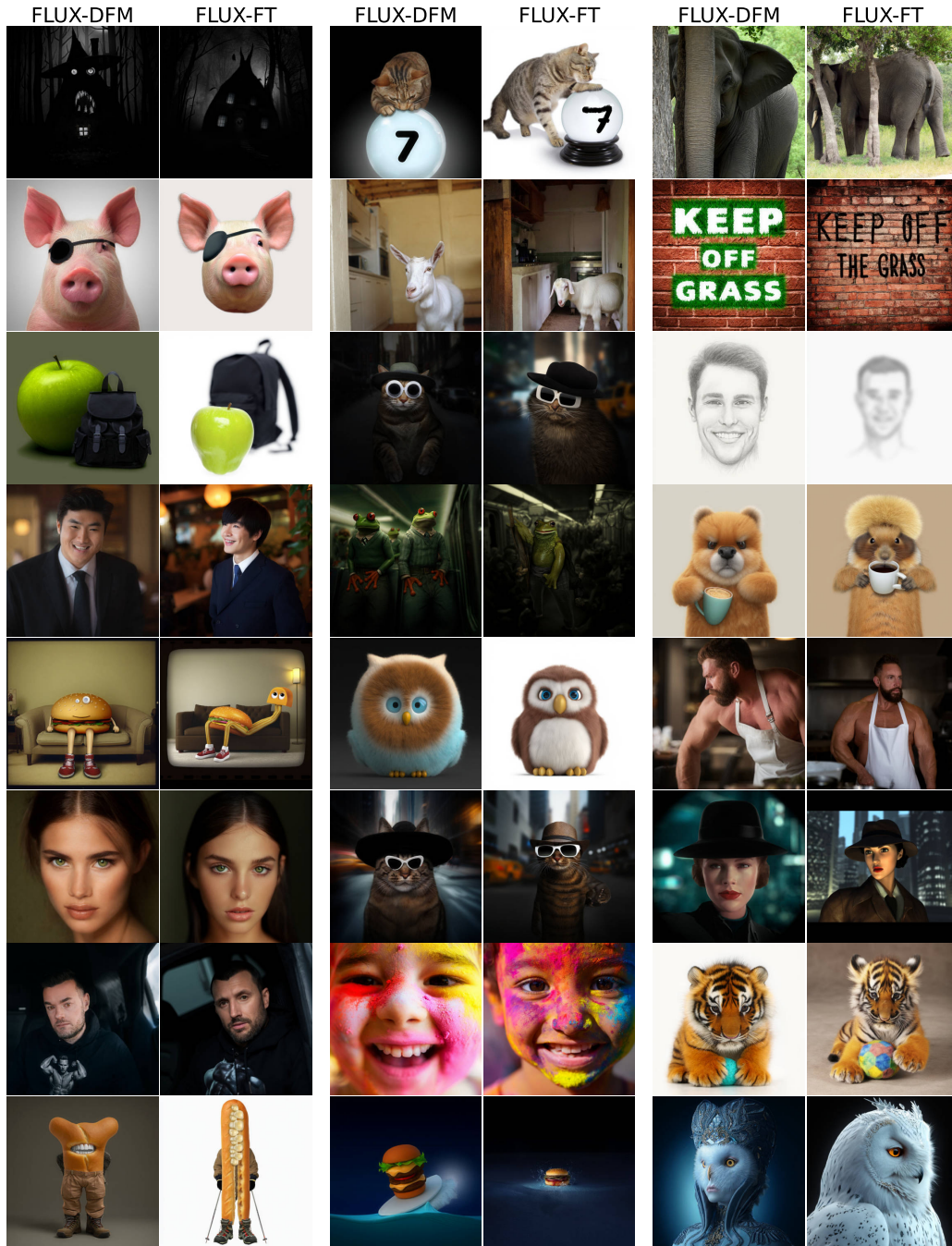


Figure 11: Comparison of finetuning FLUX-DEV with DFM against standard full finetuning trained finetuned for 24k steps. Samples are generated with cfg 4.5.

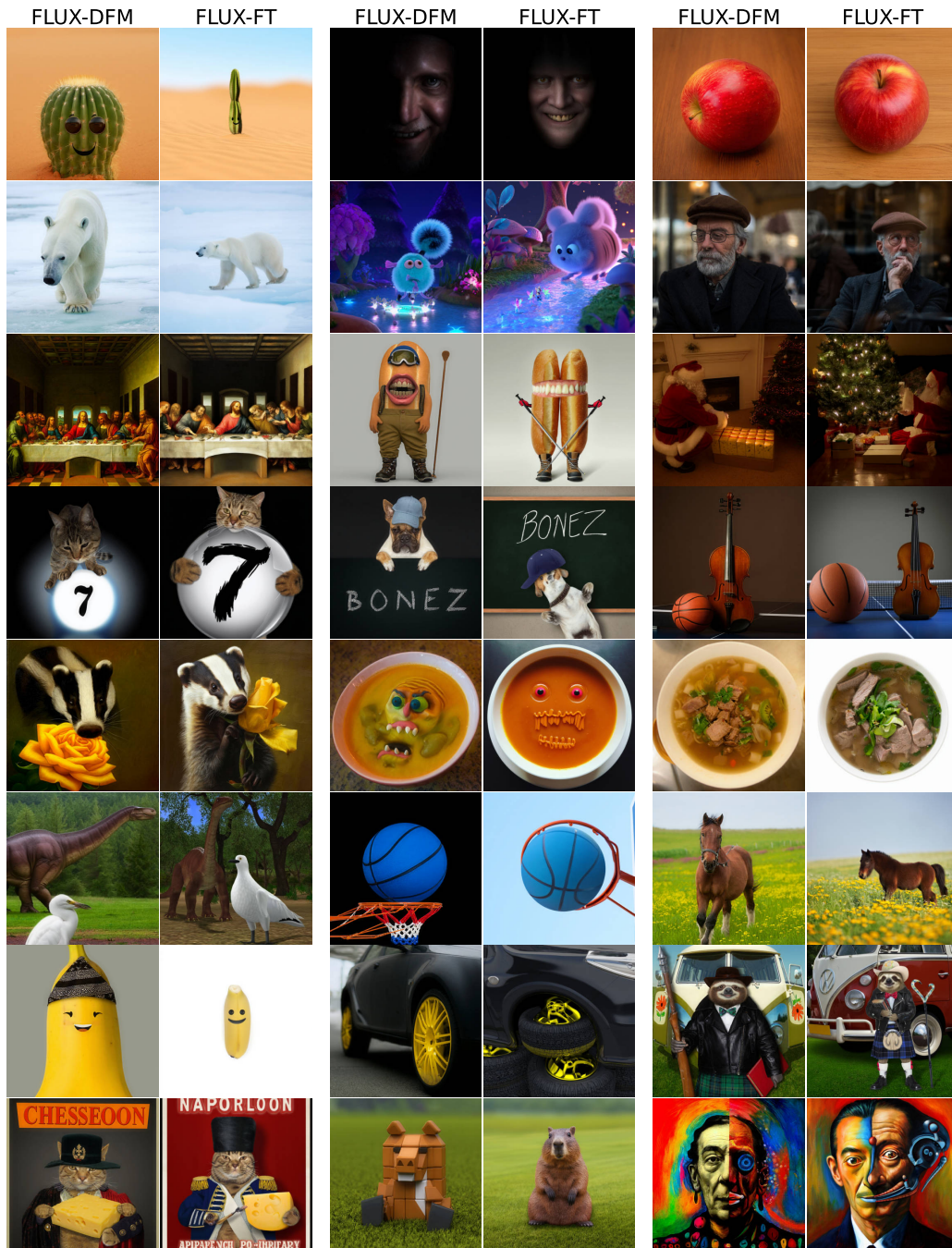


Figure 12: Comparison of finetuning FLUX-DEV with DFM against standard full finetuning trained finetuned for 24k steps. Samples are generated with cfg 4.5.



Figure 13: Ablation of per-stage sampling steps on DiT-XL trained on ImageNet-1K 512px [2] for 500k steps. Samples are selected to highlight the effects of varying sampling parameters. Samples are generated with cfg 3.0.

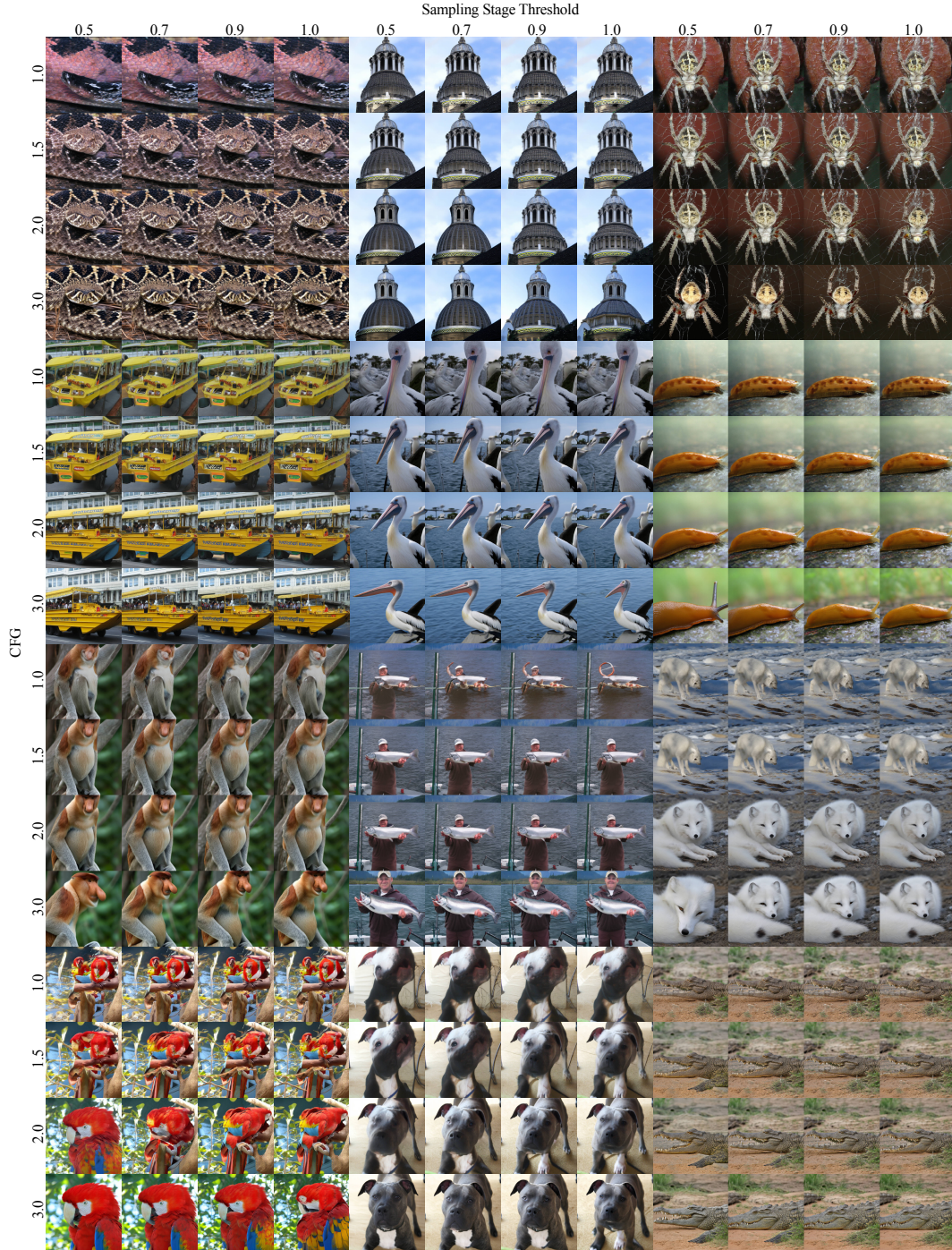


Figure 14: Ablation of the effect of cfg values and sampling threshold τ on DiT-XL trained on ImageNet-1K 512px [2] for 500k steps. Samples are selected to highlight the effects of varying sampling parameters. Samples are generated with cfg 3.0.



Figure 15: Qualitative results from selected classes on ImageNet-1K [2] 512px produced by DiT-XL trained with DFM for 1.3M steps. We use 30 and 10 sampling steps respectively for the first and second stages and a cfg value of 3.0.



Figure 16: Qualitative results from selected classes on ImageNet-1K [2] 512px produced by DiT-XL trained with DFM for 1.3M steps. We use 30 and 10 sampling steps respectively for the first and second stages and a cfg value of 3.0.



Figure 17: Qualitative results from selected classes on ImageNet-1K [2] 512px produced by DiT-XL trained with DFM for 1.3M steps. We use 30 and 10 sampling steps respectively for the first and second stages and a cfg value of 3.0.

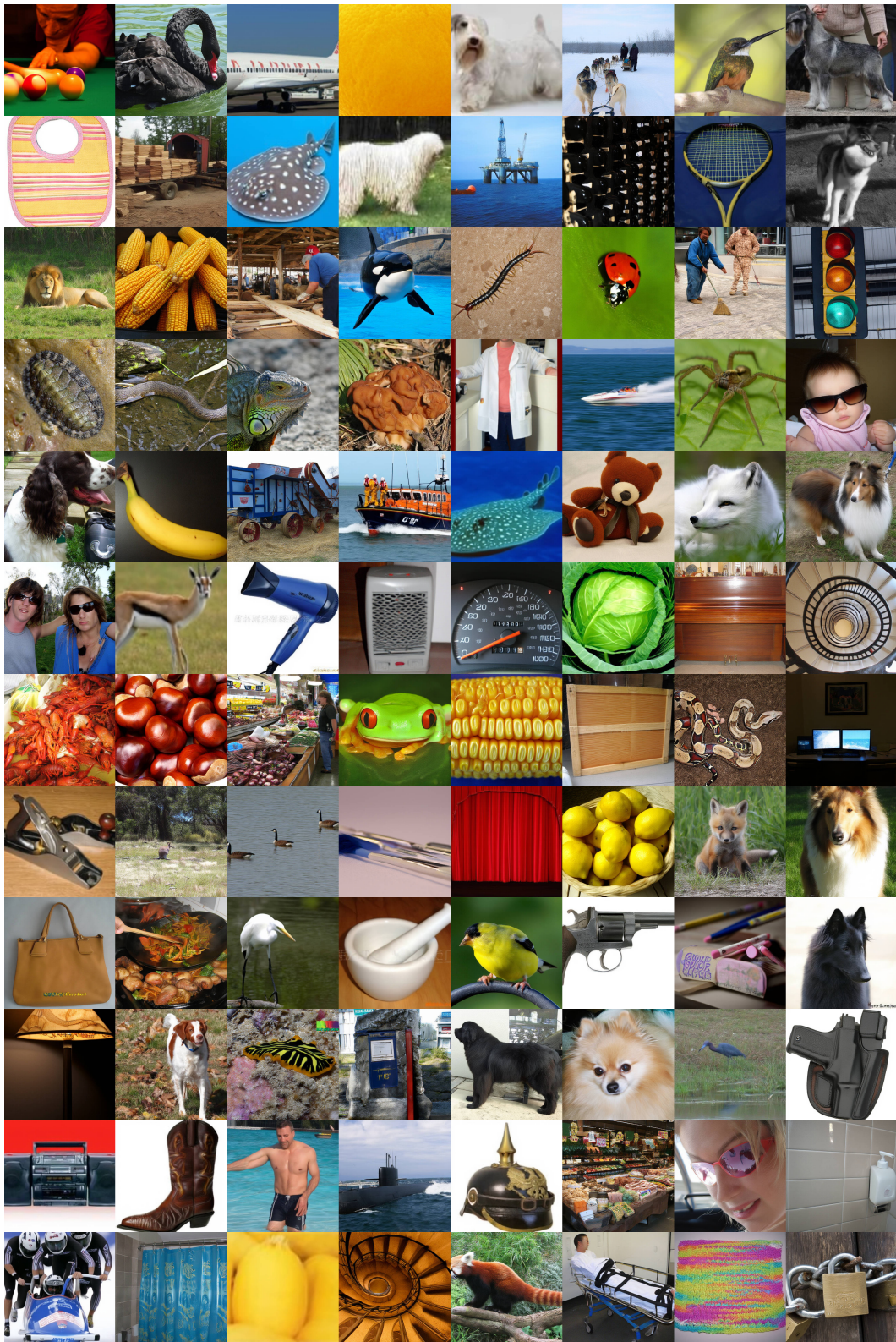


Figure 18: Fully uncured samples from ImageNet-1K [2] 512px produced by DiT-XL trained with DFM for 1.3M steps. We use 30 and 10 sampling steps respectively for the first and second stages and a cfg value of 3.0.

References

- [1] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv*, 2019.
- [2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [3] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv*, 2024.
- [4] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.
- [5] Black Forest Labs. Flux, 2024. URL <https://bfl.ai/>.
- [6] Mang Ning, Mingxiao Li, Jianlin Su, Haozhe Jia, Lanmiao Liu, Martin Beneš, Albert Ali Salah, and Itir Onal Ertugrul. Dctdiff: Intriguing properties of image generative modeling in the dct space. *arXiv*, 2024.
- [7] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.
- [8] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023.
- [9] Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. In *ICML*, 2025.
- [10] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.
- [11] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwei Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wenten Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025.
- [12] Haiyang Wang, Yue Fan, Muhammad Ferjad Naeem, Yongqin Xian, Jan Eric Lenssen, Liwei Wang, Federico Tombari, and Bernt Schiele. Tokenformer: Rethinking transformer scaling with tokenized model parameters. *arXiv*, 2025.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].

- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”.**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The main claims made in the paper reflect the contributions highlighted at the end of the Introduction section. We perform extensive experimental evaluation in Section 4 supporting the claimed performance improvements over baselines.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We provide a "Limitations" section as part of the main paper in Section 5 providing details on the main observed limitations of the framework.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper does not provide theoretical results

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The framework and its implementation details are comprehensively described in Section 4.1 and the *Appendix*. We base our architecture of the widely-used DiT [7], we comprehensively describe our evaluation settings and produce our main results on academic datasets (Imagenet-1K [2] and Kinetics-700 [1]) to favor reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may

be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: At the time of submission we do not possess authorization to publish our code. However, we comprehensively describe our framework and evaluation procedures to allow reproducibility and produce our main results on public datasets.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide complete details on the experimental settings in Section 4.1 and the *Appendix*.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Due to the expensive nature of generative models training and inference, we do not provide statistical significance information.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We detail in the *Appendix* details for the computational resources on which each experiment is run.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The work adheres to the NeurIPS Code of Ethics

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The work falls within the umbrella of generative methods for visual modalities. No special considerations apply related to its progressive generation formulation.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: We do not consider safeguards.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: We provide citations for the used models and data and utilize them in compliance to their license. We report dataset licenses in the *Appendix*.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[NA\]](#)

Justification: The paper does not release new assets.

Guidelines: The paper does not release new assets.

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

568 • Including this information in the supplemental material is fine, but if the main contribu-
569 tion of the paper involves human subjects, then as much detail as possible should be
570 included in the main paper.

571 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation,
572 or other labor should be paid at least the minimum wage in the country of the data
573 collector.

574 **15. Institutional review board (IRB) approvals or equivalent for research with human**
575 **subjects**

576 Question: Does the paper describe potential risks incurred by study participants, whether
577 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)
578 approvals (or an equivalent approval/review based on the requirements of your country or
579 institution) were obtained?

580 Answer: [NA]

581 Justification: The paper does not involve crowdsourcing nor research with human subjects

582 Guidelines:

583 • The answer NA means that the paper does not involve crowdsourcing nor research with
584 human subjects.

585 • Depending on the country in which research is conducted, IRB approval (or equivalent)
586 may be required for any human subjects research. If you obtained IRB approval, you
587 should clearly state this in the paper.

588 • We recognize that the procedures for this may vary significantly between institutions
589 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
590 guidelines for their institution.

591 • For initial submissions, do not include any information that would break anonymity (if
592 applicable), such as the institution conducting the review.

593 **16. Declaration of LLM usage**

594 Question: Does the paper describe the usage of LLMs if it is an important, original, or
595 non-standard component of the core methods in this research? Note that if the LLM is used
596 only for writing, editing, or formatting purposes and does not impact the core methodology,
597 scientific rigorousness, or originality of the research, declaration is not required.

598 Answer: [NA]

599 Justification: The core method development in this research does not involve LLMs as any
600 important, original, or non-standard components

601 Guidelines:

602 • The answer NA means that the core method development in this research does not
603 involve LLMs as any important, original, or non-standard components.

604 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
605 for what should or should not be described.