Understanding Noise Injection in GANs
Supplementary Material

Ruili Feng [1] **Deli Zhao** [2] **Zheng-Jun Zha** [† 1]

A. Dimension drop in GANs

Here we validate Lemma 1 and Lemma 2 empirically. Specifically, we want to make the following two questions clear.

1. Does the Jacobian rank decrease as the network gets deeper?
2. Does the intrinsic dimension of feature space decrease as the network gets deeper?

However, those two things are not easy to validate. Estimating the Jacobian rank of complicated function couplings and estimating the dimension of complicated data manifolds are open questions in data science. For that reason, we are only able to conduct the estimation to those simple structures, such as linear functions and manifolds produced by them.

In what follows, we conduct the estimation to the first eight dense layers of StyleGAN2 on FFHQ. Those layers have sufficiently simple structures for PCA, but also play vital role in the network as discussed in (Karras et al., 2019b;a).

Each of the eight dense layers, denoted as Dense0, Dense1,...,Dense7, is composed of a linear transformation $l_i(x) = W_i x + b_i$ and a LeakyRelu activation

$$act(x) = \left\{ \begin{array}{ll} x & x > 0 \\ 0.2x & x \le 0. \end{array} \right. \tag{S1}$$

Thus the function coupling that maps input to the output of the k-th layer is

$$Dense_k(x) = act(l_k) \circ \cdots \circ act(l_0). \tag{S2}$$

We conduct PCA to the Jacobian of each dense layer. The results are reported in Fig. S1. We compute the number of components that have strength larger than 1% of that of the maximum component. The results are reported in Tab. S1. We can find that as layer gets deeper, the component strengths gather towards the left components, which means the number of valid components gets smaller and the rank of corresponding Jacobian gets smaller. This means the rank drop indicated by Lemma 2 does happen in practice.

Table S1. Estimated rank of Jacobian.

Layer	Rank
Dense0	445
Dense1	351
Dense2	276
Dense3	22
Dense4	174
Dense5	138
Dense6	110
Dense7	82

Table S2. Estimated rank of Weight matrices.

Layer	Rank
W_0	445
W_1	447
W_2	449
W_3	448
W_4	447
W_5	446
W_6	446
W_7	445

In fact, we find that each weight matrix W_i has a low rank structure. We conduct PCA to the weight matrix $W_0, \dots, W_7$. The results are reported in Fig. S2. We compute the number of components that have strength larger than 1% of that of the maximum component. The results are reported in Tab. S2. We can find that the valid components of each weight matrix are around 450, which means each of the weight matrix will drop around 60 dimensions of the inputs.

We then look into the intrinsic dimension of the feature spaces produced by those dense layers. For each dense layer, we sample 51200 random inputs z from $\mathcal{N}(0, 1)$ and feed them to the layer to produce 51200 points in the corresponding feature space. We then conduct PCA to those

† Corresponding author [1]University of Science and Technology of China, Hefei, China. [2]Alibaba Group. Correspondence to: Ruili Feng <ruilifengustc@gmail.com>, Deli Zhao <zhaodeli@gmail.com>, Zhen-Jun Zha <zhazj@ustc.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

Table S3. Estimated intrinsic dimension of intermediate feature space.

Layer	Intrinsic dimension
Dense0	512
Dense1	508
Dense2	422
Dense3	289
Dense4	207
Dense5	148
Dense6	105
Dense7	91

points. The results are reported in Fig. S3. We compute the number of components that have strength larger than 1% of that of the maximum component. The results are reported in Tab. S3. We can find that as layer gets deeper, the component strengths gather towards the left components, which means the number of valid components gets smaller and the intrinsic dimension of the corresponding feature space gets smaller.

In conclusion, we validate that the dimension drop does happen in practice, which supports the condition to induce the adversarial dimension trap in practice.

B. Proof to theorems

B.1. Lemma 1

Proof. Lemma 1 is a natural extension of Sard's Theorem and the rank theorem on manifolds (Petersen et al., 2006).

Lemma C (Sard's Theorem)**.** *Let $f : \mathcal{N} \to \mathcal{M}$ be smooth functions between smooth manifolds $\mathcal{N}$ and $\mathcal{M}$. Define the set of critical points of f as*

$$C_f = \{z \in \mathcal{N} : rank(J_z f) < dim(\mathcal{N})\}. \tag{S3}$$

Then $f(C_f)$ has zero measure in $\mathcal{M}$.

Lemma D (Rank Theorem)**.** *Suppose that $\mathcal{M}$ and $\mathcal{N}$ are smooth manifolds of dimensions m and n, and $f : \mathcal{N} \to \mathcal{M}$ with $f(\mathcal{N}) = \mathcal{M}$ is a smooth mapping with constant rank r. For each $z \in \mathcal{N}$, there exists a smooth char (U, ϕ) around z and a smooth chart (V, ψ) around $f(z)$ such that $f(U) \subset V$, and*

$$\psi \circ f \circ \phi^{-1}(a_1, \dots, a_n) = (a_1, \dots, a_r, 0, \dots, 0). \tag{S4}$$

Let $r = \max_{z \in \mathcal{N}} J_z f$, and $\mathcal{R} = \{z \in \mathcal{N} : rank_z f = r\}$. Then Lemma D says that $f(\mathcal{R})$ and $\mathcal{N}$ have intrinsic dimension r, and $\mathcal{N} \setminus \mathcal{R}$ belongs to the set of critical points. By Lemma B, $f(\mathcal{N} \setminus \mathcal{R})$ is a zero measure set. Thus for almost every point $x \in \mathcal{M}$, its preimage has rank r. □

D.1. Lemma 2

By the chain rule of differential (Rudin et al., 1964), we have

$$rank(J(f^1 \circ f^2)) = rank(Jf^1 Jf^2). \tag{S5}$$

Recall that for any two matrices A and B

$$rank(AB) \leq \min\{rankA, rankB\}. \tag{S6}$$

We then have Lemma 2.

D.2. Theorem 1

Proof. Denote the dimensions of $G(\mathcal{Z})$ and $\mathcal{X}$ as d_g and d_x, respectively. There are two possible cases for G: d_g is lower than d_x, or d_g is higher than or equal to d_x.

For the first case, a direct consequence is that, for almost all points in $\mathcal{X}$, there are no pre-images under G. This means that for an arbitrary point $x \in \mathcal{X}$, the possibility of $G^{-1}(x) = \emptyset$ is 1, as $\{x \in \mathcal{X} : G^{-1}(x) \neq \emptyset\} \subset G(\mathcal{Z}) \cap \mathcal{X}$, which is a zero measure set in $\mathcal{X}$. This also implies that the generator is unable to perform inversion. Another consequence is that, the generated distribution P_g can never get aligned with real data distribution P_r. Namely, the distance between P_r and P_g cannot be zero for arbitrary distance metrics. For the KL divergence, the distance will even approach infinity.

Specifically, let p_r and p_g be the densities of P_r and P_g respectively. For the Jensen-Shannon divergence, we have

$$\begin{aligned} D_{JS}(P_r, P_g) = \frac{1}{2} \int \log\left(\frac{2p_r}{p_r + p_g}\right) p_r \\ + \frac{1}{2} \int \log\left(\frac{2p_g}{p_r + p_g}\right) dp_g. \end{aligned} \tag{S7}$$

As the support of p_g is a zero measure set of the support of p_r, we have

$$\begin{aligned} \int \log\left(\frac{2p_r}{p_r + p_g}\right) dP_r &= \int_{p_g=0} \log\left(\frac{2p_r}{p_r + p_g}\right) dP_r \\ = \int_{p_g=0} \log\left(\frac{2p_r}{p_r}\right) dP_r &= \log 2 \int_{p_g=0} dP_r = \log 2, \end{aligned} \tag{S8}$$

and

$$\int \log\left(\frac{2p_g}{p_r + p_g}\right) dP_g \geq 0. \tag{S9}$$

Thus $D_{JS} \geq \frac{\log 2}{2}$.

For the second case, $d_g \geq d_x > d_{\mathcal{Z}}$. We simply show that a Lipschitz-continuous function cannot map zero measure set into positive measure set. Specifically, the image of low dimensional space of a Lipschitz-continuous function has measure zero. Thus if $d_g \geq d_x$, G cannot be Lipschitz. As Lipschitz constant is the supremum of gradient norm, we then prove our theorem.

Now we prove our claim.

Suppose that $f : \mathbb{R}^n \to \mathbb{R}^m, n < m$, and f is Lipschitz with Lipschitz constant L. We show that $f(\mathbb{R}^n)$ has measure zero in $\mathbb{R}^m$. As $\mathbb{R}^n$ is a zero measure subset of $\mathbb{R}^m$, by the Kirszbraun theorem (Deimling, 2010), f has an extension to a Lipschitz function of the same Lipschitz constant on $\mathbb{R}^m$. For convenience, we still denote the extension as f. Then the problem reduces to proving that f maps zero measure set to zero measure set. For every $\epsilon > 0$, we can find countable union of balls $\{B_k\}_k$ of radius r_k such that $\mathbb{R}^n \subset \cup_k B_k$ and $\sum_k m(B_k) < \epsilon$ in $\mathbb{R}^m$, where $m(\cdot)$ is the Lebesgue measure in $\mathbb{R}^m$. But $f(B_k)$ is contained in a ball with radius Lr_k. Thus we have $m(f(\mathbf{R^n})) \leq L^m \sum_k m(B_k) < L^m \epsilon$, which means that it is a zero measure set in $\mathbb{R}^m$. For the mapping between manifolds, using the chart system can turn it into the case we analyze above, which completes our proof. □

We want to remind the readers that, even if the generator suits one of the cases in Theorem 1, the other case can still occur. For example, G could succeed in capturing the distribution of certain parts of the real data, while it may fail in the other parts. Then for the pre-image of those successfully captured data, the generator will not have finite Lipschitz constant.

D.3. Theorems 2 & 3

Proof. Theorems 2 & 3 are classical conclusions in Riemannian manifold. We refer readers to section 5.5 of the book written by Petersen et al. (2006) for detailed proofs and illustration. □

D.4. Theorem 4

Proof. Theorem 4 is a natural extension of the Heine-Borel theorem (Rudin et al., 1964).

Lemma E (Heine-Borel Theorem)**.** *For any compact set $\mathcal{M}$, if $\{U_i\}_{i\in I}$ is an open cover of $\mathcal{M}$, (that is, for each $i \in I$, U_i is an open set, and $\mathcal{M} \subset \cup_{i\in I} U_i$), then there exist finite many elements $U_{i_1}, ..., U_{i_k}$ of $\{U_i\}_{i\in I}$, such that $\mathcal{M} \subset \cup_{1\leq j\leq k} U_{i_j}$.*

Let the skeleton set be all points of $\mathcal{M}$. Then the representative pairs in Theorems 2 & 3 define an open cover of $\mathcal{M}$. By Lemma E, we can pick finite many points of skeleton set $\mu_1, ..., \mu_k$, such that their representative pairs also define an open cover of $\mathcal{M}$.

For each local neighborhood of representative pairs, it is easy to see that the error is $o(r)$ by Taylor expansion of Theorem 3.

□

E.1. Theorem 5

Proof.

$$\begin{aligned}\mathbf{E}[\|g(x) - g(y)\|_2] &\leq \|\mu(x) - \mu(y)\|_2 \\ &+ \mathbf{E}[\|\sigma(x)\epsilon - \sigma(y)\delta\|_2] \\ &\leq L_\mu \|x - y\|_2 + 2C\|\sigma\|_\infty \\ &\leq L_\mu \|x - y\|_2 + o(1),\end{aligned} \tag{S10}$$

where C is a constant related to the dimension of the image space of σ and L_μ is Lipschitz constant of μ. □

F. Why Gaussian distribution?

We first introduce the notion of fuzzy equivalence relations (Zhang & Zhang, 2005; Murali, 1989).

Definition 1. *A t-norm is a function $T : [0,1] \times [0,1] \to [0,1]$ which satisfies the following properties:*

1. *Commutativity: $T(a,b) = T(b,a)$.*
2. *Monotonicity: $T(a,b) \leq T(c,d)$, if $a \leq c$ and $b \leq d$.*
3. *Associativity: $T(a, T(b,c)) = T(T(a,b),c)$.*
4. *The number 1 acts as identity element: $T(a,1) = a$.*

Definition 2. *Given a t-norm T, a T-equivalence relation on a set X is a fuzzy relation E on X and satisfies the following conditions:*

1. *$E(x,x) = 1, \forall x \in X$ (Reflexivity).*
2. *$E(x,y) = E(y,x), \forall x,y \in X$ (Symmetry).*
3. *$T(E(x,y), E(y,z)) \leq E(x,z)\ \forall x,y,z \in X$ (T-transitivity).*

Then it is easy to check that $T(x,y) = xy$ is a t-norm, and $E(x,y) = e^{-d(x,y)}$ is a T-equivalence for any distance metric d on X, as

$$T(E(x,y), E(y,z)) = e^{-(d(x,y)+d(y,z))} \tag{S11}$$

$$\leq e^{-d(x,z)} = E(x,z). \tag{S12}$$

Considering that we want to contain the fuzzy semantics of real world data in our local geometries of feature manifolds, a natural solution will be that we sample points from the local neighborhood of μ with different densities on behalf of different strengths of semantic relations with μ. Points with stronger semantic relations will have larger densities to be sampled. A good framework to model this process is the fuzzy equivalence relations we mention above, where the degrees of membership E are used as the sampling density. However, our expansion of the exponential map Exp_μ carries an error term of $o(\|v\|_2)$. We certainly do not want the local error to be out of control, and we also wish to constrain the sampling locally. Thus we accelerate the decrease of density when points depart from the center μ, and constrain the integral of E to be identity, which turns E to the density of standard Gaussian.

G. Datasets

FFHQ Flickr-Faces-HQ (FFHQ) (Karras et al., 2019a) is a high-quality image dataset of human faces, originally created as a benchmark data for generative adversarial networks (GANs). The dataset consists of 70,000 high-quality PNG images and contains considerable variations in terms of age, pose, expression, hair style, ethnicity and image backgrounds. It also covers diverse accessories such as eyeglasses, sunglasses, hats, etc.

LSUN-Church and Cat-Selected LSUN-Church is the church outdoor category of LSUN dataset (Yu et al., 2015), which consists of 126 thousand church images of various styles. Cat-Selected contains 100 thousand cat images selected by ranking algorithm (Zhou et al., 2004) from the LSUN cat category. The plausibility of using PageRank to rank data was analyzed in (Zhou et al., 2004). We also used the algorithm presented in (Zhao & Tang, 2009) to construct the graph from the cat data.

CIFAR-10 The CIFAR-10 dataset (Krizhevsky et al., 2009) consists of 60,000 images of size 32x32. There are all 10 classes and 6000 images per class. There are 50,000 training images and 10,000 test images.

H. Implementation details

H.1. Models

We illustrate the generator architectures of StyleGAN2 based methods in Figure S4. For all those models, the discriminators share the same architecture as the original StyleGAN2. The generator architecture of DCGAN based methods are illustrated in Figure S5. For all those models, the discriminators share the same architecture as the original DCGAN.

I. Experiment environment

All experiments are carried out by TensorFlow 1.14 and Python 3.6 with CUDA Version 10.2 and NVIDIA-SMI 440.64.00. We basically build our code upon the framework of NVIDIA official StyleGAN2 code, which is available at `https://github.com/NVlabs/stylegan2`. We use a variety of servers to run the experiments as reported in Table S4.

J. Image encoding and GAN inversion

From a mathematical perspective, a well behaved generator should be easily invertible. In the last section, we have shown that our method is well conditioned, which implies that it could be easily invertible. We adopt the methods in Image2StyleGAN (Abdal et al., 2019) to perform GAN inversion and compare the mean square error and perceptual loss on a manually collected dataset of 20 images. The source code of inversion is from Luxemburg (2020). The images are shown in Figure S6 and the quantitative results are provided in Table S5. For our RNI methods, we further optimize the α parameter in Eq. 7 in section 4.3, which finetunes the local geometries of the network to suit the new images that might not be settled in the model. Considering that α is limited to $[0, 1]$, we use $\frac{(\alpha^*)^t}{(\alpha^*)^t+(1-\alpha^*)^t}$ to replace the original α and optimize t. The initial value of t is set to 1.0 and α^* is constant with the same value as α in the converged RNI models.

During the experiments, we find that the StyleGAN2 model is prone to work well for full-face, non-blocking human face images. For this type of images, we observe comparable performance for all the GAN architectures. We think that this is because those images are closed to the 'mean' face of FFHQ dataset (Karras et al., 2019a), thus easy to learn for the StyleGAN based models. For faces of large pose or partially blocked ones, the capacity of different models differs significantly. Noise injection methods outperform the bald StyleGAN2 by a large margin, and our method achieves the best performance.

K. Ablation study of RNI

In Tab. S6, we perform the ablation study of the proposed RNI method on the FFHQ dataset. We test 5 different choices of RNI implementation and compare their FID and PPL scores after convergence.

1. No normalization: in this setting we remove the normalization of $\tilde{\mu}$ in Eq. (14), and use the unnormalized $\tilde{\mu}$ to replace s in the following equations. The network comes to a minimum FID of 23.77 after training on 1323 thousand images, and then quickly falls into mode collapse after that.

2. No stabilization: in this setting we remove the stabilization technique in Eq. (16). The network comes to a minimum FID of 50.27 after training on 963 thousand images, and then quickly falls to mode collapse after

Table S4. GPU environments for all experiments in this work.

Experiment	Environment
StyleGAN2 based GAN model training	8 NVIDIA Tesla V100-SXM2-16GB GPUs (DGX-1 station)
DCGAN based GAN model training	4 TITAN Xp GPUs
Metrics measurement	8 GeForce GTX 1080Ti GPUs
GAN inversion	1 TITAN Xp GPU

Table S5. Image inversion metrics for different StyleGAN2 based models. The perceptual loss is the mean square distance of VGG16 features between the original and projected images as in Abdal et al. (2019)

GAN arch	**Overall**		**Hard Cases**	
	MSE (↓)	Perceptual Loss (↓)	MSE (↓)	Perceptual Loss (↓)
Bald StyleGAN2	1.34	5.42	2.86	11.34
StyleGAN2 + ENI	1.24	4.86	2.58	9.82
StyleGAN2-NoPathReg + RNI	1.24	5.11	2.70	10.49
StyleGAN2 + RNI	**1.13**	**4.52**	**2.23**	**8.47**

that.

3. No decomposition: in this setting we remove the decomposition in Eq. (15). The network successfully converges, but admits a large PPL score.

4. CNN: in this setting we use a convolutional neural network to replace the procedure that we get σ in section 4.3. Namely, we take $\sigma = \mathbf{CNN}(\mu)$. The network successfully converges, but admits a very large FID score.

The zero PPL scores in 'No normalization' and 'No stabilization' suggest that the generator output is invariant to small perturbations, which means mode collapse. We can find that the stabilization and normalization in the RNI implementation in section 4.3 is necessary for the network to avoid numerical instability and mode collapse. The implementation of RNI method reaches the best performance in PPL score and comparable performance against the 'no decomposition' method in FID score. As analyzed in StyleGAN (Karras et al., 2019a) and StyleGAN2 (Karras et al., 2019b), for high fidelity images, PPL is more convincing than the FID score in measuring the synthesis quality. Therefore, the RNI implementation is the best among these methods.

Table S6. Ablation study of different noise injection methods on FFHQ. The zero values of PPL scores in the first two methods suggest mode collapse.

Method	FID	PPL
No normalization	628.94	0
No stabilization	184.30	0
No decomposition	**6.48**	18.78
CNN	22.54	14.53
RNI	7.31	**13.05**

References

Abdal, R., Qin, Y., and Wonka, P. Image2StyleGAN: How to embed images into the StyleGAN latent space? In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4432–4441, 2019.

Deimling, K. *Nonlinear functional analysis*. Courier Corporation, 2010.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019a.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of StyleGAN. *arXiv preprint arXiv:1912.04958*, 2019b.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Luxemburg, R. StyleGAN encoder. `https://github.com/rolux/stylegan2encoder`, 2020.

Murali, V. Fuzzy equivalence relations. *Fuzzy sets and systems*, 30(2):155–163, 1989.

Petersen, P., Axler, S., and Ribet, K. *Riemannian geometry*, volume 171. Springer, 2006.

Rudin, W. et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Zhang, L. and Zhang, B. Fuzzy reasoning model under quotient space structure. *Information Sciences*, 173(4): 353–364, 2005.

Zhao, D. and Tang, X. Cyclizing clusters via zeta function of a graph. In *Advances in Neural Information Processing Systems*, pp. 1953–1960, 2009.

Zhou, D., Weston, J., Gretton, A., Bousquet, O., and Schölkopf, B. Ranking on data manifolds. In *Advances in neural information processing systems*, pp. 169–176, 2004.

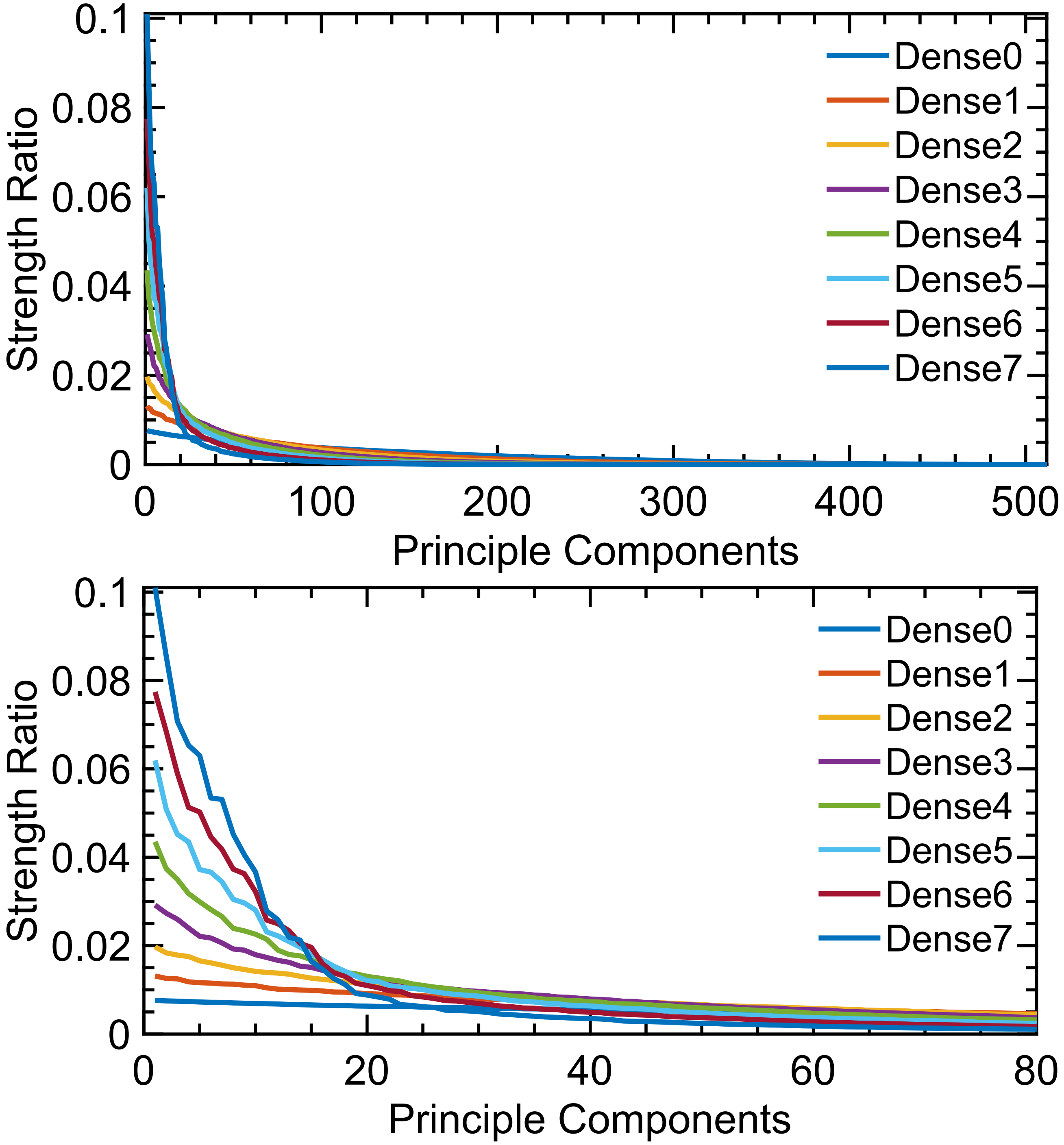


Figure S1. Strengths of the principal components of the Jacobian matrix of each dense layer. The sum of strengths of all components is normalized to 1. We can find that as layer gets deeper, the component strengths gather towards the left components, which means the number of valid components gets smaller and the rank of corresponding Jacobian gets smaller.

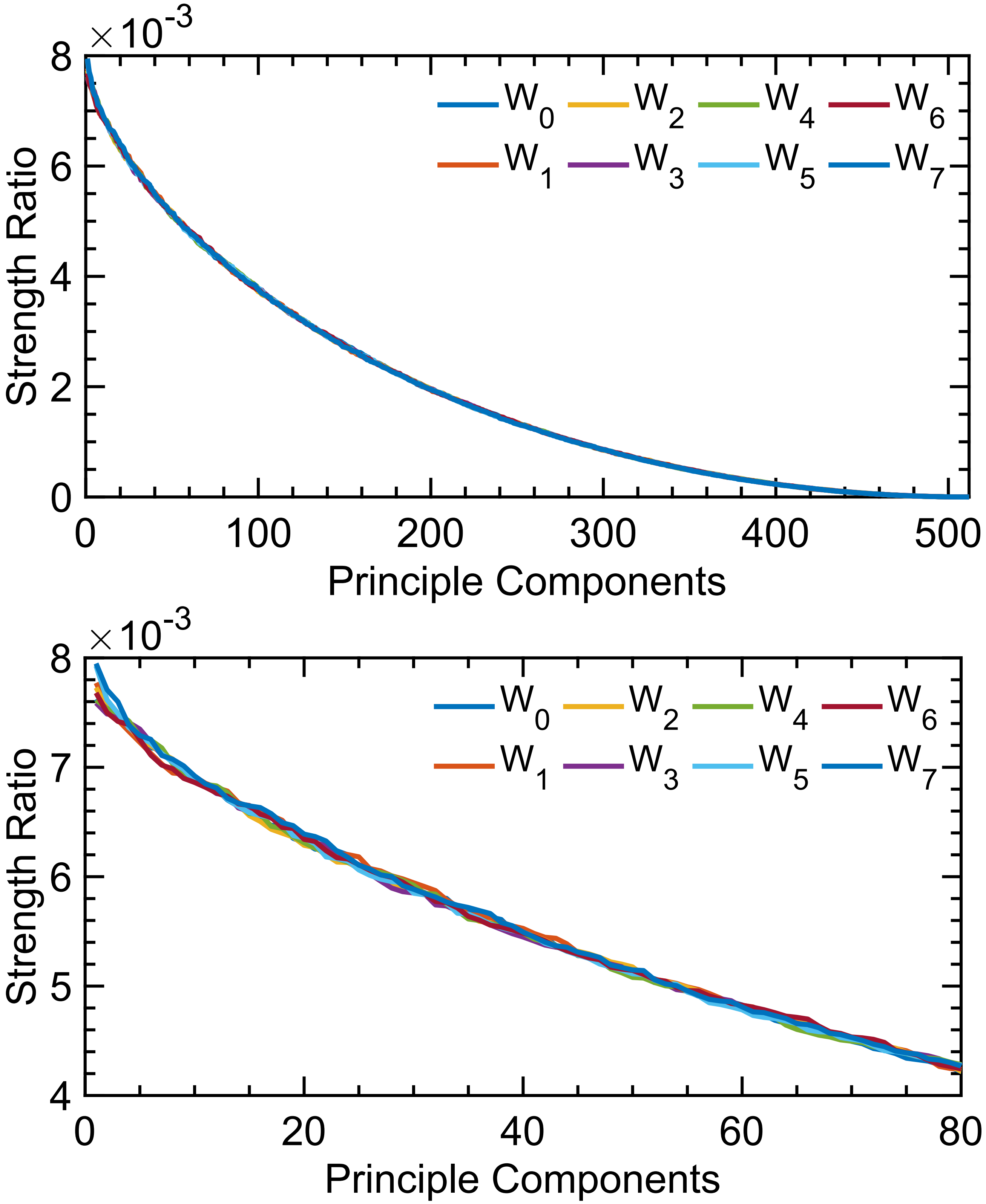


Figure S2. Strengths of the principal components of the weight matrices. The sum of strengths of all components is normalized to 1. We can find that the valid components of each weight matrix is around 450, which means each of the weight matrices will drop around 60 dimensions of the inputs

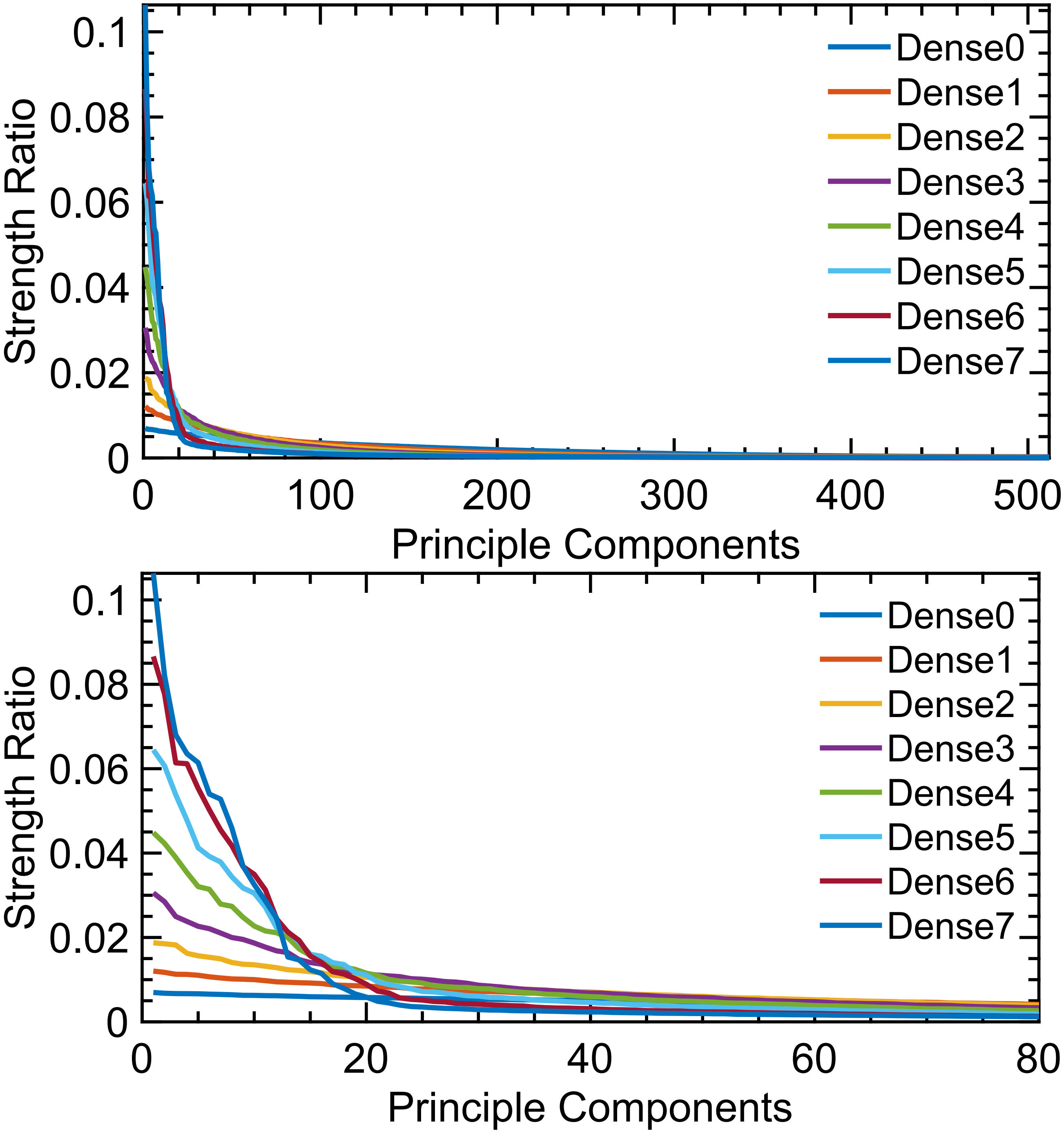
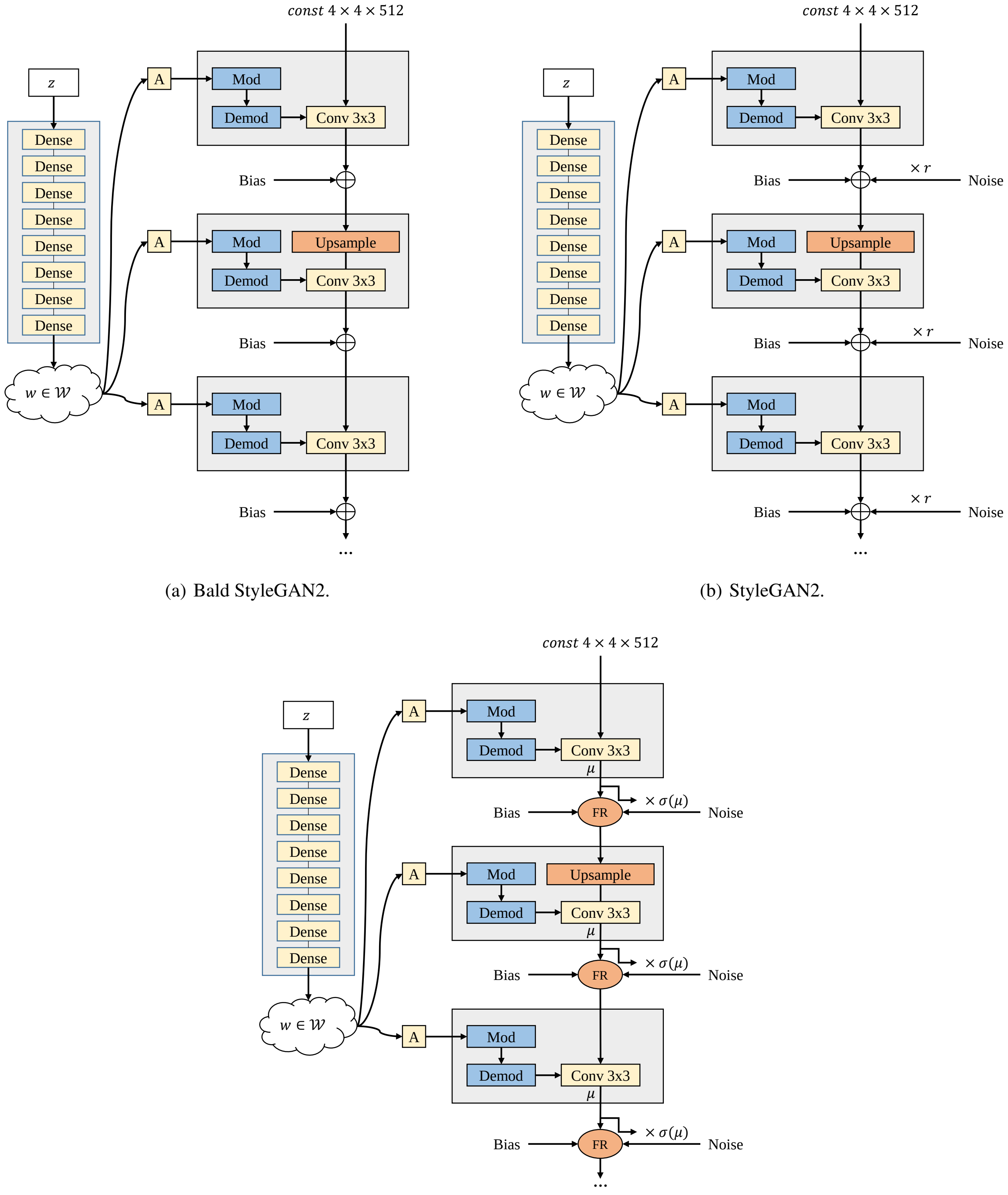


Figure S3. Strengths of the principal components of the output space of each dense layer. The sum of strengths of all components is normalized to 1. We can find that as layer gets deeper, the component strengths gather towards the left components, which means the number of valid components gets smaller and the intrinsic dimension of corresponding feature space gets smaller.

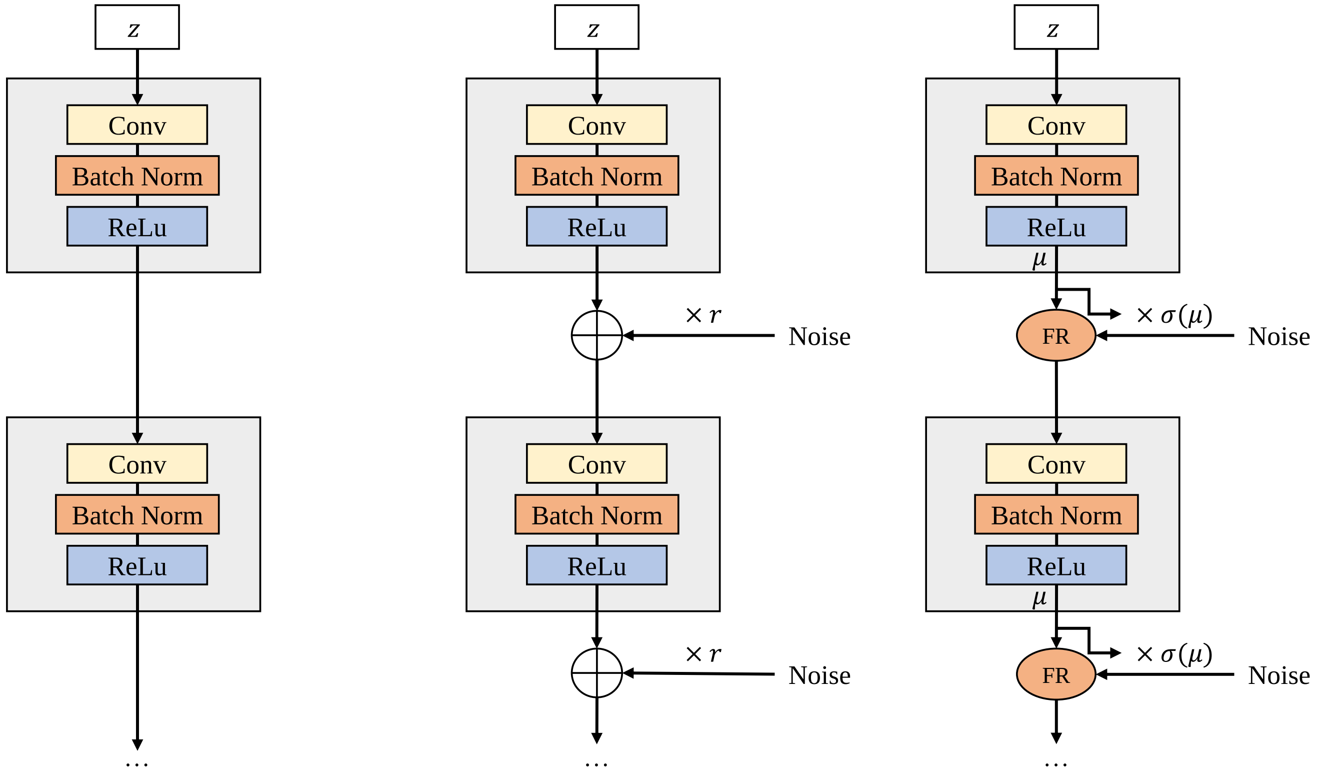


(a) Bald StyleGAN2.

(b) StyleGAN2.

(c) Fuzzy Reparameterization.

Figure S4. Generator architectures of StyleGAN2 based models. (a) The generator of bald StyleGAN2. (b) The generator of StyleGAN2. (c) The generator of StyleGAN2 + RNI and StyleGAN2-NoPathReg + RNI. 'Mod' and 'Demod' denote the weight demodulation method proposed in section 2.2 of StyleGAN2 (Karras et al., 2019b). A denotes a learned affine transformation from the intermediate latent space $\mathcal{W}$.



(a) DCGAN.

(b) Additive Noise Injection.

(c) Fuzzy Reparameterization.

Figure S5. Generator architecture of DCGAN based models. (a) The generator of DCGAN. (b) The generator of DCGAN + Additive Noise. (c) The generator of DCGAN + RNI.

Figure S6. Manually collected 20 images for GAN inversion.