DiverseDiT: Towards Diverse Representation Learning in Diffusion Transformers

Supplementary Material

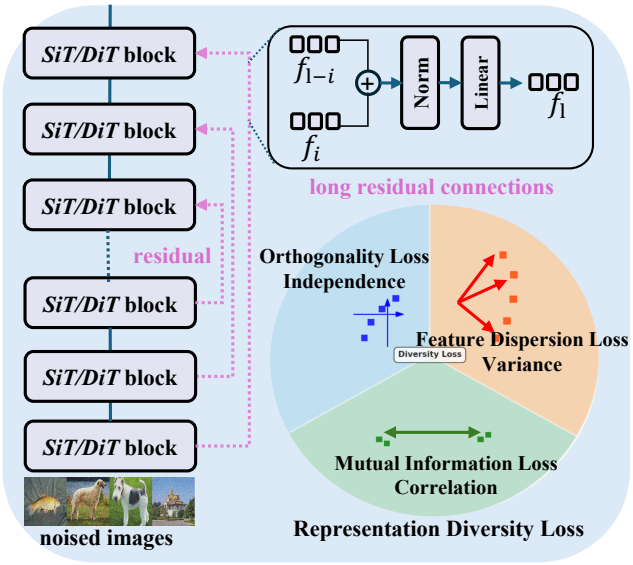

Figure A1. **Detailed diagram of our proposed DiverseDiT.** DiverseDiT incorporates long residual connections to diversify input representations across blocks and a representation diversity loss to encourage blocks to learn distinct features.

A. Appendix Overview

This supplementary material is organized as follows: First, we present more implementation details in Sec. B. Followed by the evaluation details and brief introduction of comparison baseline methods in Sec. C and Sec. D, respectively. Then, Sec. E shows the detailed quantitative results of our comprehensive analysis in Sec. 2. In the following, we present more quantitative comparison results under various settings in Sec. F and more ablation results in Sec. G. Moreover, Sec. H discusses the limitations and potential future works of our method. Finally, Sec. I illustrates more uncurated images generated by our proposed method.

B. More Implementation Details

Detailed diagram of our DiverseDiT. Our diversity loss is theoretically motivated from the following perspectives: 1) introducing an explicit inductive bias that encourages block-wise diversity to model the underlying observed distribution; 2) improving the representational orthogonality across blocks, thus reducing the redundancy and mutual correlation of different blocks; 3) promoting the coverage of the representation space, enabling blocks to specialize in complementary structures. For our proposed long residual connections and representation diversity loss, we first concatenate the hidden features of two blocks and perform layer normalization and a lightweight linear layer, as illustrated in Fig. A1.

More implementation details. We implement our proposed techniques on the original SiT [44] and REPA [70] implementation, leaving other details unchanged. Regarding the representation diversity loss, we calculate the corresponding loss items following the definition of each loss in Sec. 3, we randomly select 10 layers for computing the loss, as we find that computing on more layers gains similar performance improvement. Such observation aligns with that of DispLoss [62], where the effect of representation diversity loss propagates to other blocks, even though it is not directly applied to them. Further, the detailed setup for hyperparameters is presented in Tab. A1. To speed up the training process and save pre-processing time, we adopt mixed-precision (fp16) with gradient clipping and pre-compute VAE latents with stable diffusion VAE [51] (sd-vae-ft-mse) following REPA. For all optimization, we adopt AdamW [41] with a learning rate of 1e-4, and the training batchsize is 256. When classifier-free guidance (CFG) [24] is applied for generating images, we use the same guidance interval as that of REPA, which has been identified to improve the model performance, also illustrated in our results in Tab. A4. Additionally, we implement our method on the official MeanFlow [17] and follow their default configurations for training and evaluation for one-step generation experimental evaluation.

Sampler. Following the practice of REPA, we employ the Euler-Maruyama sampler with the SDE sampling with a diffusion coefficient σ_t. The sampling step for generating each image is set to 250.

Discussion on hyperparameter sensitivity. In our evaluation, we use the same hyperparameter settings when applying our method to various models, namely SiT [44], REPA [70], DispLoss [62], SRA [27] and MeanFlow [17]. Our method consistently improves performance across different backbones and different sampler settings (multi-step and one-step sampling). These results indicate that although our diversity loss introduces several hyperparameters, it is robust to hyperparameter choices.

Computing resources. All experiments were conducted on NVIDIA H100 (80GB) or H200 (141GB) GPUs. For training, the speed is about 5.8 steps/s for training SiT-XL +

Table A1. **Hyperparameter setup for the main experiment.** We strictly follow the setup of the baseline SiT and REPA for our training and evaluation for fair comparison.

	Table 1 (SiT-B)	Table 1 (SiT-L)	Table 1 (SiT-XL)	Table 2 (SiT-XL)	Table 3 (SiT-XL)
SiT + Ours					
Input dim.	32×32×4	32×32×4	32×32×4	32×32×4	64×64×4
Num. layers	12	24	28	28	28
Hidden dim.	768	1,024	1,152	1,152	1,152
Num. heads	12	16	16	16	16
REPA + Ours					
λ	0.5	0.5	0.5	0.5	0.5
Alignment depth	5	8	8	8	8
$\text{sim}(\cdot,\cdot)$	cos. sim.	cos. sim.	cos. sim	cos. sim.	cos. sim.
Encoder $f(x)$	DINOv2-B	DINOv2-B	DINOv2-B	DINOv2-B	DINOv2-B
Optimization					
Loss adaptive range	[0.1, 0.5]	[0.1, 0.5]	[0.1, 0.5]	[0.1, 0.5]	[0.1, 0.5]
Training iteration	400K	400K	400K	4M	1M
Training Batch size	256	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
lr	0.0001	0.0001	0.0001	0.0001	0.0001
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Interpolants					
α_t	$1-t$	$1-t$	$1-t$	$1-t$	$1-t$
σ_t	t	t	t	t	t
w_t	σ_t	σ_t	σ_t	σ_t	σ_t
Training objective	v-prediction	v-prediction	v-prediction	v-prediction	v-prediction
Sampler	Euler-Maruyama	Euler-Maruyama	Euler-Maruyama	Euler-Maruyama	Euler-Maruyama
Sampling steps	250	250	250	250	250
Guidance	1.0	1.0	1.0	1.35	1.35

Ours, and it takes about 1.38 hours to generate 50,000 images for evaluation (10.04 images/s). We have uploaded the compute report for detailed GPU hours used for our analysis and evaluation.

Pretrained vision foundation models for representation alignment. In our systematic analysis and experimental evaluation, we use three pretrained visual encoders as external representation guidance, namely DINOv2-B [48], MAE [22], and MoCov3 [9]. For DINOv2-B[1] and MAE[2] models, we download the pretrained model officially released by the original authors. Regarding the MoCov3 model, we download the -L version from the implementation of RCG[3] [38]; following REPA. For representation alignment, we perform projection with three MLP layers with SiLU activations following the exact configuration of REPA.

- **DINOv2** [48]: DINOv2 employs a vision transformer (ViT) architecture and learns self-supervised representations by enforcing consistency between different views of an image. It measures the feature distance between the representations of real and generated images, capturing high-level semantic information.
- **MAE** [22]: MAE trains an encoder and a lightweight decoder with a reconstruction objective. It learns to reconstruct masked patches of an image, learning robust representations.
- **MoCov3** [9]: based on the philosophy of contrastive learning, MoCov3 empirically revisits prior MoCo series [8, 21] and scales to larger model sizes to learn representations by maximizing the similarity between different views of the same image while minimizing the similarity between views of different images.

C. Evaluation Details

Implementation details for evaluation. We strictly follow the setup and use the same reference batches of ADM [14] for evaluation, following their official implementation. Specifically, for 256×256 evaluation, we generate 50,000 images and convert them into a .npz file and compute the quantitative metrics from the reference batch (VIRTUAL_imagenet256_labeled.npz) of ADM[4]. Similarly, we quantify the result of 512×512 evaluation via computing

[1] https://github.com/facebookresearch/dino
[2] https://github.com/facebookresearch/mae
[3] https://github.com/LTH14/rcg
[4] https://github.com/openai/guided-diffusion/tree/main/evaluations

the metrics between our generated images and the reference batch (VIRTUAL_imagenet512.npz).

Evaluation metrics. We adopt several popular metrics for evaluation: Fréchet Inception Distance (FID) [23], structural FID (sFID) [46], Inception Score (IS) [52], Precision (Prec.) and Recall (Rec.) [33]. Their main concepts are:

- **FID** [23] computes the Fréchet Distance between two observed data distributions, which represent the feature distributions of synthesized and real images extracted by the pre-trained Inception-V3 [59]. Formally, FID is calculated by

$$\mathrm{FID}(\mathrm{X},\mathrm{Y}) = \|\mu_s - \mu_r\|^2 + \mathrm{Tr}\left(\Sigma_s + \Sigma_r - 2\left(\Sigma_s\Sigma_r\right)^{\frac{1}{2}}\right), \tag{A1}$$

  where X and Y represent the synthesized distribution and real distribution, respectively. μ and Σ correspond to the mean and variance of the distribution, and $\mathrm{Tr}(\cdot)$ is the trace operation.
- **sFID** [46] is a variant of FID that aims to be more robust to structural differences between real and generated images. Instead of using the standard Inception-V3 features, sFID uses features extracted from different layers of the network, focusing on structural information. This makes it more sensitive to the arrangement of objects and their parts, and less sensitive to color or texture differences.
- **IS** [52] measures the quality and diversity of generated images. It uses the Inception-V3 model to predict the class of each generated image. A good Inception Score means that the generated images are clear and belong to a specific class (high confidence), and that the generated images cover a wide range of classes (high diversity). Formally, Inception Score is calculated by:

$$\mathrm{IS} = \exp(\mathbb{E}_{\mathrm{x}\sim\mathrm{X}}[D_{KL}(p(\mathrm{y}|\mathrm{x})||p(\mathrm{y}))]) \tag{A2}$$

  where x represents the generated images, X is the distribution of generated images. $p(\mathrm{y}|\mathrm{x})$ is the conditional probability distribution of the class y given the image x, predicted by the Inception model, $p(\mathrm{y})$ is the marginal probability of class y. D_{KL} is the Kullback-Leibler divergence.
- **Precision and Recall** [33] are used to evaluate the quality of generated images by comparing them to real images. Precision measures how much the generated images resemble real images, while Recall measures how much of the real image distribution is captured by the generated images.
- **Centered Kernel Alignment** (CKA) is a widely adopted metric for quantifying neural network representations [12, 32], which has been demonstrated with several advantages: 1) CKA is invariant to orthogonal transformation and isotropic scaling, thus it is stable under various image transformations; 2) CKA can capture the non-linear correspondence between representations benefit from its kernel mapping in the kernel space; and 3) CKA can quantify the correspondence between different features across different widths, whereas previous metrics fail [32]. Formally, CKA is normalized from Hilbert-Schmidt Independence Criterion (HSIC) [18] to be invariant to orthogonal transformation and isotropic scaling:

$$\mathrm{CKA}(\mathrm{X},\mathrm{Y}) = \frac{\mathrm{HSIC}(\mathrm{x},\mathrm{y})}{\sqrt{\mathrm{HSIC}(\mathrm{x},\mathrm{x})\mathrm{HSIC}(\mathrm{y},\mathrm{y})}}. \tag{A3}$$

  HSIC identifies whether two distributions (X, Y) are independent: $\mathrm{HSIC}(K,L) = \frac{1}{(n-1)^2}\mathrm{Tr}(KHLH)$, where $K_{ij} = k(\mathrm{x}_i,\mathrm{x}_j)$ and $L_{ij} = l(\mathrm{y}_i,\mathrm{y}_j)$, where k and l are kernels. Note that for kernel selections of k and l in Eq. (A3), we find that different kernels (RBF, polynomial, and linear) reflect similar discrepancies across various representations of DiTs, while the RBF kernel contributes to the distinguishability of quantitative results.

D. Comparison Baselines

In this part, we briefly introduce the main concept of baseline methods that are used for our evaluation.

D.1. Multi-step baseline models

- **ADM** [14] achieved improved synthesis performance with architectural improvement on traditional Unet-based diffusion models and developed classifier guidance to improve the synthesis fidelity for class-conditional tasks.
- **VDM++** [29] demonstrated that commonly used diffusion model objectives equate to a weighted integral of ELBOs over different noise levels, where the weighting depends on the specific objective used. Based on this, a sample adaptive noise schedule was introduced for improved training efficiency.
- **CDM** [26] proposed a cascaded architecture that trains multiple models across different resolutions, starting from the lowest resolution to higher resolution.
- **LDM** [51] developed latent diffusion models that train diffusion in a low-dimensional compressed latent space to improve the training efficiency. Specifically, the images are first encoded into latent codes and then added noise for training, and the denoised latents are decoded back to pixel space for sampling.
- **MDTv2** [16] introduced an asymmetric encoder-decoder paradigm for efficient training of diffusion transformer. To stabilize the training and improve model performance, they further employ U-Net-like

Table A2. **Detailed quantitative results of our systematic analysis**. All implementation details strictly follow the default settings of SiT and REPA for our investigation. All baselines are reported using vanilla-REPA [70] for training.

Model	Alignment?	Encoder.	Align Depth.	Iter.	FID$_\downarrow$	sFID$_\downarrow$	IS$_\uparrow$	Prec.$_\uparrow$	Rec.$_\uparrow$
SiT-B									
	✗	✗	✗	50k	223.93	2.26	250.89	-	-
	✗	✗	✗	100k	-	6.77	40.09	0.51	0.63
	✗	✗	✗	200k	-	6.77	40.09	0.51	0.63
	✗	✗	✗	400k	-	6.77	40.09	0.51	0.63
	✗	✗	✗	450k	12.04	5.24	110.19	0.71	0.54
REPA									
	✓	DINOv2-B	5	450k	5.37	5.35	175.07	0.75	0.58
	✓	DINOv2-B	8	450k	7.67	5.60	150.87	0.72	0.58
	✓	DINOv2-B	10	450k	10.85	6.12	128.34	0.70	0.58
	✓	DINOv2-B	[2,5,8]	450k	13.25	5.27	105.64	0.70	0.55
	✓	DINOv2-B	[3,6,9]	450k	13.54	5.50	104.88	0.69	0.56
	✓	MAE	5	450k	10.15	5.11	123.24	0.72	0.55
	✓	MAE	8	450k	11.40	5.22	115.20	0.72	0.55
	✓	MAE	10	450k	12.11	5.30	111.01	0.71	0.55
	✓	DINOv2, MAE	5	450k	5.77	5.10	166.15	0.76	0.57
	✓	DINOv2, MAE	8	450k	7.44	5.37	150.12	0.73	0.57
	✓	DINOv2, MAE	10	450k	11.03	6.17	124.89	0.70	0.55
	✓	DINOv2, MAE	[3,8]	450k	11.46	5.20	113.95	0.72	0.55
	✓	DINOv2, MAE	[5,10]	450k	11.73	5.22	111.77	0.71	0.54
	✓	DINOv2, MoCoV3	[3,8]	450k	11.36	5.24	115.75	0.71	0.55
	✓	DINOv2, MoCoV3	[5,10]	450k	12.71	6.08	104.46	0.66	0.56

long-shortcuts in the encoder and dense input-shortcuts in the decoder.

- **MaskDiT** [73] used a similar encoder-decoder architecture with MDTv2, while the model was trained with an auxiliary reconstruction objective like [22] to reconstruct masked inputs.
- **SD-DiT** [75] extended the reconstruction-based MaskDiT architecture, while introducing a self-supervised discrimination objective with a momentum encoder for improved training.
- **DiT** [49] proposed to replace the conventional Unet-based architectures with transformers and further explored different condition injection mechanisms for conditional generation.
- **SiT** [44] systematically investigated the connections between discrete diffusion to continuous flow matching and developed practical training configurations for achieving strong synthesis performance.
- **REPA** [70] connected diffusion training dynamics and representation learning, revealing that pretrained external guidance could facilitate the representation learning of diffusion transformers.
- **REG** [64] further advanced REPA with a decoupled representation alignment technique, which entangled image latents and class tokens to imporve the conditional discrimination capability.
- **E2E-REPA** [35] unlocked a end-to-end training paradigm for joint tuning both the VAE and diffusion models throughout the training process, improving the VAE itself and downstream generation performance simultaneously.
- **SRA** [27] leveraged representations from later layers with lower noise of the EMA teacher to guide representations of earlier layers with higher noise, enabling a scheme of self-alignment.
- **DispLoss** [27] introduced a regularized dispersive loss to encourage internal features to spread out in the embedding space, thus facilitating the model to learn informative representations.

D.2. One-step baseline models

- **MeanFlow** [17] introduced average velocity that was defined as the ratio of displacement to a time interval, with displacement given by the time integral of the instantaneous velocity. An intrinsic relation between the average and instantaneous velocities was then derived to guide efficient and effective one-step generative training.
- **Shortcut** [15] enhanced the few-step flow matching by adding a self-consistency loss, designed to learn the relationships between flow behaviors observed at different discrete time points.
- **IMM** [74] learned a model that enforces self-consistency among stochastic interpolants evaluated at different points in time.
- **iCT** [57] leveraged consistency constraints across net-

work outputs at different time steps to ensure that they predict the same endpoints along the trajectory.

E. Detailed Results of Our Analysis

Detailed Quantitative Results. Tab. A2 presents the detailed quantitative results of our systematic analysis in Sec. 2. These quantitative results consistently reflect the findings in our analysis: 1) aligning external representations on more blocks (*e.g.,* aligning DINOv2-B features on [2, 5, 6]-th blocks and [3, 6, 9]-th blocks) does not bring obvious performance improvements, indicating that indiscriminate alignment can be detrimental and reduce the overall diversity between blocks, such observation is also reflected by the CKA similarity heatmaps in Fig. 2. 2) aligning with earlier blocks (e.g., Block 5) generally results in better performance than aligning with later blocks (e.g., Block 10), as evidenced by the lower FID scores, which is also identified in the original REPA. 3) combining different external encoders (DINOv2 and MAE) on different blocks does not consistently improve performance, further indicating that the representation diversity across blocks is a crucial factor for high-quality synthesis. Together, the quantitative results and CKA similarity heatmaps in Fig. 2 consistently reveal that the key for representation learning is increasing the discrepancies of block representations. Which provides explainable motivations for our proposed method in explicitly encouraging the representation diversity from the perspective of input and internal features' correlations.

CKA similarity across various timesteps. In Fig. 2, we present the CKA similarity heatmaps between representations of different blocks at the final denoising timestep. To investigate the difference of block representations across different timesteps, we calculate their representational discrepancies of different timesteps in Fig. A2. The results are computed from aligning DINOv2-B features on REPA-B for 400K training iterations like Fig. 2. We could observe that the representation similarities between different blocks across different timesteps show a very similar pattern. That is, the representational discrepancy across diffusion transformer blocks originates from the internal representation instead of different denoising timesteps. Moreover, we can see that as the inference steps increases, the representational discrepancy between different blocks at different timesteps tends to slightly increase as well. Such observation is reasonable because the noisy hidden states become less noisy throughout the sampling process.

F. More Quantitative Results

Improving representation learning across various model scales on ImageNet 512×512. Tab. A3 presents the quantitative results of applying our proposed techniques to SiT and REPA across various model scales on ImageNet 512×512 without CFG. Similar to the results of ImageNet 256×256 in Tab. 1, our method consistently improves the performance of both SiT and REPA models across all scales, as evidenced by the reduction in FID and sFID scores and the increase in IS. Specifically, when applied to SiT-B, our method achieves a significant improvement in FID score (from 43.46 to 33.18 and sFID from 7.53 to 6.92), while also improving the IS score from 36.80 to 45.09. Similar improvements can be observed for REPA-B, with FID improving from 30.13 to 23.82 and IS increasing from 53.92 to 64.62. The benefits of our method are also evident for larger models. For SiT-XL, our approach reduces FID from 19.65 to 17.68 and increases IS from 71.57 to 76.45. For REPA-XL, the FID decreases from 7.91 to 7.18, and the IS increases from 127.83 to 137.09. These results further indicate that our method is effective in improving the representation learning capabilities of both SiT and REPA models, regardless of their scale. The consistent improvements in FID, sFID, and IS across different model sizes demonstrate the robustness and generalizability of our approach. The improvements in Precision and Recall also suggest that our method leads to better alignment between the generated images and the real data distribution.

Table A3. **Variation in model-scale on ImageNet 512×512 without CFG**. Our proposed method brings consistent performance gains across all model-scales when applied to both SiT and REPA. All baselines are reported using vanilla-REPA [70] for training.

Model	Iter.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B	400k	43.46	7.53	36.80	0.60	0.64
+ (Ours)	400k	**33.18**	**6.92**	**45.09**	**0.67**	**0.65**
REPA-B	400k	30.13	7.79	53.92	0.68	**0.64**
+ (Ours)	400k	**23.82**	**7.76**	**64.62**	**0.70**	0.63
SiT-L	400k	22.75	5.78	64.05	0.73	**0.63**
+ (Ours)	400k	**19.19**	**5.74**	**71.78**	**0.76**	0.61
REPA-L	400k	10.82	5.52	106.43	0.78	0.63
+ (Ours)	400k	**9.83**	**5.49**	**114.00**	**0.78**	**0.64**
SiT-XL	400k	19.65	5.55	71.57	0.75	**0.60**
+ (Ours)	400k	**17.68**	**5.48**	**76.45**	**0.77**	**0.60**
REPA-XL	400k	7.91	5.41	127.83	**0.79**	**0.65**
+ (Ours)	400k	**7.18**	**5.38**	**137.09**	0.78	0.64

Comparison results across different model scales with different CFG scales. We mainly present comparison results without using classifier-free guidance (CFG) [47] in the main paper. In this part, we present comparison results across different model scales with CFG enabled to further investigate its impact and our performance. Specifically, we conduct experiments on ImageNet 256x256 on REPA using the DINOv2-B encoder for 400K training iterations across

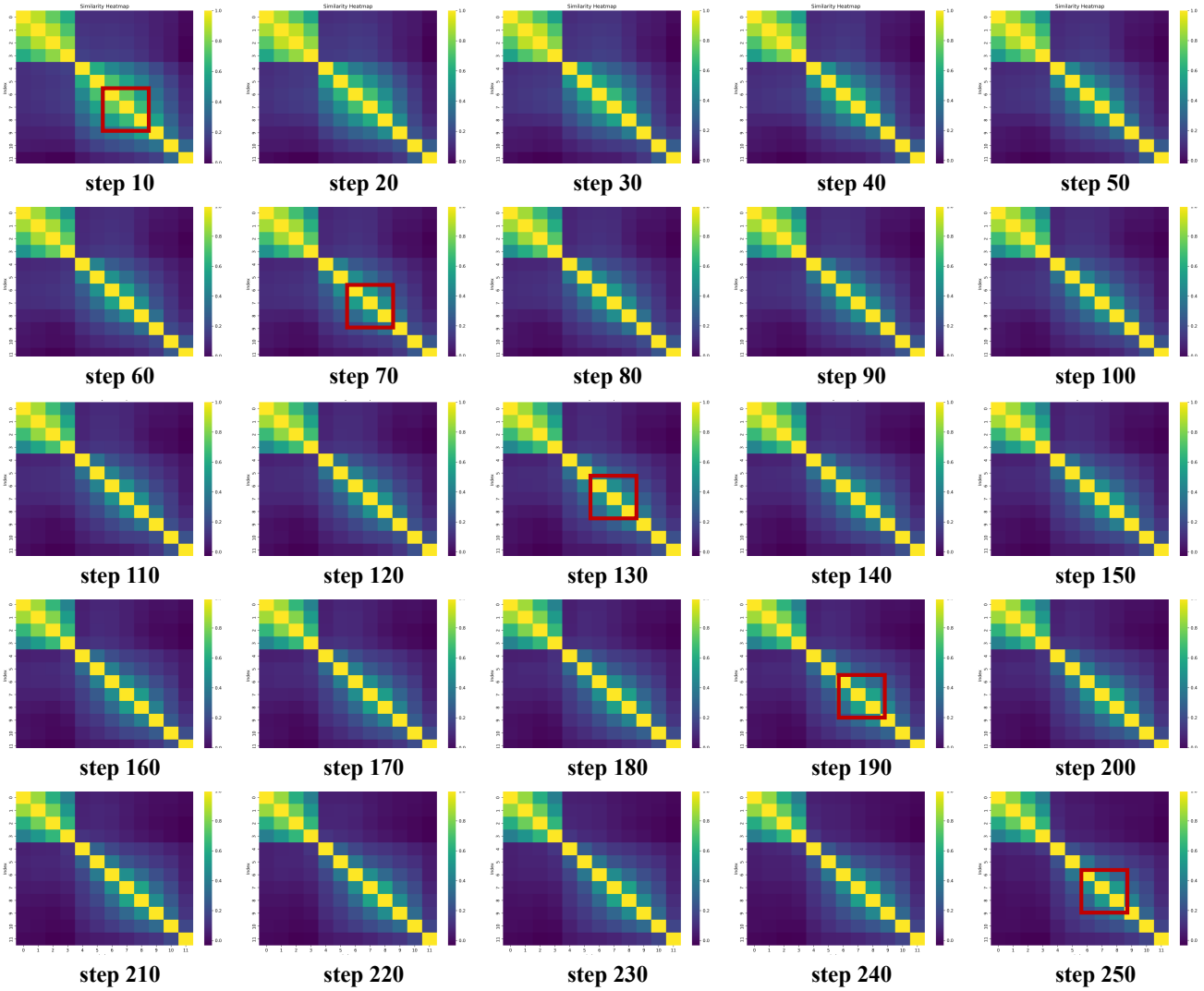

Figure A2. **CKA representation similarities across different timesteps.** The representational discrepancies across different timesteps show similar correlations.

different model sizes (REPA-B, REPA-L, and REPA-XL). We systematically evaluated the performance with different CFG scales (1.0, representing no classifier-free guidance, and 1.35). Tab. A4 presents a detailed analysis of the impact of Classifier-Free Guidance (CFG) scale on the performance of our proposed method when applied to REPA models of varying sizes (REPA-B, REPA-L, and REPA-XL).

First, the results consistently demonstrate that increasing the CFG scale from 1.0 to 1.35 leads to significant improvements in image quality and diversity across all model scales. This is evidenced by the substantial increase in IS scores and the decrease in FID scores observed across all REPA model sizes when CFG is enabled. Second, our proposed method also gains consistent performance improvement across different model scales when CFG is enabled. For instance, our model advances the FID score of REPA-B from 12.47 to 8.33 and IS score from 107.38 to 134.16 with CFG=1.35, attaining a $>32\%$ performance improvement on FID. Similarly, our model advances the FID score of REPA-XL from 3.50 to 3.16 and the IS score from 188.96 to 194.36 with CFG=1.35.

Furthermore, Tab. A5 presents the comparison results on on ImageNet 512×512 with CFG=1.35. Across all model scales, our method consistently improves the FID and sFID scores when CFG is used, indicating enhanced image quality and fidelity. For example, when applied to SiT-B, our method reduces the FID from 43.46 to 33.18 and the sFID from 7.53 to 6.92. Similarly, for REPA-B, the FID decreases from 30.13 to 23.82. Together with the results that were tested without using CFG, these results demonstrate

Table A4. **Variation in alignment depth on ImageNet 256×256 with different CFG scales**. CFG=1.0 means no classifier-free guidance is applied. Our proposed method brings consistent performance gains across all model-scales when applied to REPA with different alignment depths and evaluated with different CFG scales.

Model	Iter.	Encoder.	Align Depth.	CFG.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
REPA-B									
	400K	DINOv2-B	5	1.0	22.99	6.70	64.73	0.59	0.65
	400K	DINOv2-B	5	1.35	12.47	5.95	107.38	0.67	0.61
+Ours									
	400K	DINOv2-B	5	1.0	17.29	6.56	79.92	0.62	0.65
	400K	DINOv2-B	5	1.35	8.33	5.84	134.16	0.70	0.63
REPA-B									
	400K	DINOv2-B	8	1.0	27.94	7.19	54.32	0.56	0.64
	400K	DINOv2-B	8	1.35	16.46	6.38	90.97	0.64	0.62
+Ours									
	400K	DINOv2-B	8	1.0	23.27	6.82	62.63	0.59	0.65
	400K	DINOv2-B	8	1.35	12.92	6.08	104.46	0.66	0.63
REPA-L									
	400K	DINOv2-B	5	1.0	12.02	6.77	40.09	0.51	0.63
	400K	DINOv2-B	5	1.35	5.14	4.74	157.71	0.75	0.61
+Ours									
	400K	DINOv2-B	5	1.0	10.01	5.47	107.68	0.69	0.64
	400K	DINOv2-B	5	1.35	4.18	4.61	172.26	0.76	0.61
REPA-L									
	400K	DINOv2-B	8	1.0	9.57	5.34	113.42	0.69	0.66
	400K	DINOv2-B	8	1.35	3.86	4.82	183.18	0.75	0.63
+Ours									
	400K	DINOv2-B	8	1.0	8.47	5.42	123.03	0.69	0.67
	400K	DINOv2-B	8	1.35	3.39	4.80	196.08	0.76	0.63
REPA-XL									
	400K	DINOv2-B	5	1.0	8.27	5.19	123.85	0.69	0.66
	400K	DINOv2-B	5	1.35	3.33	4.73	196.52	0.75	0.64
+Ours									
	400K	DINOv2-B	5	1.0	8.18	5.01	126.63	0.70	0.65
	400K	DINOv2-B	5	1.35	3.17	4.71	198.30	0.77	0.62
REPA-XL									
	400K	DINOv2-B	8	1.0	8.73	5.21	118.68	0.69	0.65
	400K	DINOv2-B	8	1.35	3.50	4.72	188.96	0.76	0.63
+Ours									
	400K	DINOv2-B	8	1.0	8.09	5.02	123.23	0.70	0.65
	400K	DINOv2-B	8	1.35	3.16	5.60	194.36	0.77	0.62

the scalability and effectiveness of our proposed method to higher resolutions and different model sizes.

Quantitative results of applying our method on REPA with alignment on different blocks. Tab. A4 also provides insight into the impact of the depth of alignment on the performance of our proposed method. We evaluated the models with alignment depths of 5 and 8, while keeping other parameters constant. The results suggest that increasing the alignment depth from 5 to 8 can have varying effects depending on the model size, suggesting that the optimal alignment depth may depend on the interplay between model size. Despite these variations, our method consistently improves upon the baseline REPA models when performing alignment on different blocks, with or without CFG. For example, REPA-XL with our method and an alignment depth of 5 achieves an FID score of 3.17 at CFG 1.35, compared to 3.33 for the baseline. Similarly, the IS score improves from 196.52 to 198.30. This consistent trend of improvement, regardless of alignment depth, demonstrating the effectiveness of our approach in enhancing image generation. The consistent improvements observed across different alignment depths and model sizes further demonstrate the robustness and generalizability of our approach.

Table A5. **Variation in model-scale on ImageNet 512×512 with CFG=1.35**. Our proposed method brings consistent performance gains across all model-scales when applied to both SiT and REPA.

Model	Iter.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B	400k	43.46	7.53	36.80	0.60	0.64
+ (Ours)	400k	**33.18**	**6.92**	**45.09**	**0.67**	**0.65**
REPA-B	400k	30.13	7.79	53.92	0.68	**0.64**
+ (Ours)	400k	**23.82**	**7.76**	**64.62**	**0.70**	0.63
SiT-L	400k	22.75	5.78	64.05	0.73	**0.63**
+ (Ours)	400k	**19.19**	**5.74**	**71.78**	**0.76**	0.61
REPA-L	400k	10.82	5.52	106.43	0.78	0.63
+ (Ours)	400k	**9.83**	**5.49**	**114.00**	**0.78**	**0.64**
SiT-XL	400k	19.65	5.55	71.57	0.75	**0.60**
+ (Ours)	400k	**17.68**	**5.48**	**76.45**	**0.77**	**0.60**
REPA-XL	400k	7.91	5.41	127.83	**0.79**	**0.65**
+ (Ours)	400k	**7.18**	**5.38**	**137.09**	0.78	0.64

Table A6. **Ablation analysis on different components with CFG=1.35**.

Component	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B Baseline	23.28	6.00	65.23	0.61	0.60
SiT-B + `full`	16.21	5.45	84.00	0.66	0.60
w/o `diversity`	20.07	5.72	73.65	0.63	0.60
w/o `residual`	20.76	5.77	69.23	0.61	0.61
REPA	12.47	5.85	107.38	0.61	0.62
REPA-B + `full`	8.34	5.64	134.16	0.70	0.63
w/o `diversity`	10.75	5.75	115.42	0.68	0.62
w/o `residual`	11.02	5.77	112.93	0.64	0.62

Table A7. **Ablation analysis on selecting different number of layers for diversity loss**

$\mathcal{P}$	SiT-XL	5	10	15	20	all
FID↓	18.77	16.85	16.10	16.01	15.84	**15.77**
IS↑	71.44	77.62	79.47	82.05	83.95	**85.64**
Time (h)	18.66	19.90	21.02	23.45	25.96	28.50

G. More Ablation and Analysis Results

Ablation on layer selection $\mathcal{P}$. In our implementation for layer selection $\mathcal{P}$, we randomly select 10 layers to compute the diversity loss for experiments. To investigate its impact, here we testify the impact of $\mathcal{P}$ on SiT-XL (28 layers) for 400K training steps. The results in Tab. A7 show that selecting more layers improves the performance but increases the training time. In particular, selecting 10 layers yields a better trade-off between performance and efficiency.

Ablation on different loss variants. Here we further perform ablation on each loss component of the proposed diversity loss on SiT-B baseline. The results in Tab. A8 show the effectiveness of each loss, consistent with the findings of REPA results in Tab. 7. Specifically, using all components of the diversity loss (SiT-B + full) achieves the best performance and removing any single loss component degrades performance.

Table A8. **Ablation analysis on different loss variants on SiT-B baseline.**

Component	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B + `full`	28.05	6.04	50.66	0.57	0.63
`only` $\mathcal{L}_{\texttt{orth}}$	31.32	6.45	47.09	0.56	0.63
`only` $\mathcal{L}_{\texttt{MI}}$	29.97	6.21	48.23	0.57	0.63
`only` $\mathcal{L}_{\texttt{div}}$	36.12	6.64	45.04	0.55	0.62

Ablation on designed components with CFG. Tab. A6 presents the ablative results on the designed components of our DiverseDiT with CFG. We could see that applying the CFG consistently improves the overall scores. Similar to the results in Tab. 6, the results clearly demonstrate the importance of both the representation diversity loss and the long residual connections for optimal performance. Removing the diversity loss (w/o diversity) worsens the FID scores for both SiT-B (from 23.28 to 20.07) and REPA-B (from 12.47 to 10.75). Similarly, removing the long residual connections (w/o residual) also noticeably increases FID for both baseline models. Despite some performance degradation, we can observe that applying any of our proposed techniques to the baseline methods, *i.e.,* REPA-B and SiT-B, brings substantial performance improvements. For instance, with only long residual connections (w/o diversity), we achieve an FID of 20.07 on SiT-B and an FID of 10.75 on REPA-B, which are better than the original baseline results (23.28 for SiT-B and 12.47 for REPA-B). Similar conclusions could be observed from the results of only diversity loss (w/o residual) as well. These results confirm that both components of DiverseDiT play a crucial role in promoting diverse representation learning and improving the performance.

Effect of diversity loss variant with CFG. Tab. A9 presents an ablation analysis on different loss variants with CFG=1.35. Similar to the previous results, the table demonstrates the importance of each loss component for optimal performance. Removing any of the loss components, namely $\mathcal{L}orth$, $\mathcal{L}MI$, or $\mathcal{L}div$, degrades the FID score compared to the REPA-B + full configuration (8.34). While using only $\mathcal{L}orth$ results in an FID of 10.98, using only $\mathcal{L}MI$ gives an FID of 10.78, and using only $\mathcal{L}div$ improves the FID to 8.59. These results confirm that each loss component plays a role in improving the model's performance, which is also reflected by the better results compared with the REPA baseline when each loss is used in isolation.

Combining with existing methods for further improvement with CFG. Tab. A10 further explores the effect of combining our method with existing approaches, specifically DispLoss [62] and SRA [27], on the SiT-B baseline with CFG=1.35. Adding our method to the SiT-B

Table A9. **Ablation analysis on different loss variants with CFG=1.35.**

Component	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
REPA Baseline	12.47	5.85	107.38	0.61	0.62
REPA-B + `full`	8.34	5.64	134.16	0.70	0.63
`only` $\mathcal{L}_{\texttt{orth}}$	10.98	5.78	115.03	0.68	0.62
`only` $\mathcal{L}_{\texttt{MI}}$	10.78	5.76	115.95	0.69	0.63
`only` $\mathcal{L}_{\texttt{div}}$	8.59	5.77	131.69	0.70	0.63

Table A10. **Combining our method with prior approaches with CFG=1.35.**

Component	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
REPA Baseline	12.47	5.85	107.38	0.61	0.62
SiT-B Baseline	23.28	6.00	65.23	0.61	0.60
+ `Ours`	16.21	5.45	84.00	0.66	0.60
++ `DispLoss` [62]	13.73	5.76	95.31	0.68	0.60
+++ `SRA` [27]	11.25	5.37	108.15	0.69	0.61

baseline improves the FID from 23.28 to 16.21. Further combining with DispLoss results in an even lower FID of 13.73. This demonstrates that our method is complementary to existing techniques and can be combined with them to achieve further improvements in image generation quality. Note that SRA and DispLoss require no additional external models for representation alignment, and combining our proposed method with them achieves a better performance than that of REPA, which needs pretrained models as guidance, demonstrating the potential for representation learning through internal mechanisms.

H. Limitations and Future Work

Limitations. Despite a comprehensive investigation, our analysis could be extended in several aspects: For instance, whether DiverseDiT can be effectively adapted to diverse generation tasks, such as text-to-image synthesis or image editing, remains an open question. Besides, performing similar analysis on representation learning of other models like large-language models might reveal more interesting findings. Additionally, we do not perform extensive hyperparameter searching for the optimal performance in our experiments, the full potential of our proposed representation diversity loss could be further unlocked. Nevertheless, our study could provide potential guidelines for developing more effective methods in learning informative representations.

Future work. For future work, we plan to extend our analysis and evaluation on text-to-image synthesis tasks. We aim to investigate the application of our representation diversity loss to other generative models and modalities, such as video generation and 3D shape generation. Exploring different architectures and training strategies in conjunction with our proposed loss function could potentially lead to even more significant improvements in the quality and diversity of generated content. Meanwhile, we intend to explore theoretical connections between representation diversity and other desirable properties of generative models, such as robustness to adversarial attacks and generalization to unseen data distributions. Furthermore, considering that our proposed diversity loss alone could likely be applied as a fine-tuning step for pre-trained models without any architectural changes, we plan to explore this in our ongoing research.

I. More Qualitative Results

We present more uncurated generation results of our DiverseDiT-XL on ImageNet 256×256 in Fig. A3 - Fig. A19 with CFG (w = 4.0).

Figure A3. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Great white shark" (2).

Figure A4. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Chickadee" (19).

Figure A5. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is “Terrapin” (36).

Figure A6. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Little blue heron, Egretta caerulea" (131).

Figure A7. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is “Blenheim spaniel” (156).

Figure A8. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Golden retriever" (207).

Figure A9. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Arctic fox, White fox, Alopex lagopus" (279).

Figure A10. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "lesser panda, Red panda, Panda, Bear cat, Cat bear, Ailurus fulgens" (387).

Figure A11. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Balloon' (417).

Figure A12. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Castle' (483).

Figure A13. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Check, Convertible' (511).

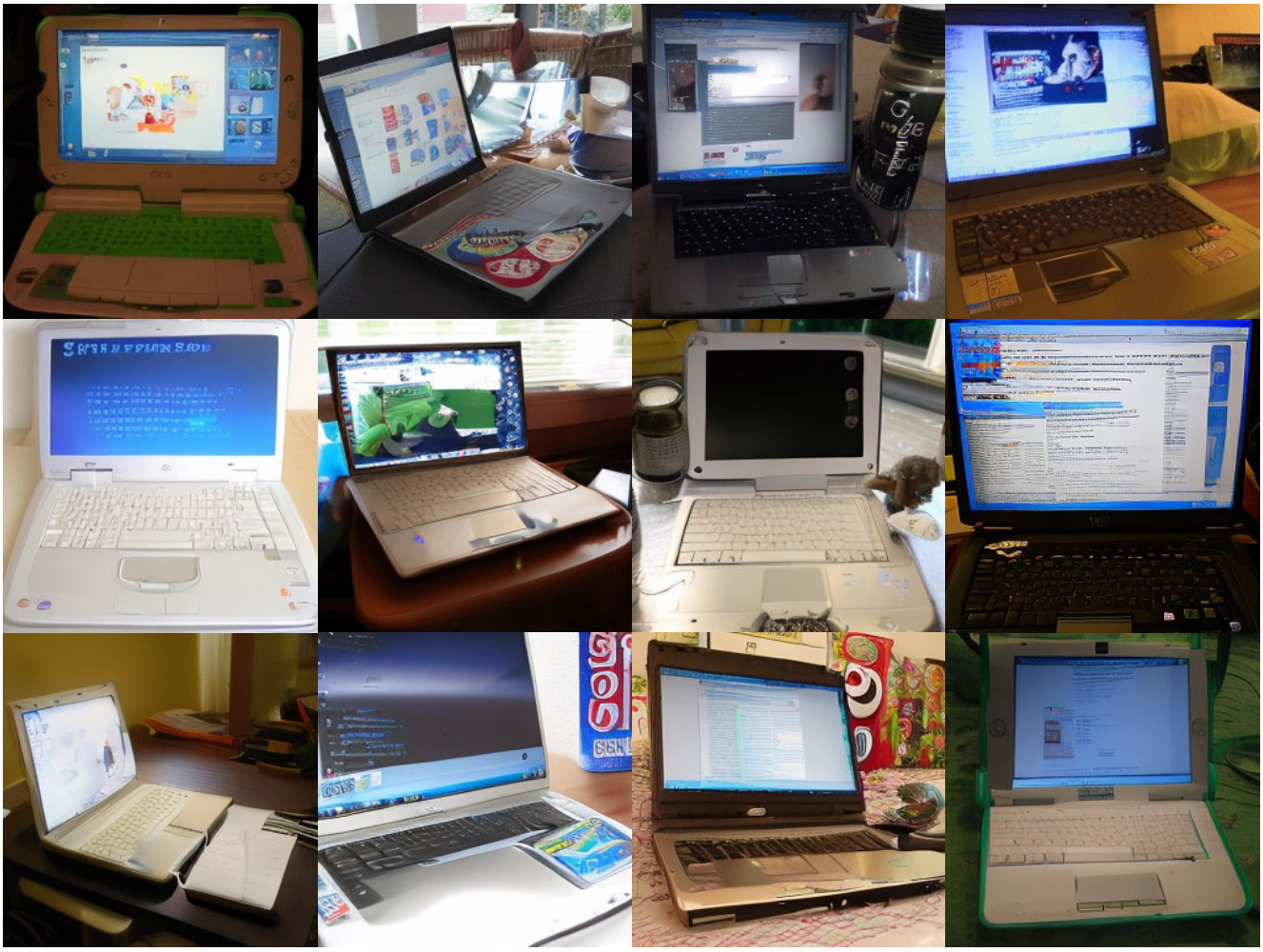

Figure A14. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Laptop, Laptop computer' (620).

Figure A15. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Pillow' (721).

Figure A16. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Check, Streetcar, Tram, Tramcar, Trolley, Trolley car' (829).

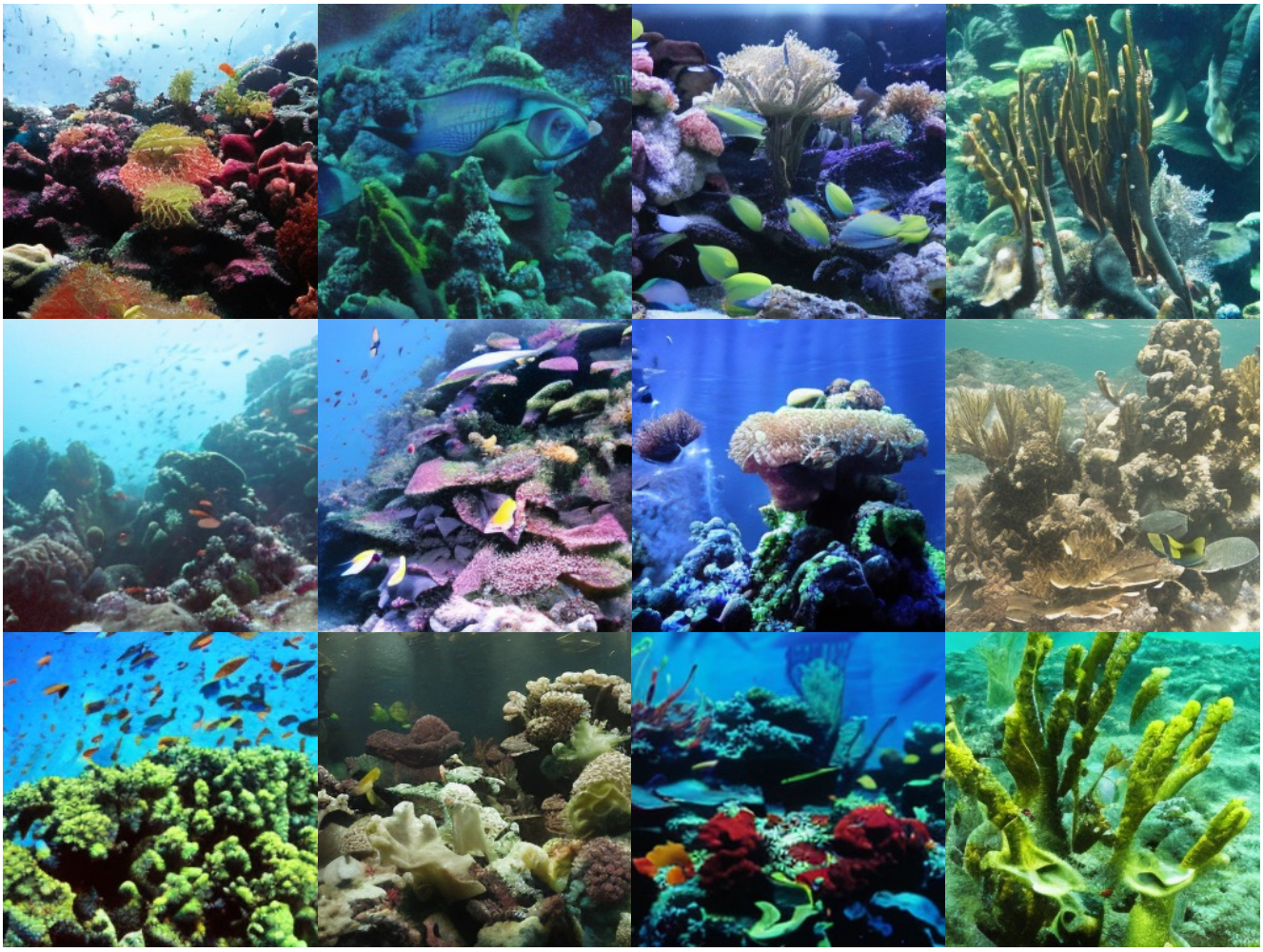

Figure A17. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Coral reef" (973).

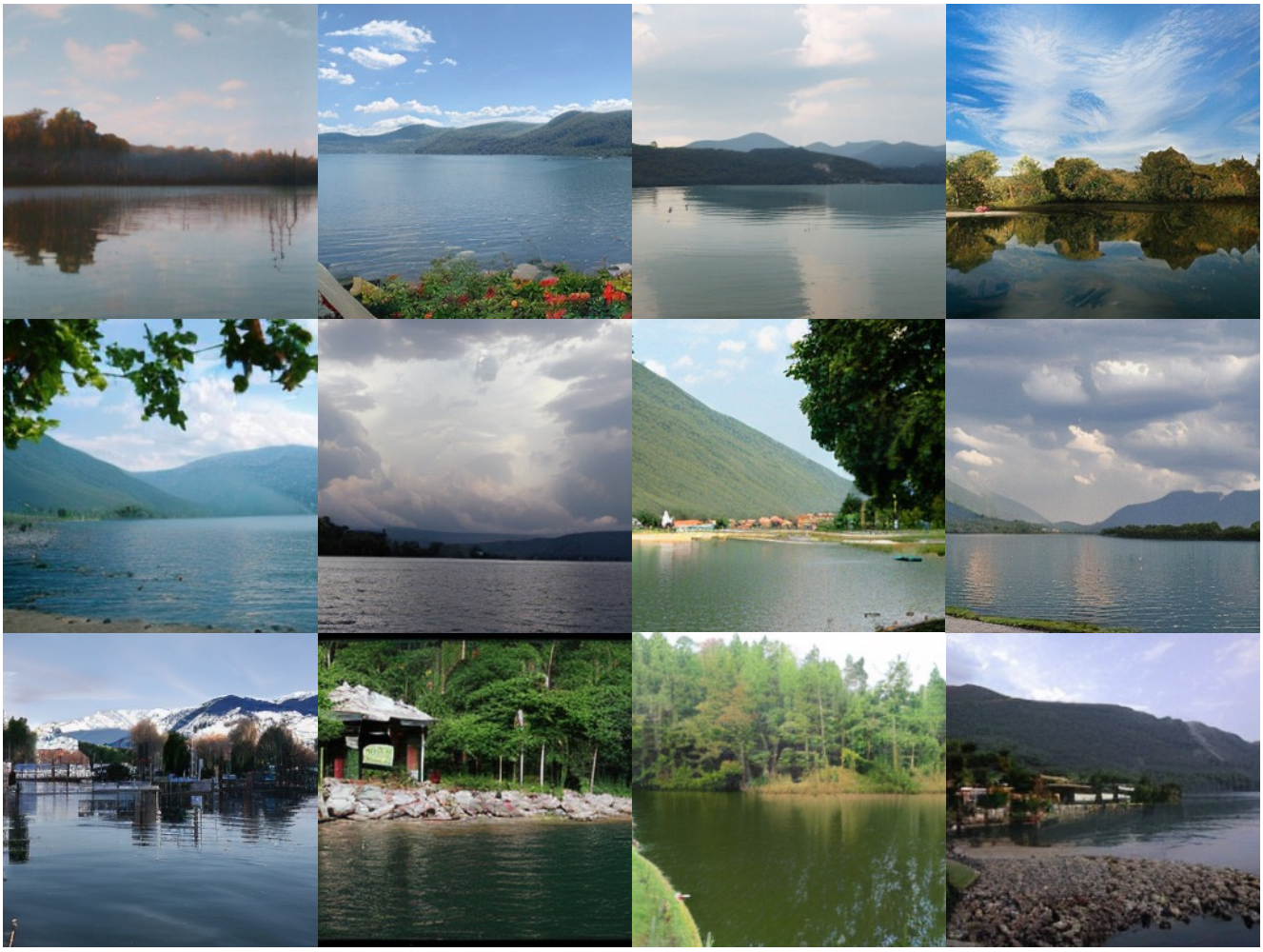

Figure A18. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Lakeside, lakeshore' (975).

Figure A19. **Uncurated generation results of our DiverseDiT-XL on ImageNet 256×256.** We use classifier-free guidance with $w = 4.0$, the lass label is "Volcano' (980).