Anomaly Score: Evaluating Generative Models and Individual Generated Images based on Complexity and Vulnerability

Supplementary Material

In this supplementary material, we include additional materials, which are not contained in the main paper because of the page limit, such as an explanation of the employed generative models and details of the linear regression on vulnerability. We also provide additional experimental results on various feature models and examples of generated images that are used for the subjective test.

A. Employed generative models

We utilize various generated datasets from `https://github.com/layer6ai-labs/dgm-eval` [30], which are listed below with respect to the target image dataset.

- CIFAR10 [20]: ACGAN [24], BigGAN [4], IDDPM [23], LoGAN [34], LSGM [32], MHGAN [12], PFGM++ [35], ReacGAN [15], ResFlow [7], StyleGAN-XL [29], StyleGAN2-ada [18], WGAN [1]
- ImageNet [8]: ADM [10], BigGAN [4], DiT-XL-2 [26], GigaGAN [16], LDM [2], Mask-GIT [6], RQ-Transformer [21], StyleGAN-XL [29], ADMG [9], ADMG-ADMU [9]
- FFHQ [17]: Efficient-vdVAE [19], InsGen [36], LDM [2], StyleGAN-XL [29], StyleGAN2-ada [18], StyleNAT [33], StyleSwin [37], Unleashing-transformers [3], Projected GAN [28]

B. Parameter settings

We examine optimal parameter settings for computing *complexity* and *vulnerability*. Tab. B.1 shows the average *complexity* and *vulnerability* of real and generated datasets with different parameter settings (i.e., ϵ, α, K, J) by using ConvNeXt as a feature model. In Tab. B.1, *complexity* of the generated dataset (by PFGM++) is smaller than that of the real dataset (CIFAR10) and *vulnerability* of the generated dataset is larger than that of the real dataset with parameter changes. The overall tendency of the *complexity* and *vulnerability* is not affected by parameter changes.

		complexity		*vulnerability*	
		real	generated	real	generated
ϵ & α	0.05	0.184	0.181	35.97	36.19
ϵ & α	**0.01**	0.099	0.098	14.57	15.24
ϵ & α	0.005	0.080	0.076	7.77	7.95
K&J=5		0.098	0.069	7.24	7.45

Table B.1. ***Complexity* and *vulnerability* with various parameter settings.** Each cell denotes the average *complexity* or *vulnerability* of the real or generated dataset. In the upper three rows, K and J are fixed as 10. In the last row, ϵ and α are 0.01.

C. Two-tailed test for Tab. 1 and Tab. 2

We report one-tailed tests in Tab. 1 and Tab. 2 of the main paper because we assume that *complexity* and *vulnerability* of generated datasets are smaller than or larger than those of real datasets. For statistical clarity, we show the two-tailed test results in Tab. C.1 on the FFHQ dataset.

D. Linear regression on vulnerability

In Sec 3.2, we explore the motivation of *vulnerability* by calculating the contributions of super-pixels of images to the changes caused by adversarial attacks. We randomly select 3 to 6 super-pixels, add adversarial perturbations into them, and obtain the changes in the features due to the perturbations. We repeat this process 20 times. Then, we apply linear regression between the feature change and the set of binary variables indicating whether each super-pixel is attacked or not. The linear regression is described as: $\mathrm{Y} = \mathrm{V}\mathrm{W} + b$, where Y is a 20 (# of trials)$\times$1 vector of the feature change, V is a 20 (# of trials)$\times$20 (# of super-pixels) matrix of variables that indicate whether each super-pixel is selected or not on each trial, W is a 20 (# of super-pixels)$\times$1 vector of the linear regression coefficient, and b is a 20 (# of trials)$\times$1 vector of bias. We consider the linear

		ViT	ConvNeXt	DINO-V2
Complexity	Reference	0.0643	0.0627	0.0311
	Generated	0.0638	0.0525	0.0302
	p-value	0.4990	<0.0001*	<0.0001*
Vulnerability	Reference	18.30	14.57	12.90
	Generated	19.22	17.21	16.34
	p-value	<0.0001*	<0.0001*	<0.0001*

Table C.1. ***Complexity* and *vulnerability* of FFHQ with two-tailed test.** We compare the average value of *complexity* and *vulnerability* for various feature models, ViT-S [11], ConvNeXt-tiny [22], and DINO-V2 [25]. 'Reference' indicates the original dataset, FFHQ [17]. 'Generated' denotes the *complexity* and *vulnerability* obtained from datasets generated by InsGen [36] trained with FFHQ. 'p-value' denotes the p-value of the two-tailed t-test under the null hypothesis that *complexity* of the generated dataset is equal to that of the reference dataset. The cases with statistical significance are marked with '$*$'.

regression coefficient W as the contribution of each super-pixel to the feature changes, i.e., *vulnerability*. If the coefficient is large, the corresponding super-pixel greatly contributes to the vulnerability. On the other hand, if the coefficient is small, the corresponding super-pixel contributes less to the vulnerability.

E. Results on various feature models

We use six feature models, ResNet50 [13], ViT-S [11], ConvNeXt-tiny [22], CLIP [27], DINO [5], and DINO-V2 [25]. Here, we present additional experimental results on these feature models, which are not included in the main paper.

E.1. Complexity and vulnerability

Tab. E.1 indicates the average values of *complexity* and *vulnerability* of the reference datasets and generated datasets when we use ResNet50, CLIP, and DINO as feature models. In most cases, *complexity* of the generated datasets is smaller than that of the reference datasets. *Vulnerability* of the generated datasets is larger than that of the reference datasets except for a few cases. These results are generally consistent with the results in the main paper (Tab. 1 and Tab. 2). However, in some cases using ResNet50 and DINO, the results are not aligned with our assumption, implying that they are less preferable as the feature model of our method.

E.2. Anomaly score

Fig. E.1 indicates evaluation results of all generative models targeting all image datasets (CIFAR10, ImageNet, and FFHQ) using the proposed AS with various feature models except for DINO-V2. The results with DINO-V2 are shown in Fig. 8 of the main paper.

Complexity		ResNet50	CLIP	DINO
CIFAR10	Reference	0.1900	1.9246	0.0647
	Generated	0.1921	1.9234	0.0626
	p-value	-	0.0853	<0.0001*
ImageNet	Reference	0.1170	1.9543	0.0326
	Generated	0.1190	1.9366	0.0331
	p-value	-	<0.0001*	-
FFHQ	Reference	0.1273	1.9899	0.0424
	Generated	0.1233	1.9893	0.0352
	p-value	<0.0001*	0.1489	<0.0001*

Vulnerability		ResNet50	CLIP	DINO
CIFAR10	Reference	44.59	6.67	37.40
	Generated	44.31	6.69	35.56
	p-value	-	< 0.05*	-
ImageNet	Reference	32.18	4.54	9.27
	Generated	35.58	5.12	11.98
	p-value	<0.0001*	<0.0001*	<0.0001*
FFHQ	Reference	30.22	4.52	13.9
	Generated	30.85	4.39	12.57
	p-value	<0.0001*	-	-

Table E.1. ***Complexity* and *vulnerability* of various datasets.** We compare the average value of *vulnerability* for various feature models, ResNet50 [13], CLIP [27], and DINO [5]. 'Reference' indicates the original dataset, such as CIFAR10 [20], ImageNet [8], and FFHQ [17]. 'Generated' denotes datasets generated by PFGM++ [35], RQ Transformer [21], and InsGen [36] trained with the respective reference datasets. 'p-value' denotes the p-value of the one-tailed t-test under the null hypothesis that *complexity* or *vulnerability* of the generated dataset is equal to that of the reference dataset. The cases with statistical significance are marked with '$*$'. '-' means that the expectation is not met, i.e., *complexity* (*vulnerability*) of the generated dataset is larger (smaller) than that of the reference dataset.

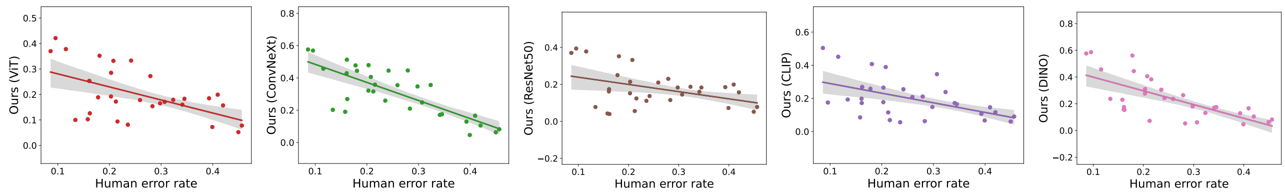
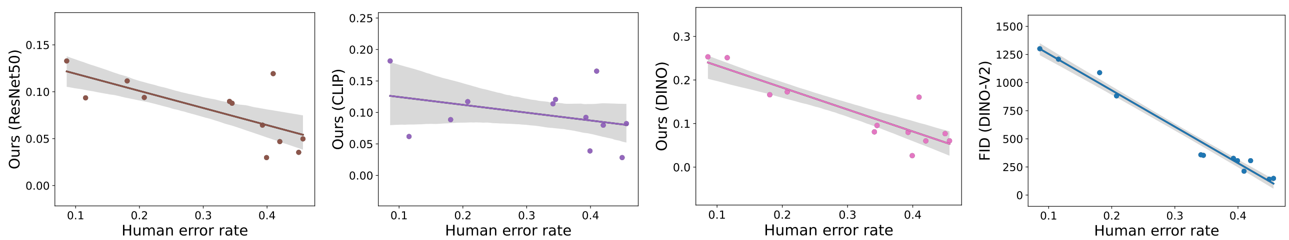
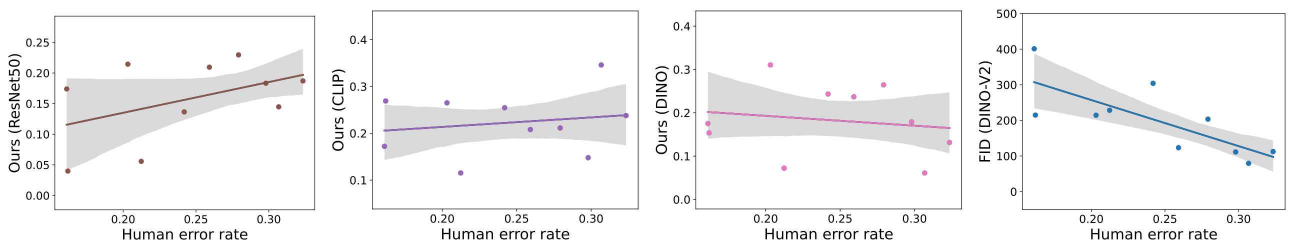


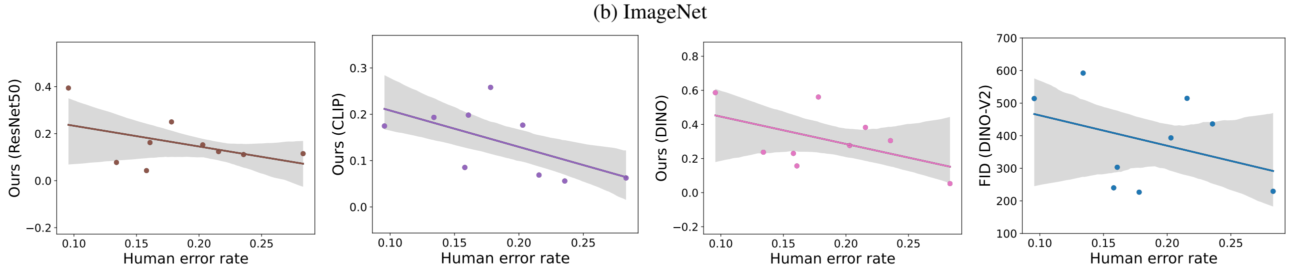
Figure E.1. **Performances of our method using various models for overall datasets.** Each dot represents a distinct dataset generated by a generative model. A high human error rate indicates a high-quality dataset, while a high AS score means a low-quality dataset.



(a) CIFAR10



(b) ImageNet



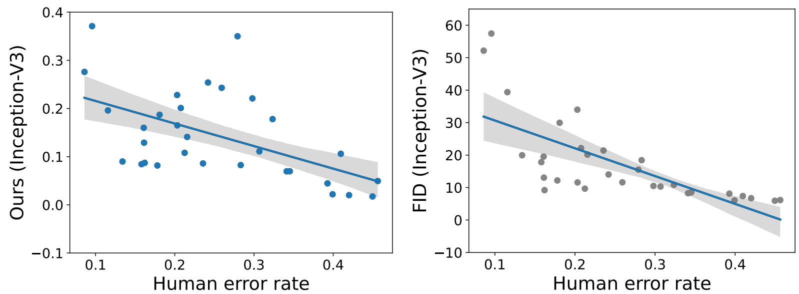
(c) FFHQ

Figure E.2. **Overall results of evaluating generative models on various datasets.** Each dot represents a distinct dataset generated by a generative model. A high human error rate indicates a high-quality dataset, while a high AS score means a low-quality dataset. The first three columns show AS with different feature models: ResNet50, CLIP, and DINO, respectively. The last column is the result of FID [14] with the DINO-V2 model.

Fig. E.2 shows evaluation results of various generative models using AS with ResNet50, CLIP, and DINO as feature models and FID with DINO-V2. In the case of CIFAR10 and FFHQ, AS correlates well with human perception (-0.72, -0.36, and -0.89 pearson correlation coefficients (PCCs) on CIFAR10, and -0.47, -0.60, and -0.51 PCCs on FFHQ, respectively). On the other hand, AS with ResNet50, CLIP, and DINO shows low correlations on generated datasets for ImageNet (0.45, 0.17, and -0.16 PCCs, respectively). Due to the weak alignment between the characteristics of the representation space of ResNet50, CLIP, and DINO and our assumptions (Appendix E.1), the performance of the anomaly score using them is lower than that using ViT-S, ConvNeXt-tiny, and DINO-V2.

F. Comparison with Inception-V3

In Sec. 4 of the main paper, we mainly use DINO-V2 as a feature model for FID since it shows high performance in [30]. Fig. F.1 shows the evaluation results using AS and FID with Inception-V3 [31] as a feature model. Experimental settings for evaluation including used generative models and parameter settings are the same as those of Fig. 8 of the main paper. The



(a) Ours (Inception-V3) (b) FID (Inception-V3)

Figure F.1. **Performances of our method and FID using Inception-V3.** We evaluate various generative models by the proposed method and FID with Inception-V3. Each dot represents a distinct dataset generated by a generative model. A high human error rate indicates a high-quality dataset, while a high AS score means a low-quality dataset.

PCC of ours is -0.54, which has a comparatively weaker correlation than one of our methods using DINO-V2 (-0.81). On the other hand, FID using Inception-V3 shows a comparatively stronger correlation (PCC=-0.71) compared to FID using DINO-V2 (PCC=-0.54). However, FID using Inception-V3 provides poor evaluation performance on generative models targeting ImageNet [30]. Thus, in the main paper, we mainly compare our method with FID using DINO-V2.

G. Transformation of anomaly score

We define anomaly score by comparing the distributions of *complexity* and *vulnerability*. Here, we provide the additional experimental results when we evaluate generative models using the average of individual AS-i. For evaluating each generated dataset targeting FFHQ utilizing ConvNeXt as a feature model, we first compute the individual score, AS-i, of each image and then take the average across the images in the dataset. As shown in Fig. G.1, the average of AS-i does not work well in evaluating generative models. This seems to be because numerical differences in AS-i is limited to capture distributional differences between real and generated datasets.

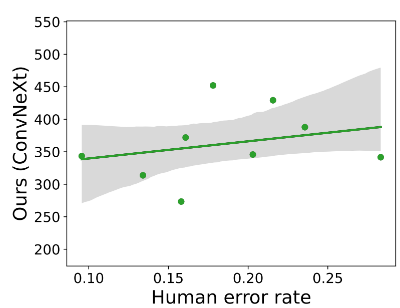


Figure G.1. **Performance of the average of AS-i.** We evaluate generative models targeting FFHQ by the average of AS-i using ConvNeXt as a feature model. Each dot represents a distinct generated dataset. A high human error rate indicates a high-quality dataset, while a high average of AS-i means a low-quality dataset.

H. Images for subjective test

In Sec. 5 of the main paper, we evaluate our anomaly score for individual images, AS-i, by conducting the subjective test with 20 images for each AS-i level. Fig. H.1 shows example images according to each AS-i level. If an image has a low AS-i level, the image looks natural and clear, like real images. Images with higher AS-i levels contain more unnatural components, such as abnormal patterns in faces and backgrounds. Fig. H.1 shows that the severity of the unnatural pattern in the image increases as the AS-i level increases.

Figure H.1. **Examples having various levels of AS-i.**

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 1

[2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 1

[3] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In *ECCV*, 2022. 1

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2018. 1

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2

[6] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 1

[7] Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *NeurIPS*, 2019. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 1

[10] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[12] Kangning Du, Huaqiang Zhou, Lin Cao, Yanan Guo, and Tao Wang. Mhgan: Multi-hierarchies generative adversarial network for high-quality face sketch synthesis. *IEEE Access*, 8:212995–213011, 2020. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*, 2017. 3

[15] Minguk Kang, Woohyeon Shim, Minsu Cho, and Jaesik Park. Rebooting acgan: Auxiliary classifier gans with stable training. *NeurIPS*, 2021. 1

[16] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *CVPR*, 2023. 1

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 1

[19] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014. 1

[20] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, 2009. 1, 2

[21] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, 2022. 1, 2

[22] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 2

[23] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 1

[24] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017. 1

[25] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[26] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 1

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*. PMLR, 2021. 2

[28] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *NeurIPS*, 2021. 1

[29] Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings*, pages 1–10, 2022. 1

[30] George Stein, Jesse C. Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Leigh Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L. Caterini, J. Eric T. Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *arXiv preprint arXiv:2306.04675*, 2023. 1, 3, 4

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *CVPR*, 2016. 3

[32] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *NeurIPS*, 2021. 1

[33] Steven Walton, Ali Hassani, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. StyleNAT: Giving each head a new perspective. *arXiv preprint arXiv:2211.05770*, 2022. 1

[34] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019. 1

[35] Yilun Xu, Ziming Liu, Yonglong Tian, Shangyuan Tong, Max Tegmark, and Tommi Jaakkola. PFGM++: Unlocking the potential of physics-inspired generative models. In *ICML*, 2023. 1, 2

[36] Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. *NeurIPS*, 2021. 1, 2

[37] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *CVPR*, 2022. 1