A. Proof

A.1. Proof of PSR bounded between 0 and 1

Proof. In the main paper, our proposed scoring rule PSR_P is defined as

$$\text{PSR}_P(y_j) = 1 - \prod_{i=1}^{N} \left(1 - \text{Pr}(y_j \in S_P(x_i))\right),$$

where

$$\text{Pr}(y_j \in S_P(x_i)) = \begin{cases} 1 - \frac{||x_i - y_j||_2}{R}, & \text{if } ||x_i - y_j||_2 \leq R \\ 0, & \text{otherwise} \end{cases}$$

and

$$R = \frac{a}{N} \sum_{i=1}^{N} \text{NND}_k(x_i).$$

By the definition of $\text{Pr}(y_j \in S_P(x_i))$, it is bounded between 0 and 1 for all i:

$$0 \leq \text{Pr}(y_j \in S_P(x_i)) \leq 1, \forall_i$$

Thus,

$$0 \leq 1 - \text{Pr}(y_j \in S_P(x_i)) \leq 1, \forall_i$$

Multiplication of N values within [0,1] is also between 0 and 1 by the inequality properties:

$$0 \leq \prod_{i=1}^{N} \left(1 - \text{Pr}(y_j \in S_P(x_i))\right) \leq 1$$

Finally,

$$0 \leq \text{PSR}_P(y_j) = 1 - \prod_{i=1}^{N} \left(1 - \text{Pr}(y_j \in S_P(x_i))\right) \leq 1$$

Therefore, $\text{PSR}_P(y_j)$ is bounded within [0,1]. The same procedure symmetrically applies to $\text{PSR}_Q(x_i)$. ■

B. Related work

Statistical divergence metrics mainly measure the disparity between real and generated image distributions into a single value. For instance, the Inception Score [14] (IS) assesses the quality of generated images, utilizing the label distribution assigned by the InceptionV3 model. High diversity and label confidence in generated images result in a better score. The Frechet Inception Distance [5] (FID) computes the distance between real and generated image distributions in a feature space defined by the Inception network. A lower FID implies that generated images are closer to real ones in this feature space. Meanwhile, the Kernel Inception Distance [2] (KID) refines the concept behind FID by relaxing its Gaussian assumption. Instead, KID calculates the squared Maximum Mean Discrepancy between Inception representations of real and generated samples with a polynomial kernel. Although these metrics are widely adopted due to their alignment with human evaluations, they fall short in providing detailed insights into fidelity and diversity.

Precision and recall metrics Precision and Recall [13, 10] introduced an approach using precision and recall to distinguish between fidelity and diversity of generative models. In this context, precision evaluates how closely generated samples resemble real ones, while recall gauges the generator's capability to reproduce all samples from the training set. Building on this foundation, subsequent research endeavored to address perceived limitations in the precision and recall framework.

Table 1: The participants were asked to choose between two generated images that were sorted by L. Specifically, L_{low} consists of images with high DSR value but low PSR value, while L_{high} consists of images with high PSR value but low DSR.

	PSR vs. DSR	
	L_{low}	L_{high}
Preference Score ↑	17.40%	**82.60%**

Density and Coverage [11] expressed concerns over the contemporary precision and recall metrics, highlighting that they: 1) struggle to recognize a perfect match between identical distributions, 2) are not sufficiently robust in the presence of outliers, and 3) rely on arbitrary evaluation hyperparameters. As a remedy, they introduced Density and Coverage metrics. While precision traditionally examines if a generated sample falls within any neighborhood sphere, Density evaluates the number of real-sample neighborhood spheres encompassing that generated sample. However, in this paper, we have systemically demonstrated that D&C still remain vulnerable to outliers and have conceptual limitation that make Coverage insensitive to distribution change.

In a subsequent contribution, another work [1] proposed a tri-dimensional evaluation measure, encompassing α-Precision, β-Recall, and Authenticity, aimed at delineating the fidelity, diversity, and generalization prowess of generative models. They posit that a fraction $1 - \alpha$ (or $1 - \beta$) of real (or synthetic) samples can be considered as outliers, with the remainder deemed typical. Here, α-Precision represents the proportion of synthetic samples that align with the most typical α real samples. Conversely, β-Recall indicates the proportion of real samples enveloped by the most representative β synthetic samples. Importantly, α-Precision and β-Recall span the entire [0, 1] interval, offering comprehensive precision-recall curves over a singular value. In order to employ their metric, they embed samples into hyperspheres, with a majority of samples clustered around the centers. Although their approach commendably addresses the outlier conundrum by adjusting metrics for varying supports, it relies on specific support modifications. This calls for extra training to create a specific embedding domain, such as a center-heavy hypersphere, culminating in computational challenges and practical constraints, particularly with expansive datasets like ImageNet. Contrarily, our proposed technique is a versatile solution, apt for any feature domain.

C. Comparing scoring rules

C.1. User study

We conducted a user study to evaluate the effectiveness of scoring rules in quantifying the quality of generated images. The study followed a 2-alternative force-choice paradigm as presented in previous work [12]. Participants were presented with two images, one from the set L_{low} which consisted of images with the lowest L values as described in the main paper, and the other from the set L_{high} which contained images with the highest L values. Participants were then asked to select the image they considered to have better quality, i.e., more realistic and with fewer artifacts, based on the presented options. The result presented in Tab. 1 affirms the effectiveness of our proposed scoring rule, PSR, in reflecting the quality of generated images.

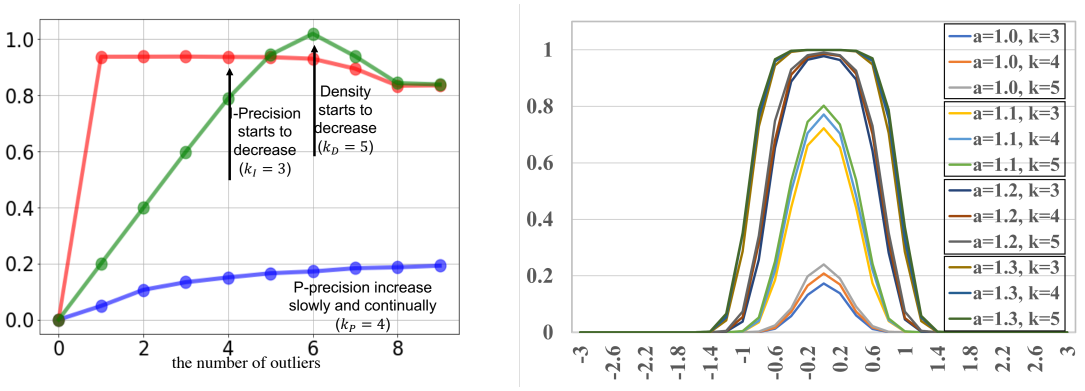
C.2. Additional examples

We provide additional qualitative examples of generated images sorted by condition L (PSR - DSS). Fig. 3 shows the generated images from BigGAN [3] trained on CIFAR-10 [9], sorted by condition L without cherry-picking. Images with the highest L tend to be more realistic than those with the lowest L. Fig. 4 also shows the generated images from StyleGAN [7] trained on FFHQ [7], sorted by condition L also without cherry-picking. Fig. 4a shows images with the highest L whereas Fig. 4b shows images with the lowest L. Most images with high L values are easily recognizable, with simple backgrounds. In contrast, most images with low L values exhibit severe distortions, and their backgrounds are complicated and distorted.

D. Additional experiments

D.1. Hyperparameter selection

Our metric has two defining hyperparameters: a and k. In order to pinpoint hyperparameters that optimize both the robustness and sensitivity of the metric, we revisited the outlier experiment from Sec. 5.1 (in the main paper) for varying a and k, and the results are shown in Fig. 1b. An increase in a diminishes the metric's sensitivity to shifts in distribution, a consequence of our scoring rule becoming more lenient (the slope of the linear function becomes more gradual) with escalating a values. After our analyses, we settled on $a = 1.2$ since it amplifies the metric value when both distributions are identical and



(a)

(b)

Figure 1: (a) Behavior of metrics between $X \sim N(0, I)$ and $Y \sim N(0, vI)$ as v changes between [0.2, 1.5]. Because Density goes over 1 (up to nearly 1000), the y-axis for Density is on the right side of the plot for better visualization. (b) Ablation study for selecting hyperparameter a and k.

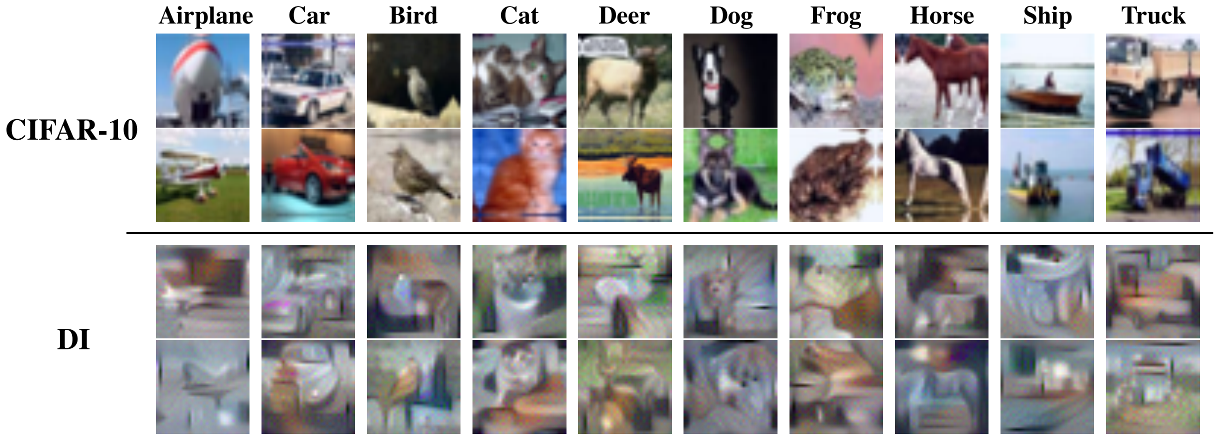


Figure 2: Qualitative examples of CIFAR-10 [9] and the images generated by DI [15].

avoids saturation near $u = 0$. The choice of k doesn't profoundly sway the metric's sensitivity, which is also illustrated in Fig. 3 of the main paper. Therefore, we opted for $k = 4$, a value intermediate between the choices for P&R and D&C.

D.2. Outlier experiment with real-world distribution

We further investigate the robustness of metrics by using CIFAR-10 [9] and DeepInversion (DI) model [15] trained for CIFAR-10. DI is an inversion model trained to generate images that match the pre-trained neural network's intermediate feature statistics without explicitly observing the source images. Therefore, DI generates images that are somewhat distorted and unrecognizable (See Fig. 2), yet preserves the statistics of CIFAR-10. We split generated images from DI into two subsets: inliers and outliers. We identify the outliers by computing the average of the k_{th} nearest distance among real samples and selecting generated images whose nearest distance from real samples is greater than the average. Then, we substitute the real samples with these outliers. In Fig. 1, we present the metric increments with respect to the number of outliers. We compare the increment of each metric with respect to the case where there are no outliers. The result shows that I-precision increases by about 0.9 with just one outlier, indicating its vulnerability to outliers. In addition, with only five outliers, Density shows a significant increase and eventually exceeds the increment of I-precision. This is because DSR accumulates the constant-density of hyperspheres, as we discussed in the main paper. Furthermore, we observe that I-precision and Density suddenly decrease as the number of outliers exceeds their hyperparameter k. This is not desirable behavior for a reliable and consistent metric. The sudden drop occurs because the size of the kNN hypersphere can change dramatically as the number of outliers exceeds k,

Table 2: Quantitative result of various generative models on real-world datasets. The reported values are obtained by measuring each metric five times and taking the average. Bolded values indicates the top scores evaluated by the corresponding metric.

Model	FID↓	PP↑	PR↑	F_1↑	IP↑	IR↑	F_1↑	D↑	C↑	F_1↑
ImageNet 256x256										
ADM [4]	4.95	0.538 (1.0e-5)	**0.732** (1.2e-5)	0.621	0.681 (3.4e-5)	**0.688** (2.5e-4)	0.684	1.52 (6.1e-3)	0.876 (1.1e-5)	1.112
ADM-G [4]	**4.58**	0.699 (3.1e-5)	0.587 (1.4e-5)	**0.638**	0.818 (7.4e-5)	0.606 (1.7e-5)	**0.696**	2.071 (1.4e-3)	**0.956** (2.4e-5)	1.307
BigGAN [3]	8.12	**0.751** (3.4e-5)	0.465 (2.8e-5)	0.574	**0.874** (5.4e-5)	0.403 (2.1e-5)	0.551	**2.481** (3.4e-3)	0.945 (2.9e-5)	**1.368**
AFHQv2 512x512										
StyleGAN2 [8]	4.62	**0.568** (1.6e-5)	0.689 (3.6e-5)	0.623	**0.716** (1.4e-4)	0.494 (3.1e-5)	0.584	**1.886** (4.4e-3)	0.790 (1.9e-5)	**1.113**
StyleGAN3-R [6]	4.40	0.564 (9.4e-6)	0.725 (1.4e-5)	0.634	0.685 (6.4e-5)	**0.591** (3.7e-5)	**0.635**	1.576 (5.1e-4)	0.770 (7.1e-5)	1.034
StyleGAN3-T [6]	**4.04**	0.567 (6.2e-5)	**0.727** (5.1e-5)	**0.637**	0.699 (1.6e-4)	0.578 (8.1e-5)	0.632	1.624 (1.1e-3)	**0.792** (1.2e-5)	1.065
LSUN Bedroom 256x256										
DDPM [8]	4.88	0.799 (4.8e-5)	0.749 (1.4e-5)	0.773	0.606 (2.2e-4)	0.444 (1.4e-4)	0.512	1.701 (3.4e-3)	0.977 (6.4e-5)	1.241
ADM [4]	**1.91**	**0.839** (8.8e-5)	0.731 (6.4e-5)	**0.781**	**0.659** (5.8e-5)	0.494 (3.4e-5)	**0.565**	**1.929** (2.8e-3)	**0.993** (1.2e-5)	**1.311**
StyleGAN2 [8]	2.35	0.801 (2.1e-5)	**0.753** (1.8e-5)	0.776	0.591 (1.9e-5)	**0.501** (2.7e-5)	0.543	1.732 (8.6e-4)	0.986 (2.9e-5)	1.257
MetaFaces 1024x1024										
StyleGAN2 [8]	15.22	0.896 (7.8e-5)	0.703 (6.7e-5)	0.788	**0.797** (4.4e-4)	0.291 (1.4e-4)	0.426	**2.171** (3.8e-3)	0.996 (1.1e-5)	**1.366**
StyleGAN3-R [6]	**15.11**	0.890 (3.6e-5)	**0.770** (2.9e-5)	**0.825**	0.738 (1.4e-4)	0.461 (1.6e-5)	0.567	1.864 (2.3e-3)	0.992 (3.4e-5)	1.295
StyleGAN3-T [6]	15.33	**0.901** (3.4e-5)	0.747 (2.5e-5)	0.817	0.748 (8.1e-5)	**0.478** (5.4e-5)	**0.583**	1.885 (9.7e-4)	**0.997** (1.1e-5)	1.304
ImageNet 64x64										
ADM [4]	**2.60**	0.738 (5.1e-5)	**0.734** (3.4e-5)	**0.736**	0.734 (1.1e-5)	**0.647** (3.4e-5)	**0.688**	1.867 (2.6e-3)	**0.956** (5.4e-5)	1.265
BigGAN [3]	4.07	**0.776** (5.4e-5)	0.612 (7.2e-5)	0.684	**0.792** (8.9e-5)	0.531 (2.1e-5)	0.635	**2.360** (4.4e-3)	0.954 (1.4e-5)	**1.359**
ImageNet 128x128										
ADM [4]	**5.91**	0.601 (5.8e-5)	**0.737** (4.9e-5)	**0.662**	0.700 (8.4e-5)	**0.697** (7.2e-5)	**0.699**	1.613 (9.1e-4)	0.925 (3.4e-5)	1.176
BigGAN [3]	6.01	**0.772** (1.1e-5)	0.473 (2.3e-5)	0.586	**0.862** (3.6e-5)	0.452 (1.6e-5)	0.593	**2.521** (1.7e-3)	**0.954** (3.8e-5)	**1.384**

highlighting the susceptible property of instance-specific kNN. On the other hand, P-precision is less affected by the outlier and increases linearly, demonstrating robustness to both outliers and k.

D.3. Evaluating generative models

Here, we provide additional quantitative results of the state-of-the-art generative models trained on various datasets using existing metrics. We use officially pre-trained models from their official codes[1][2] and take FID scores from their papers. The result of FID, IP&IR, D&C, and our PP&PR is reported in Tab. 2. For all metrics except FID, we measure the metrics between 50K generated samples and all available real samples, but up to 10K, as recommended in [4].

References

[1] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, pages 290–306. PMLR, 2022. 2

[2] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 1

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *International Conference on Learning Representations*, 2019. 2, 4, 6

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 4

[5] 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. *Advances in neural information processing systems*, 30, 2017. 1

[6] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34, 2021. 4

[1] https://github.com/openai/guided-diffusion

[2] https://github.com/NVlabs/stylegan3

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2, 7

[8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 4

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2, 3, 6

[10] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019. 1

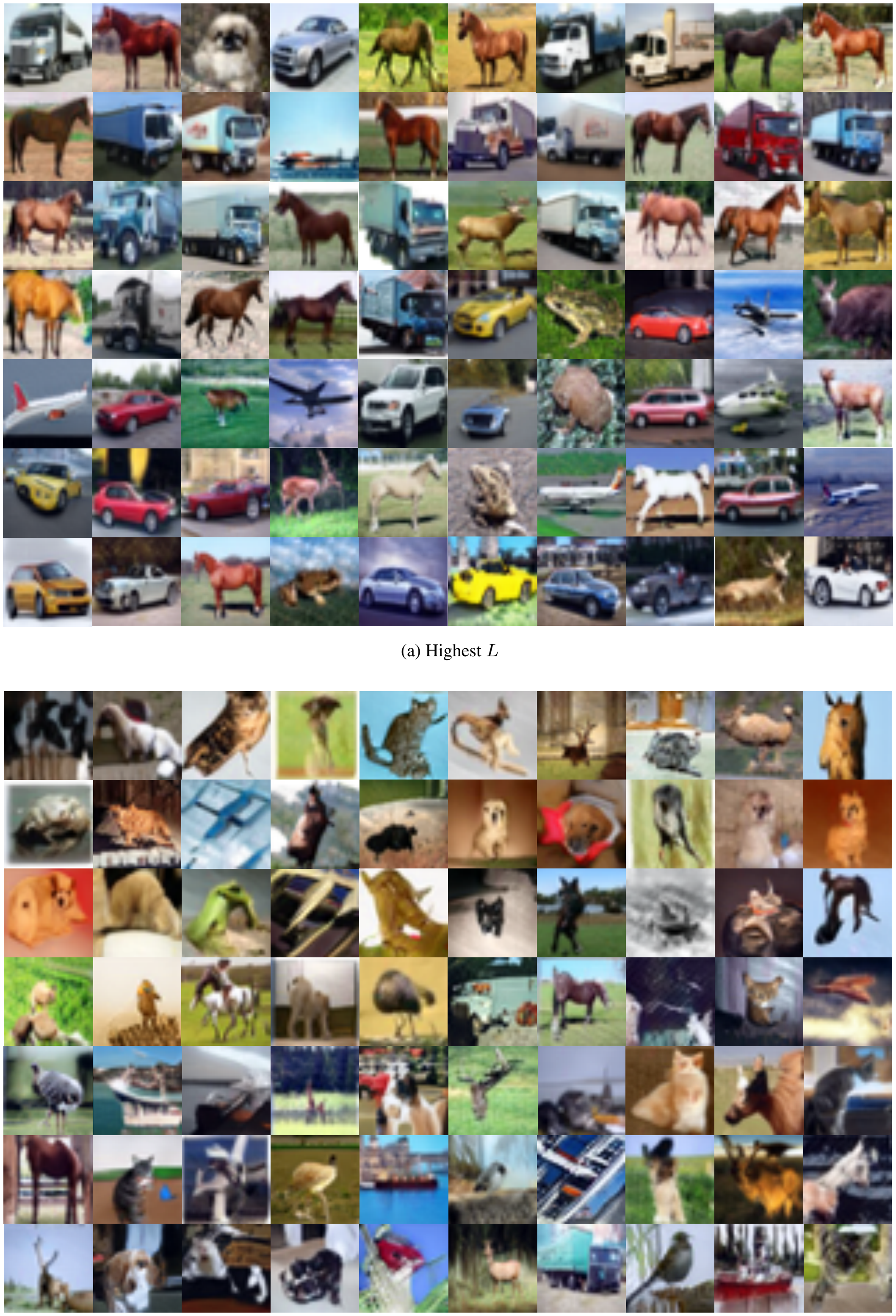
[11] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020. 2

[12] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[13] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018. 1

[14] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 1

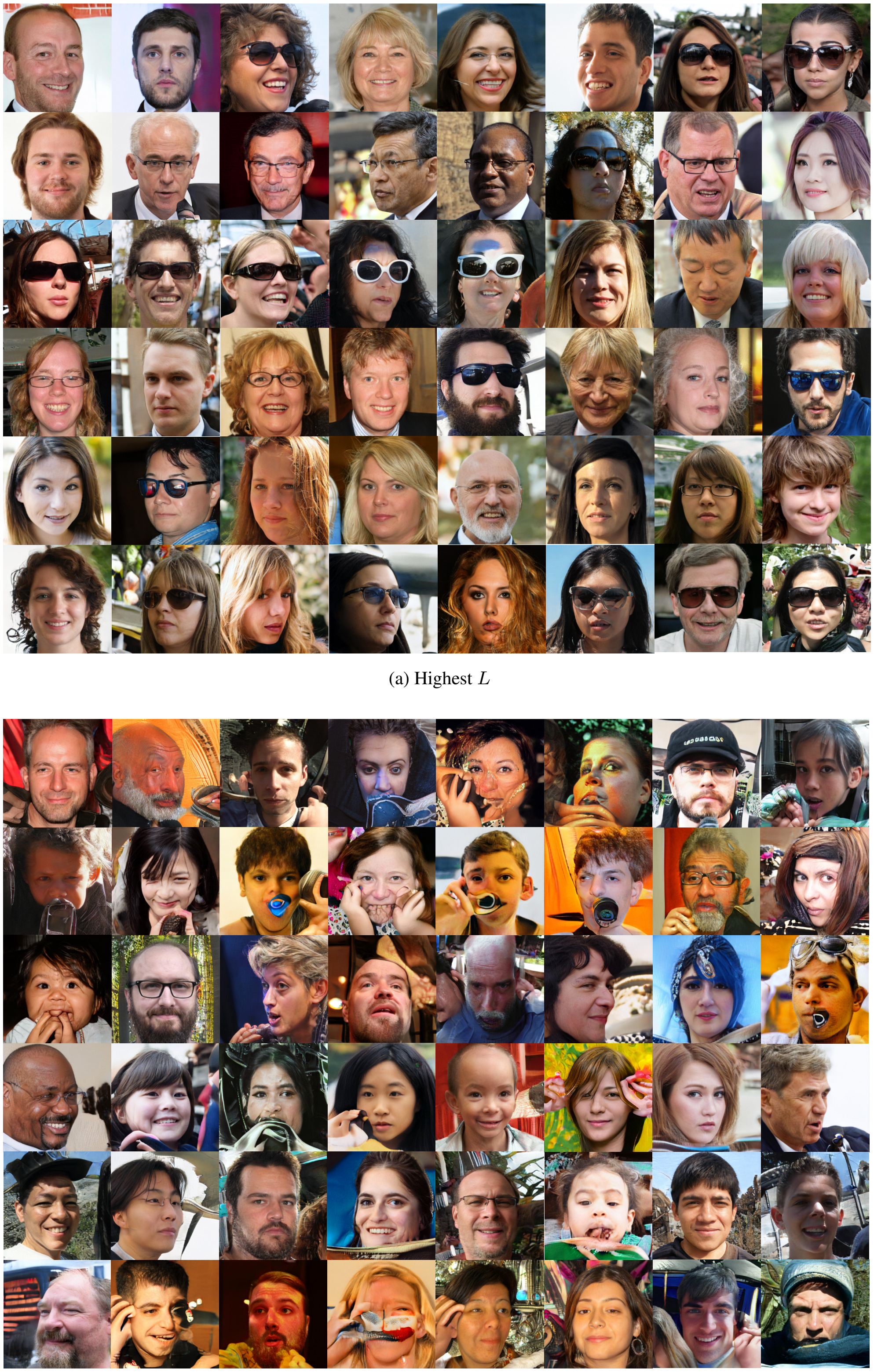
[15] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 3



(a) Highest L

(b) Lowest L

Figure 3: Qualitative examples of BigGAN [3] generated images on CIFAR-10 [9] sorted according to L (PSR - DSR). (a) are images with highest L and (b) are images with lowest L.



(a) Highest L

(b) Lowest L

Figure 4: Qualitative examples of StyleGAN [7] generated images on FFHQ [7] sorted according to L (PSR - DSR). (a) are images with highest L and (b) are images with lowest L.