

# Spectral Regularization for Combating Mode Collapse in GANs

Kanglin Liu, Wenming Tang, Ruitao Xie, and Guoping Qiu



Fig. 1: Conditional samples generated by our model base on ImageNet with setting  $E_{2048-64}$  in Table I. With the same setting, SN-GANs suffer from mode collapse, which starts at iteration = 40k. On the contrary, our model can gain performance improvement steadily with iterations, and no mode collapse occurs.

**Abstract**—Despite excellent progress in recent years, mode collapse remains a major unsolved problem in generative adversarial networks (GANs). In this paper, we present spectral regularization for GANs (SR-GANs), a new and robust method for combating the mode collapse problem in GANs. Theoretical analysis shows that the optimal solution to the discriminator has a strong relationship to the spectral distributions of the weight matrix. Therefore, we monitor the spectral distribution in the discriminator of spectral normalized GANs (SN-GANs), and discover a phenomenon which we refer to as spectral collapse, where a large number of singular values of the weight matrices drop dramatically when mode collapse occurs. We show that there are strong evidences linking mode collapse to spectral collapse; and based on this link, we set out to tackle spectral collapse as a surrogate of mode collapse. We have developed a spectral regularization method where we compensate the spectral distributions of the weight matrices to prevent them from collapsing, which in turn successfully prevents mode collapse in GANs. We provide theoretical explanations for why SR-GANs are more stable and can provide better performances than SN-GANs. We also present extensive experimental results and analysis to show that SR-GANs not only always outperform SN-GANs but also always succeed in combating mode collapse where SN-GANs fail. The code is available at <https://github.com/max-liu-112/SRGANs>.

**Index Terms**—Generative Adversarial Networks, Spectral Regularization, Mode Collapse, Lipschitz Constraint.

## I. INTRODUCTION

Generative Adversarial Networks (GANs) [1] are one of the most significant developments in machine learning research of the past decade. Since their first introduction, GANs have attracted intensive interest in the machine learning community not only for their ability to learn highly structured probability distributions but also for their theoretically implications [1], [2], [3], [4]. Essentially, GANs are constructed around two

K. Liu is with the Department of Information Engineering, Shenzhen University, Shenzhen, 518061, China. e-mail: max.liu.426@gmail.com.

G. Qiu is with University of Nottingham and Shenzhen University.

Manuscript received \*\*\*, 2019; revised \*\*\*, \*\*\*.

functions [5], [6]: the generator  $\mathbf{G}$ , which maps a sample  $z$  to the data distribution, and the discriminator  $\mathbf{D}$ , which is trained to distinguish real samples of a dataset from fake samples produced by the generator. With the goal of reducing the difference between the distributions of generated and real samples, a GAN training algorithm trains  $\mathbf{G}$  and  $\mathbf{D}$  in tandem.

GAN training is dynamic and sensitive to nearly every aspect of its setup, from optimization parameters to model architecture [7]. Training instability, or mode collapse, is one of the major obstacles in developing applications. Despite excellent progresses in recent years [8], [9], [10], [11], [12], the mode collapse problem still persists. For example, one of the most impressive works to emerge recently is BigGANs [7], which is the largest published GAN system based on the state of the art Spectral Normalization (SN-GAN)[10]. However, BigGANs can still suffer from the training instability problem, especially when the batch size is scaled up. Although implementing training stabilization measures such as employing  $R_1$  zero-centred gradient penalty term [7] in the loss metric of the discriminator to prevent spectral noise can improve stability, this can cause severe degradation in performance, resulting in a 45% reduction in Inception Score.

In this paper, we present Spectral Regularization, a robust method for combating the mode collapse problem in GANs. Theoretically, we have found that spectral distribution is closely related to  $D$ 's performance, and affect how the supremum of Lipschitz constraint can be reached. Regarding mode collapse as the rapid decay in performance, we reason that the spectral distributions in  $\mathbf{D}$  may also have a strong relation to mode collapse. Through comprehensive analysis of spectral distributions in a large number of GAN models trained with the state of the art SN-GAN algorithm, we discover that when mode collapse occurs to a model, spectral distributions of  $\bar{W}_{\text{SN}}(W)$  in  $\mathbf{D}$  also collapse, where  $\bar{W}_{\text{SN}}(W)$  is spectral normalized weight matrix. Specifically, we observe that when a model performs well and no mode collapse occurs, there are a large number of singular values of  $\bar{W}_{\text{SN}}(W)$  in  $\mathbf{D}$  very

close to 1, and that when mode collapse occurs to a model, singular values of  $\bar{W}_{\text{SN}}(W)$  in  $\mathbf{D}$  will drop dramatically. We refer to the phenomenon where a large number of singular values drop significantly as **spectral collapse**.

In all GAN models of various sizes and trained with a variety of parameter settings on datasets extensively used in the literature, we observe that mode collapse and spectral collapse always go side by side. This fact leads us to reason that mode collapse in SN-GANs is caused by spectral collapse in  $\mathbf{D}$ 's weight matrices. Based on such insight into spectral distributions of  $\bar{W}_{\text{SN}}(W)$ , we propose a new and robust method called spectral regularization to prevent GANs from mode collapse. In addition to normalizing the weight matrices, spectral regularization imposes constraints on  $\mathbf{D}$ 's weight matrices by compensating their spectral distributions to avoid spectral collapse. Theoretical analysis shows that spectral regularization is better than spectral normalization at preventing weight matrix from concentrating into one particular direction. We show that SN-GANs are a special case of spectral regularization, and in a series of extensive experiments we demonstrate that spectral regularization not only provides superior performances to spectral normalization but also can always avoid mode collapse in cases where spectral normalization failed.

Our contributions can be summarized as follows:

(1) Through theoretical analysis and extensive experimental observations, we provide an insight into the likely causes of mode collapse in a state of the art GAN normalization technique, spectral normalization (SN-GANs). We introduce the concept of **spectral collapse** and provide strong evidence to link spectral collapse with mode collapse in SN-GANs.

(2) Based on above insight, we have developed a new robust regularization method, **Spectral Regularization**, where we compensate the spectral distributions of the weight matrices in  $\mathbf{D}$  to prevent spectral collapse, thus preventing mode collapse in GANs. Extensive experimental results show that spectral regularization not only can always prevent mode collapse but also can consistently provide improved performances over SN-GANs.

## II. ANALYSIS OF MODE COLLAPSE IN SN-GANs

### A. A Brief Summary of SN-GANs

For easy discussion, we first briefly recap the essential ideas of the spectral normalization technique for training GANs [10]. As far we are aware, this is currently one of the best methods in the literature and has been successfully used to construct large systems such as BigGANs [7]. For convenience, we largely follow the notation convention of [10]. Considering a simple discriminator of a neural network of the following form:

$$f(x, \theta) = W^{L+1}(a_L \cdot W^L \cdot a_{L-1} \cdot W^{L-1} \cdots a_1 W^1 x) \quad (1)$$

where  $\theta := \{W^1, \dots, W^L, W^{L+1}\}$  is the learning parameters set,  $W^l \in \mathbb{R}^{d_l \times d_{l-1}}$ ,  $W^{L+1} \in \mathbb{R}^{1 \times d_L}$ , and  $a_l$  is an element-wise non-linear activation function. We omit the bias term of each layer for simplicity. The final output of the discriminator is given by

$$D(x, \theta) = \mathcal{A}(f(x, \theta)) \quad (2)$$

where  $\mathcal{A}$  is an activation function corresponding to the divergence of a distance measure of users' choice.

The standard formulation of GANs is given by [10], [2]:

$$\min_G \max_D V(G, D) \quad (3)$$

where min and max of  $G$  and  $D$  are taken over the set of the generator and discriminator functions respectively. The conventional form of  $V(G, D)$  is given by  $E_{x \sim q_{\text{data}}}[\log D(x)] + E_{x' \sim q_G}[\log(1 - D(x'))]$  [10], where  $q_{\text{data}}$  is the data distribution and  $q_G$  is the model (generator) distribution.

To guarantee Lipschitz continuity, spectral normalization [10] controls the Lipschitz constant of the discriminator function by literally constraining the spectral norm of each layer:

$$\bar{W}_{\text{SN}}(W) := W/\sigma(W) \quad (4)$$

where  $\sigma(W)$  is the spectral norm of the weight matrix  $W$  in the discriminator network, which is equivalent to the largest singular value of  $W$ .

The authors of SN-GANs [10] and those of BigGANs [7] have demonstrated the superiority of spectral normalization over other normalization or regularization techniques, e.g., gradient penalty [8], weight normalization[11] and orthonormal regularization [13]. However, as a state of the art GAN model, BigGANs (based on spectral normalization) can still suffer from mode collapse. Therefore, mode collapse remains an unsolved open problem, seeking better and more robust solution is very important for advancing GANs.

### B. Theoretical Analysis

Our theoretical analysis has found that the spectral distribution of the weight matrix determine how the supremum of Lipschitz constraint can be reached, and affect performance of  $D$ .

To be specific. SN-GANs enforce Lipschitz constraint on the weight matrix in each convolutional layers of  $D$  by constraining the spectral norm, thus guaranteeing  $D$  satisfy Lipschitz constraint. Furthermore, we have found that the spectral distribution affects how the supremum of Lipschitz constraint can be reached. To verify this, Corollary 1 is present here. (see proof in Appendix)

**Corollary 1.** If a linear function  $f = Wx$  satisfies Lipschitz constraint:  $\|f(x_1) - f(x_2)\| \leq \|x_1 - x_2\|$ , then the supremum of the Lipschitz constraint is obtained when all the singular values of the weight matrix  $W$  are 1.

On the other hand, through singular value decomposition, we find that spectral distribution affect the performance of  $D$ .

The weight matrix  $W$  in the convolutional layers can be expressed as:

$$\begin{aligned} W &= U \cdot \Sigma \cdot V^T \\ &= \sigma_1 u_1 v_1 + \sigma_2 u_2 v_2 + \cdots + \sigma_n u_n v_n \end{aligned} \quad (5)$$

where both  $U$  and  $V$  are orthogonal matrix, the columns of  $U$ ,  $[u_1, u_2, \dots, u_m]$ , are called left singular vectors of  $W$ , the columns of  $V$ ,  $[v_1, v_2, \dots, v_n]$ , are called right singular vectors of  $W$ , and  $\sigma_1 \cdots \sigma_n$  are the singular values of  $W$ .

We can see that,  $\sigma$  in Equation 5 determine how corresponding singular vectors are utilized in implementing convolutional operation  $Wx$ . Extremely, if  $\sigma_1 > 0$ , and  $\sigma_2 = \cdots = \sigma_n = 0$ ,

TABLE I: Experiment settings. The experiments are divided into 5 groups  $A, B, C, D$  and  $E$ . Within each group, the models share exactly the same network architecture but differ in batch size. For groups  $A - D$ , we vary the batch sizes inside each group to study how batch sizes relate to mode collapse, and we change the channel sizes between groups to investigate how discriminator capacity affects mode collapse. Group  $E$  is experiments applied to a different data set. The purpose is to evaluate how different data affect mode collapse. Batch represents the batch size. CH is the channel size of the discriminator. The subscript of each group name annotates the batch and channel setting of that experiment, e.g.,  $A_{a-b}$  represents setting with a batch size  $a$  and a CH size  $b$ .

Setting	Batch	CH	Dataset	Setting	Batch	CH	Dataset
$A_{16-128}$	16	128	CIFAR-10	$C_{8-32}$	8	32	CIFAR-10
$A_{32-128}$	32	128	CIFAR-10	$C_{16-32}$	16	32	CIFAR-10
$A_{64-128}$	64	128	CIFAR-10	$C_{32-32}$	32	32	CIFAR-10
$A_{128-128}$	128	128	CIFAR-10	$C_{64-32}$	64	32	CIFAR-10
$A_{256-128}$	256	128	CIFAR-10	$D_{128-256}$	128	256	CIFAR-10
$A_{512-128}$	512	128	CIFAR-10	$D_{256-256}$	256	256	CIFAR-10
$A_{1024-128}$	1024	128	CIFAR-10	$D_{512-256}$	512	256	CIFAR-10
$B_{8-64}$	8	64	CIFAR-10	$E_{64-128}$	64	128	STL-10
$B_{16-64}$	16	64	CIFAR-10	$E_{256-128}$	256	128	STL-10
$B_{32-64}$	32	64	CIFAR-10	$E_{256-64}$	256	64	STL-10
$B_{64-64}$	64	64	CIFAR-10	$E_{256-32}$	256	32	STL-10
$B_{128-64}$	128	64	CIFAR-10	$E_{512-64}$	512	64	ImageNet
$B_{256-64}$	256	64	CIFAR-10	$E_{2048-64}$	2048	64	ImageNet

we can see that the convolutional operation  $Wx$  only uses the first singular value and its corresponding singular vector.

Through the theoretical analysis above, it is clear that spectral distribution plays a key role in determining Lipschitz constraint and  $D$ 's performance. However, spectral normalization only constrains the spectral norm, and takes no consideration on spectral distribution. Recalling that mode collapse can be regarded as the intense decay in  $D$ 's performance, we reason that mode collapse may also have a strong relationship with spectral distribution. Furthermore, finding such a relationship may help solve the mode collapse problem.

### C. Mode Collapse vs Spectral Collapse

In order to find the likely link between mode collapse and spectral distributions, we have conducted a series of experiments for unconditional image generation on CIFAR-10 [14], STL-10 [15] and conditional image generation on ILRSVRC2012 [16] datasets. Our implementation is based on the SN-GANs architecture of [10], which uses the hinge loss as the discriminator objective and is given by:

$$\begin{aligned} L_D = & E_{x \sim q_{data}} [\min(0, -1 + D(x))] \\ & + E_{x \sim q_G} [\min(0, -1 - D(x))] \end{aligned} \quad (6)$$

The optimization settings follow literature [10], [17]. Previous authors have shown that increasing batch size or decreasing discriminator capacity could potentially lead to mode collapse [7]. We therefore conduct experiments for various combinations of batch and channel sizes as listed in Table I. We follow the practices in the literature of using Inception Score (IS) [18] and Fréchet Inception Distance (FID) [15] as approximate measures of sample quality, and results are shown in Table II where we also identify all settings where

mode collapse has occurred to SN-GANs. Through monitoring Inception Scores, Fréchet Inception Distance and synthetic images during training, mode collapse is observed in 10 settings including  $B_{64-64}, B_{128-64}, B_{256-64}, C_{8-32}, C_{16-32}, C_{32-32}, C_{64-32}, E_{256-64}, E_{256-32}$  and  $E_{2048-64}$ . In other 16 settings, mode collapse has not happened.

Mode collapse is a persistent problem in GAN training and is also a major issue in SN-GANs as has been shown in BigGANs[7] and in Table II. Here, we monitor the entire spectral distributions of SN-GANs, i.e., all singular values of  $\bar{W}_{SN}(W)$  in the discriminator network during training.

The discriminator network in our implementation uses the same architecture as that in the original SN-GANs[10]. For image generation on CIFAR-10 and STL-10, there are 10 convolutional layers. As for image generation on ImageNet, 17 convolutional layers are included. Please see Appendix for the setting details. In order to discover the likely link of mode collapse to spectral distribution, we plot the spectral distributions of every layer of the discriminator for all 26 settings. In the following, we present some typical examples and readers are referred to the Appendix for all other plots.

Figure 2 shows the spectral distributions of the last layer ( $layer\_9$  for task on CIFAR-10 and STL-10,  $layer\_16$  for task on ImageNet) of 5 settings where mode collapse does not happen. Figure 3 shows the spectral distributions of the last layer of all 10 settings where mode collapse has occurred. Through analyzing the spectral distribution plots in Figure 2 and Figure 3, we notice a very interesting pattern. In the cases where no mode collapse happens, the shapes of the spectral distribution curves do not change significantly with the number of training iteration. On the other hand, for those settings where mode collapse has occurred, the shapes of the spectral distribution curves change significantly as training progresses. In particular, a large number of singular values become very small when training passes a certain number of iterations. This is as if the curves have "collapsed", and we refer to this phenomenon as **spectral collapse**.

The phenomenon of spectral collapse is also observed across different settings. Figure 4 plots the spectral distributions of the 5 groups of experimental settings in Table I. It is seen that in groups  $A$  and  $D$ , the spectral distributions across different settings are very similar and no spectral collapse is observed. Very interestingly, no mode collapse is observed either. In group  $B$ , the spectral distributions of  $B_{64-64}, B_{128-64}$  and  $B_{256-64}$  have collapsed, not surprisingly, mode collapse also happens to these 3 settings. In group  $C$ , the spectral distributions of all settings have collapsed, i.e., most singular values are very small (except for the first one which is forced to be 1 by spectral normalization). Again as expected, mode collapse happens to all settings in this group. In group  $E$ , it is seen that settings  $E_{256-64}, E_{256-32}$  and  $E_{2048-64}$  have suffered from spectral collapse. Again, mode collapse is observed for these three settings.

In order to understand what has happened when spectral collapse occurs, Figure 5 shows how a typical spectral distribution relates to Inception Score and Fréchet Inception Distance during training. It is seen that up to 19k iterations both IS and FID are showing good performances, and the

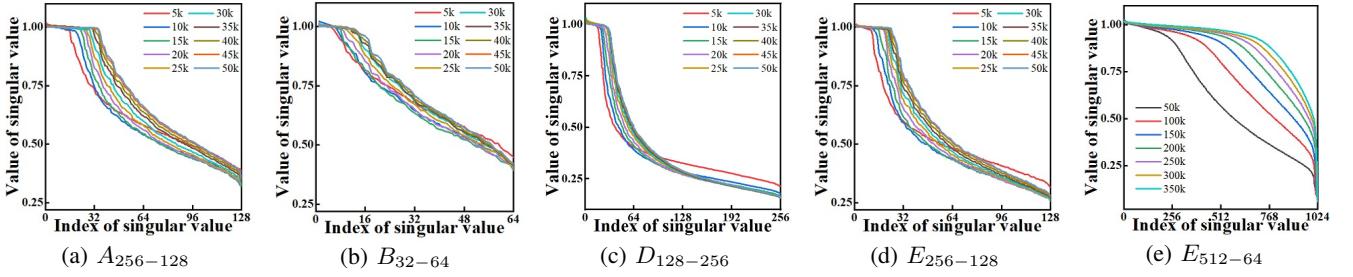


Fig. 2: Spectral distributions in the last layer for Good GANs (no mode collapse) at different number of iterations. The curves represent the spectral distributions after 5k iterations, 10k iterations, ..., 50k iterations, ... .

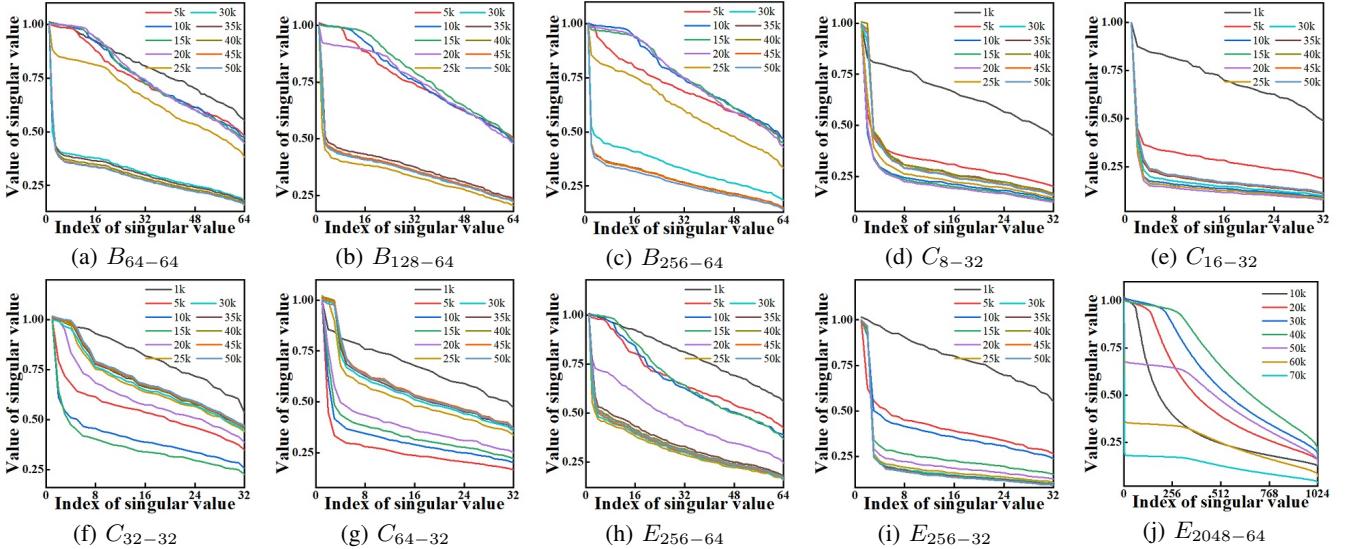


Fig. 3: Spectral distributions in the last layer for settings where mode collapse occurs. The curves represent the spectral distributions after 1k iterations, 10k iterations, ..., 50k iterations, ... .

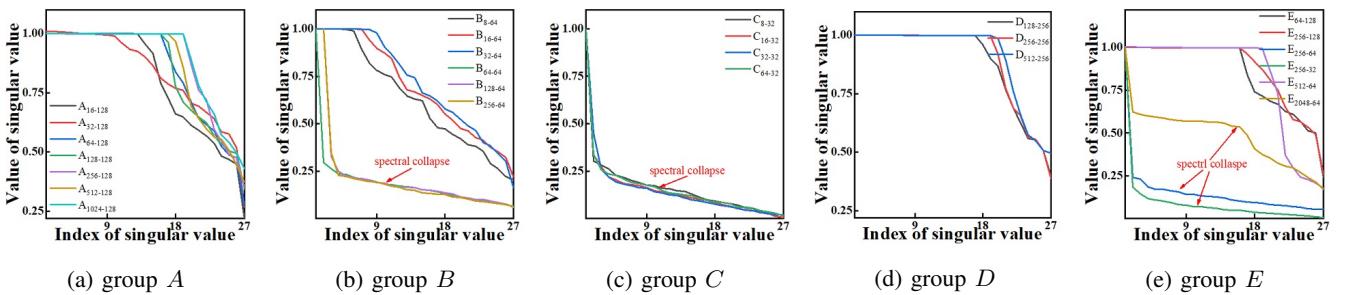


Fig. 4: Spectral distributions (after 50k iterations) in *layer\_0* for different settings.

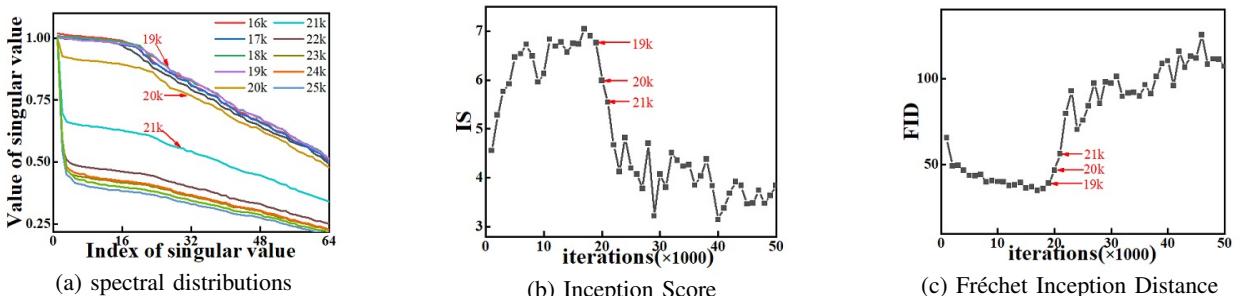


Fig. 5: An example showing how spectral distributions relate to Inception Score and Fréchet Inception Distance. Here the setting is  $B_{128-64}$  and the spectral distributions correspond to those of *layer\_9*.

corresponding spectral distribution has a large number of large singular values. At 20k iterations, IS and FID performances start to drop, correspondingly, the spectral distribution starts to fall. At 21k iterations, the IS and FID performances have dropped significantly and mode collapse has started, and very importantly, the spectral distribution has dropped dramatically - starting to collapse.

The association of mode collapse with spectral collapse is observed for all the layers and on all settings (readers are referred to the Appendix for more examples). We therefore believe that mode collapse and spectral collapse happen at the same time, and spectral collapse is the likely cause of mode collapse. In the following section, we will introduce spectral regularization to prevent spectral collapse thus avoiding mode collapse.

### III. SPECTRAL REGULARIZATION

We have now established that spectral collapse is closely linked to mode collapse in SN-GANs. In this section, we introduce spectral regularization, a technique for preventing spectral collapse. We show that preventing spectral collapse can indeed solve the mode collapse problem, thus demonstrating that spectral collapse is the cause of mode collapse rather than a mere symptom.

Performing singular value decomposition, the weight matrix  $W$  can be expressed as:

$$W = U \cdot \Sigma \cdot V^T \quad (7)$$

where both  $U$  and  $V$  are orthogonal matrix, the columns of  $U$ ,  $[u_1, u_2, \dots, u_m]$ , are called left singular vectors of  $W$ , the columns of  $V$ ,  $[v_1, v_2, \dots, v_n]$ , are called right singular vectors of  $W$ , and  $\Sigma$  can be expressed as:

$$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \quad (8)$$

where  $D = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_r\}$  represents the spectral distribution of  $W$ .

When mode collapse occurs, spectral distributions concentrate on the first singular value, and the rest singular values drop dramatically (spectral collapse). To avoid spectral collapse, we can apply  $\Delta D$  to compensate  $D$ . Here, we introduce two methods for implementation of  $\Delta D$ . The first method is referred to as static compensation, where  $\Delta D$  is given by  $\text{diag}\{\sigma_1 - \sigma_1, \sigma_1 - \sigma_2, \dots, \sigma_1 - \sigma_i, 0, \dots, 0\}$ , and  $i$  is a hyperparameter ( $1 \leq i \leq r$ ). In other words, static compensation is applied according to the difference between the first and  $i$ -th singular value.

The other method is referred to as dynamic compensation. In the training process, we monitor the maximum ratio  $\gamma_j = \max(\frac{\sigma'_j}{\sigma'_1})$ , where  $\sigma'_j$  represents recorded  $j$ -th singular value in the training process. Then, we utilize the difference between maximum ratio value and current value as the compensation. As a result,  $\Delta D$  is given by  $\text{diag}\{0, \gamma_2 \cdot \sigma_1 - \sigma_2, \dots, \gamma_r \cdot \sigma_1 - \sigma_r\}$ . Comparison of these two compensation method will be discussed in Section IV.

Spectral compensation turns  $D$  into  $D'$  as follows:  $D' = D + \Delta D$ . Correspondingly,  $W$  turns to  $W'$ :  $W' = W + \Delta W$ ,

where  $\Delta W$  is given by:

$$\Delta W = U \cdot \begin{bmatrix} \Delta D & 0 \\ 0 & 0 \end{bmatrix} \cdot V^T = \sum_{k=2}^N (\Delta \sigma_k) u_k v_k^T \quad (9)$$

where  $N$  is the number of singular value to be compensated, and  $\Delta \sigma_k$  represents the compensation value in  $k$ -th singular value. For static compensation,  $N$  in Equation 9 equals to the hyperparameter  $i$ , while for dynamic compensation,  $N$  equals to the number of singular values  $r$ .

Finally, we apply spectral normalization to guarantee Lipschitz continuity, and obtain our spectral regularized  $\bar{W}_{\text{SR}}(W)$ :

$$\bar{W}_{\text{SR}}(W) = \frac{W + \Delta W}{\sigma(W)} = \bar{W}_{\text{SN}}(W) + \Delta W / \sigma(W) \quad (10)$$

Clearly, spectral normalization is a special case of spectral regularization, when no compensation is applied.

#### A. Gradient Analysis of Spectral Regularization

We perform gradient analysis to show that spectral regularization provides a more effective way over spectral normalization in preventing  $W$  from concentrating into one particular direction during training and thus avoiding spectral collapse.

From Equation (10), taking static compensation as an example, we can write the gradient of  $\bar{W}_{\text{SR}}(W)$  with respect to  $W_{ab}$  as:

$$\begin{aligned} \frac{\partial \bar{W}_{\text{SR}}(W)}{\partial W_{ab}} &= \frac{1}{\sigma(W)} \{E_{ab} - \bar{W}_{\text{SN}}[u_1 v_1^T]_{ab} \\ &\quad - \frac{\Delta W}{\sigma(W)} [u_1 v_1^T]_{ab} + \sum_{k=2}^i [u_1 v_1^T - u_k v_k^T]_{ab} \cdot u_k v_k^T\} \end{aligned} \quad (11)$$

where  $[\cdot]_{ab}$  represents the  $(a, b)$ -th entry of corresponding matrix,  $E_{ab}$  is the matrix whose  $(a, b)$ -th entry is 1 and zero everywhere else.

We would like to comment on the implication of Equation (11). The first two terms,  $E_{ab} - \bar{W}_{\text{SN}}[u_1 v_1^T]_{ab}$ , are the gradient of spectral normalization  $\frac{\partial \bar{W}_{\text{SN}}(W)}{\partial W_{ab}}$  [10], this is very easy to see from Equation (10). As explained in [10], the second term can be regarded as being able to prevent the columns space of  $W$  from concentrating into one particular direction in the course of training. In other words, spectral normalization prevents the transformation of each layer from becoming sensitive only in one direction. However, as we have seen (e.g. Figure 3), despite performing spectral normalization, the spectral distributions of  $\bar{W}_{\text{SN}}(W)$  can still concentrate on the first singular value thus causing spectral collapse. This shows the limited ability of spectral normalization in preventing  $W$  from spectral collapse.

In addition to the first two terms of spectral normalization, spectral regularization introduces the third and fourth terms in Equation (11). It can be seen that the third term enhances the effect of the second term, through which  $W$  is much less likely to concentrate into one particular direction. Furthermore, the fourth term can be seen as the regularization term, encouraging  $W$  to move along all  $i$  directions pointed to by  $u_k v_k^T$ , for  $k = 1, 2, \dots, i$ , each weighted by the adaptive regularization coefficient  $[u_1 v_1^T - u_k v_k^T]_{ab}$ . This encourages  $W$  to make full use of the directions pointed to by  $u_j v_j^T$ , thus preventing  $W$

TABLE II: IS and FID results for different settings, where IS is Inception Score and FID is Fréchet Inception Distance. For IS, higher is better, while lower is better for FID. SN, SR represent Spectral normalization and Spectral Regularization, respectively. MC stands for mode collapse, and SC stands for spectral collapse,  $\times$  represents that no mode collapse or spectral collapse occurs. SN in the MC column or SC column represents that mode collapse or spectral collapse occurred to spectral normalization. Note that neither mode collapse nor spectral collapse happen to spectral regularization for all settings.

Experiment Setting	IS		FID		MC	SC	Experiment Setting	IS		FID		MC	SC
	SN	SR	SN	SR				SN	SR	SN	SR		
$A_{16-128}$	8.15 $\pm$ .09	8.35 $\pm$ .09	22.31 $\pm$ .28	24.67 $\pm$ .28	$\times$	$\times$	$C_{8-32}$	4.21 $\pm$ .18	4.93 $\pm$ .20	80.00 $\pm$ 1.12	66.05 $\pm$ 2.12	<b>SN</b>	<b>SN</b>
$A_{32-128}$	8.38 $\pm$ .07	8.45 $\pm$ .10	25.96 $\pm$ .42	22.00 $\pm$ .17	$\times$	$\times$	$C_{16-32}$	4.05 $\pm$ .15	4.78 $\pm$ .23	79.69 $\pm$ .21	59.25 $\pm$ .43	<b>SN</b>	<b>SN</b>
$A_{64-128}$	8.39 $\pm$ .15	8.65 $\pm$ .12	21.15 $\pm$ .15	20.31 $\pm$ .18	$\times$	$\times$	$C_{32-32}$	4.29 $\pm$ .08	4.70 $\pm$ .15	78.39 $\pm$ .17	62.10 $\pm$ .24	<b>SN</b>	<b>SN</b>
$A_{128-128}$	8.61 $\pm$ .12	8.72 $\pm$ .08	21.01 $\pm$ .23	19.98 $\pm$ .19	$\times$	$\times$	$C_{64-32}$	4.30 $\pm$ .14	5.00 $\pm$ .14	85.15 $\pm$ 1.20	56.11 $\pm$ .54	<b>SN</b>	<b>SN</b>
$A_{256-128}$	8.45 $\pm$ .14	8.48 $\pm$ .03	20.87 $\pm$ .25	19.87 $\pm$ .21	$\times$	$\times$	$D_{128-256}$	8.14 $\pm$ .06	8.92 $\pm$ .18	24.43 $\pm$ .41	18.95 $\pm$ .23	$\times$	$\times$
$A_{512-128}$	8.34 $\pm$ .09	8.53 $\pm$ .04	21.85 $\pm$ .14	20.13 $\pm$ .12	$\times$	$\times$	$D_{256-256}$	8.29 $\pm$ .12	8.83 $\pm$ .14	22.54 $\pm$ .29	19.56 $\pm$ .11	$\times$	$\times$
$A_{1024-128}$	8.31 $\pm$ .21	8.52 $\pm$ .16	21.68 $\pm$ .35	20.34 $\pm$ .13	$\times$	$\times$	$D_{512-256}$	8.33 $\pm$ .09	8.36 $\pm$ .12	22.58 $\pm$ .16	21.82 $\pm$ .29	$\times$	$\times$
$B_8-64$	6.67 $\pm$ .05	7.42 $\pm$ .06	45.19 $\pm$ .89	35.78 $\pm$ .11	$\times$	$\times$	$E_{64-128}$	8.98 $\pm$ .20	9.14 $\pm$ .18	42.40 $\pm$ .56	39.89 $\pm$ .89	$\times$	$\times$
$B_{16-64}$	7.34 $\pm$ .06	7.59 $\pm$ .08	31.73 $\pm$ .49	29.42 $\pm$ .22	$\times$	$\times$	$E_{256-128}$	9.10 $\pm$ .13	9.11 $\pm$ .17	40.11 $\pm$ .89	40.08 $\pm$ .29	$\times$	$\times$
$B_{32-64}$	7.18 $\pm$ .03	7.48 $\pm$ .09	33.76 $\pm$ .35	28.60 $\pm$ .25	$\times$	$\times$	$E_{256-64}$	7.38 $\pm$ .14	7.67 $\pm$ .06	74.50 $\pm$ 1.52	69.20 $\pm$ .83	<b>SN</b>	<b>SN</b>
$B_{64-64}$	6.96 $\pm$ .11	7.52 $\pm$ .11	36.65 $\pm$ .29	28.40 $\pm$ .36	<b>SN</b>	<b>SN</b>	$E_{256-32}$	4.04 $\pm$ .11	4.38 $\pm$ .07	98.50 $\pm$ 1.34	89.17 $\pm$ 1.23	<b>SN</b>	<b>SN</b>
$B_{128-64}$	7.10 $\pm$ .14	7.13 $\pm$ .05	35.99 $\pm$ .48	31.41 $\pm$ .56	<b>SN</b>	<b>SN</b>	$E_{512-64}$	34.13 $\pm$ .64	<b>40.76<math>\pm</math>.73</b>	59.88 $\pm$ .57	<b>54.84<math>\pm</math>.68</b>	$\times$	$\times$
$B_{256-64}$	6.85 $\pm$ .08	7.58 $\pm$ .03	35.88 $\pm$ .42	27.68 $\pm$ .23	<b>SN</b>	<b>SN</b>	$E_{2048-64}$	21.78 $\pm$ .87	<b>31.58 <math>\pm</math>.43</b>	71.37 $\pm$ 1.14	<b>61.08 <math>\pm</math>.53</b>	<b>SN</b>	<b>SN</b>

from being concentrated on only 1 direction, which in turn stabilizes the training process.

From above analysis, it is clear that as compared to spectral normalization, spectral regularization of Equation (11) encourages  $W$  of the discriminator to move in a variety of directions thus preventing it from concentrating only on one direction, which in turn prevents spectral collapse. We will show in the experimental section that performing spectral regularization can indeed prevent mode collapse where spectral normalization has failed.

#### IV. EXPERIMENTS

For all settings listed in Table I, we have conducted experiments using SN-GANs and the newly introduced spectral regularization algorithm (we use the abbreviation: SR-GANs for the spectral regularized GANs). All procedures and settings for SN-GANs and SR-GANs are identical, except that for SR-GANs the last discriminator update implements spectral regularization (Equation 10) and SN-GANs implement spectral normalization (Equation 4).

Before studying the properties of spectral regularization, we demonstrate experimentally that the new SR-GANs is superior to SN-GANs in both quality and stability. The Inception Score (IS) and Fréchet Inception Distance (FID) performances are shown in Table II. Please note that in the cases where mode collapse have happened, IS and FID are the best results before mode collapse. It is clearly seen that in all cases, SR-GANs outperforms SN-GANs. In particular, for conditional image generation on ImageNet with setting  $E_{512-64}$  and  $E_{2048-64}$ , SR-GAN has improved IS by 19.4% and 44.9%, respectively. On average, SR-GANs have improved the IS by 13.9% and FID by 21.8% over SN-GANs. Very importantly, in all 10 settings where mode collapse has occurred to SN-GANs, none has happened to SR-GANs. In fact, we have not yet observed mode collapses in an extensive set of experiments.

#### A. Conditional Generation on ImageNet

It is clearly seen in Table II, spectral regularization greatly improves the performance, and contributes to stable training. For setting  $E_{512-64}$ , no mode collapse is observed in SN-GAN. Applying spectral regularization can further improve the performance. As for setting  $E_{2048-64}$ , spectral regularization avoids mode collapse, and greatly improve the image quality.

In Figure 6, we show the training history of SN-GAN with setting  $E_{2048-64}$ . It is clear that at around 40k iterations, mode collapse occurs, and image quality starts to decrease dramatically. Accordingly, synthetic images in Figure 6(f) are of low quality and limited diversity.

To demonstrate the superiority of SR over SN, we resume the training with the snapshot of SN-GAN at iterations = 40k, and apply Spectral Regularization with dynamic compensation (SR-d). The effect of SR on improving performance and guaranteeing training stability is rather obvious. Firstly, spectral regularization avoids the occurrence of mode collapse, which is supposed to happen at iterations = 40k in SN-GAN. What's more, spectral regularization contributes to performance improvement as well, i.e. IS, FID are improved by 44.9% and 9.1%, respectively.

Figure 6(c) shows the spectral distribution of *layer\_16* in SR-GAN. It is clearly shown that spectral regularization avoids spectral collapse observed in Figure 3(j). Owing to the stronger constraint of spectral regularization on spectral distributions of each convolutional layer, SR-GANs can gain performance improvement steadily with iterations, instead of suffering from mode collapse.

#### B. Unconditional Generation on CIFAR-10 and STL-10

We apply spectral Regularization with static compensation (SR-s) in unconditional generation on CIFAR-10 and STL-10 datasets. The default value of the hyperparameter  $i$  for static compensation in SR-GANs is empirically set as  $i = 0.5r$ , where  $r$  is the number of singular values in the corresponding

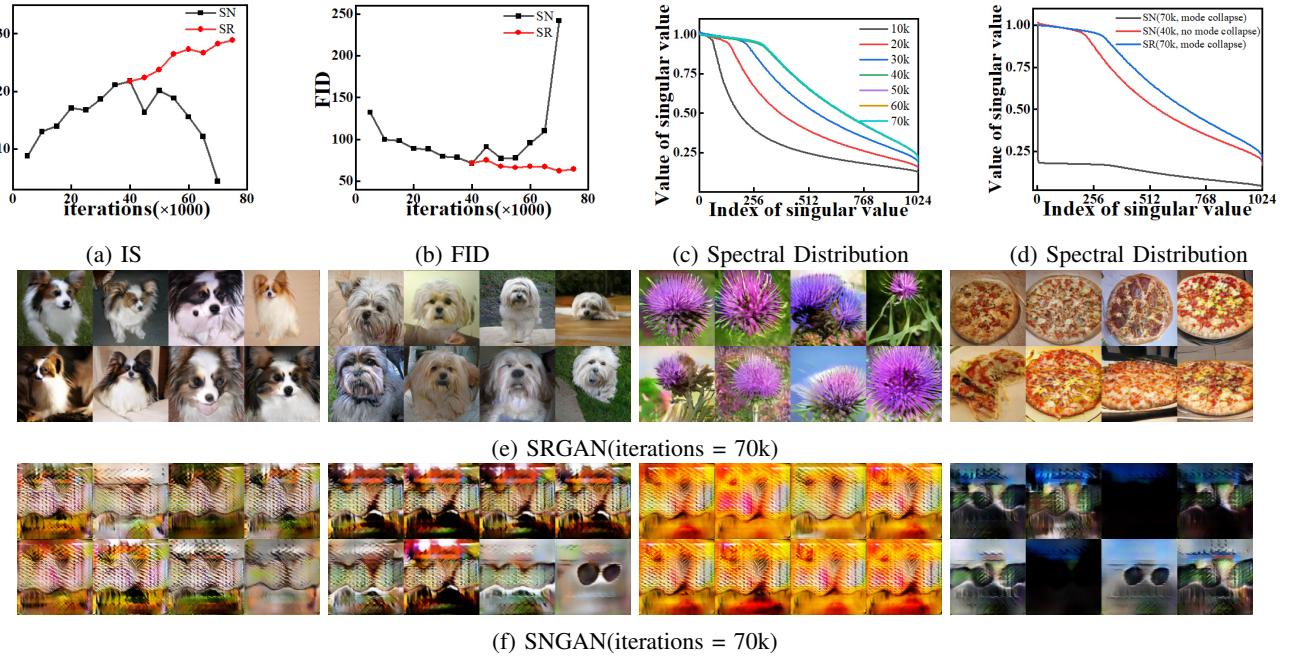


Fig. 6: Inception Score, Fréchet Inception Distance, spectral distribution and synthetic images of SN-GAN and SR-GAN for the setting  $E_{2048-64}$ . (a), (b) shows the IS and FID, respectively. (c) is the spectral distributions for *layer\_16* in SRGAN. (d) indicates the effect of SR on spectral distribution. (e), (f) are conditional generations by SRGAN and SNGAN, respectively.

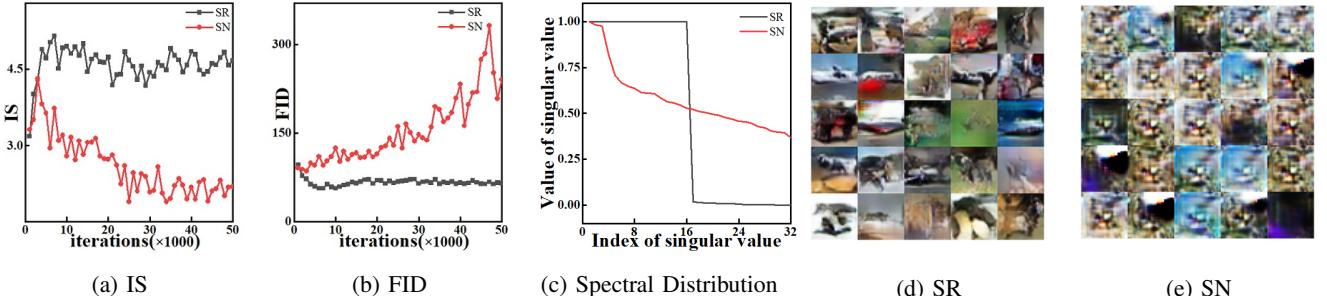


Fig. 7: Inception Score, Fréchet Inception Distance, spectral distribution and synthetic images of SN-GAN and SR-GAN for the setting  $C_{64-32}$

weight matrix. Readers are referred to Appendix for the details of the network architecture settings.

When applying spectral regularization, model performance is steadily improved, and stable training is guaranteed as shown in Table II . While for spectral normalization, SN-GANs still suffers from mode collapse. Worst of all, when channel size is small, mode collapse will happen to SN-GAN regardless of batch size as shown in our group  $C$  experiments. Figure 7 shows the training history of SN-GAN and SR-GAN for the setting  $C_{64-32}$ . It is seen that for SN-GAN, mode collapse has happened almost at the start of the training process and performance continues to deteriorate until eventually lead to mode collapse. In contrast, the performance of SR-GAN improves steadily and eventually converges (no mode collapse).

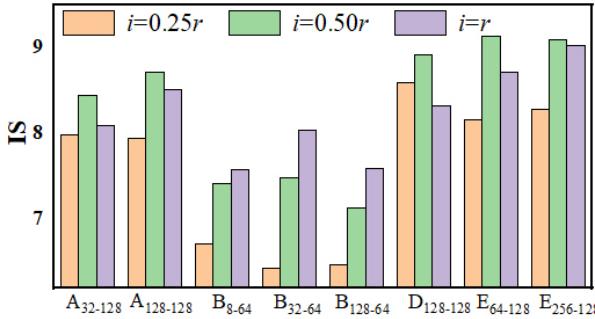
In the comparison of spectral distribution (Figure 7(c)), it is seen that spectral normalization cannot stop other singular values to drop significantly thus causing spectral collapse which in turn results in mode collapse. In contrast, static compensation in spectral regularization ensures that the first  $i$  singular values are equal in all cases, thus ensuring that

spectral collapse would not happen hence preventing mode collapse. Examples of generated images by the two training methods for this setting are also shown in the Figure 7(d) and (e). It is again clearly seen that mode collapse has indeed happened to SN-GAN while the images generated by SR-GAN are of better quality and more varieties.

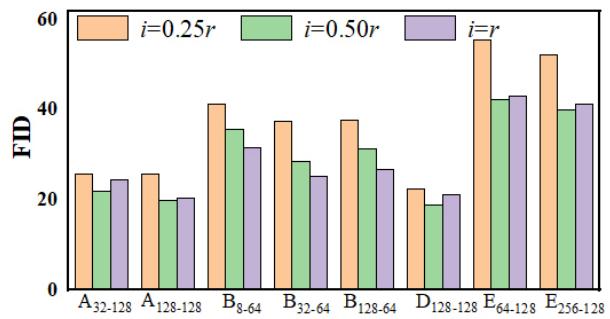
Through the example above, we can see that for experiment setting with too large batch size or too small channel size, SN-GANs cannot ensure stable training. Nevertheless, for various experiment setup spectral regularization can indeed guarantee training stability, and contribute to model performance, demonstrating that spectral regularization is a robust method for stable training.

### C. The Hyperparameter $i$ in SR-GANs

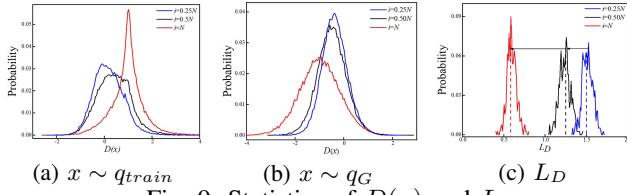
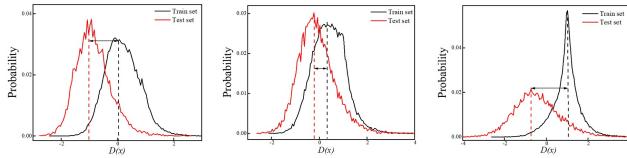
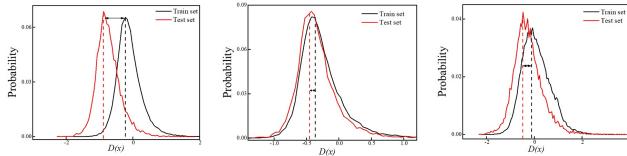
When applying static compensation in spectral regularization, there is a single hyperparameter  $i$ , and its value will affect performances. In the experiments above,  $i$  in SR-GANs is set to  $i = 0.5r$ , where  $r$  is the number of singular values. Clearly, when  $i = 1$ , SR-GAN is the same as SN-GAN, therefore SN-GAN is a special case of SR-GAN. To investigate the effect



(a) Inception Score

Fig. 8: The effect of  $i$  on model performance.  $N$  represents the number of singular values in corresponding weight matrix.

(b) Fréchet Inception Distance

Fig. 8: The effect of  $i$  on model performance.  $N$  represents the number of singular values in corresponding weight matrix.Fig. 9: Statistics of  $D(x)$  and  $L_D$ .Fig. 10: Statistics of  $D(x)$  with setting  $A_{128-128}$ .Fig. 11: Statistics of  $D(x)$  with setting  $B_{128-64}$ .

of  $i$ , we gradually increase  $i$ , and observe its influence on model performance. In Figure 8, we show the Inception Scores and Fréchet Inception Distances for different values of  $i$ . For experiment groups  $A, D$  and  $E$ , increasing  $i$  from  $0.25N$  to  $0.5N$ , the performances are improved. However, continuously increasing  $i$  from  $0.5N$  to  $N$ , the performances deteriorate. For experiments in group  $B$ , performances increase steadily with  $i$ .

To understand why  $i$  affects performances in this way, we feed the discriminator function with the generated data and real data from both the training and testing sets, and then record the statistics of  $D(x)$  in Equation (2) and the discriminator objective  $L_D$  in Equation (6). For explanation convenience, some typical results are illustrated here.

The probability distributions of  $D(x)$  for the generated data  $D(x)|_{x \sim q_G}$  and that for the training data  $D(x)|_{x \sim q_{train}}$  for the setting of  $A_{128-128}$  and different  $i$  values are shown in Figure 9 (a) and Figure 9 (b), respectively. Here  $q_{train}$  represents training set, and  $q_G$  represents generated set. The probability distributions of  $L_D$  is shown in Figure 9 (c).

When increasing  $i$  from  $0.25N$  to  $N$ , the distributions of

$D(x)|_{x \sim q_{train}}$  have a tendency of moving to the right, and at the same time the distributions of  $D(x)|_{x \sim q_G}$  have a tendency of moving to the left. This means that the discriminator can better discriminate between the real and generated samples. This is also verified by the distributions of  $L_D$  as can be clearly seen in Figure 9 (c).

To investigate discriminator's performance on the testing set, we show the probability distributions of  $D(x)|_{x \sim q_{train}}$  and  $D(x)|_{x \sim q_{test}}$  for the setting  $A_{128-128}$  in Figure 10, where  $q_{test}$  represents test set. It is seen that for  $i = 0.25r$  and  $i = 0.5r$ , the two distributions are more similar to each other than that of  $i = r$ . In the case of  $i = r$ , the discriminator behaves significantly differently between the training data and testing data, this means that overfitting has occurred and results in a drop in performances. In summary, Figure 9 and Figure 10 explain the performance drop for setting  $i = r$  in experiment groups  $A, D$  and  $E$ .

Furthermore, we monitor the statistics of  $D(x)$  for the settings in group  $B$  to explain why  $i$  affects the behaviors of SR-GANs as in Figure 8. The probability distributions of  $D(x)$  for the setting  $B_{128-64}$  are shown in Figure 11. We can see that for all the  $i$  values, the probability distributions of the discriminator output for the training and testing data agree well with each other, indicating no overfitting has occurred.

Although there is no systematic method for determining the best  $i$  value for different settings, our experiences is that setting  $i = 0.5r$  seems to work well. In a series of extensive experiments we conducted, setting  $i = 0.5r$ , SR-GANs always outperform SN-GANs and very importantly, we have not yet observed mode collapse.

#### D. Static Compensation vs Dynamic Compensation

Two kinds of compensation method are proposed in this paper: static and dynamic compensation. In Figure 6(d) and Figure 7(c), we show how these two compensation methods affect the spectral distribution.

Static compensation encourages to use the first  $i$  singular values. Therefore, spectral distributions mainly concentrate on the first  $i$  singular values, leaving the rest almost zero (Figure 7(c)). Meanwhile, the shape of spectral distributions are totally different from that in SN-GANs. Unlike static compensations, which try to carry out compensation at the largest scale, dynamic compensations focus on guaranteeing the shapes of

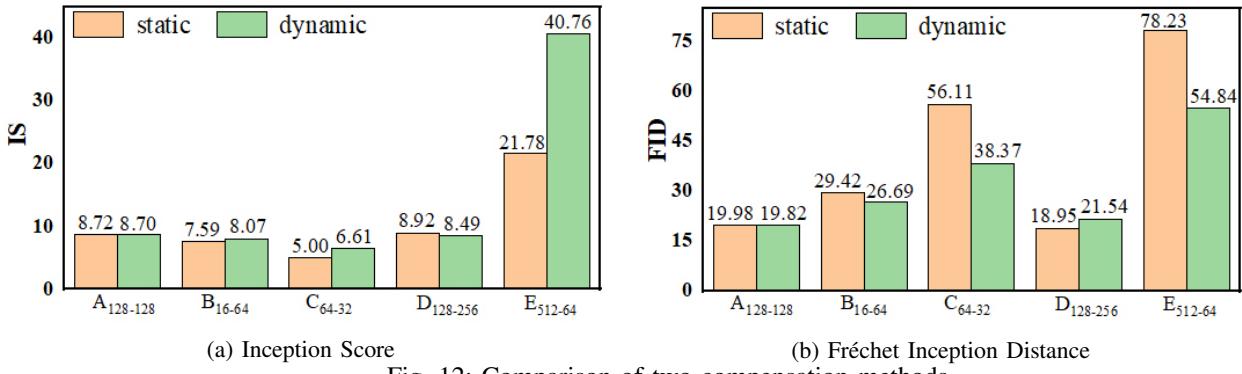


Fig. 12: Comparison of two compensation methods.

spectral distributions, and avoiding the occurrence of spectral collapse. As we can see in Figure 6(c), the shapes of spectral distributions hardly change at the end of training.

Recalling the Equation 5, static compensations averagely utilize the first  $i$  singular vectors, regardless of their differences. However, dynamic compensations consider the differences, and carry out compensation adaptively. To further show the difference between static and dynamic compensation, comparative trials are conducted based on settings  $A_{128-128}$ ,  $B_{16-64}$ ,  $C_{64-32}$ ,  $D_{128-256}$  and  $E_{512-64}$ . For these setting, two kinds of compensation methods are applied, and results are shown in Figure 12. static compensations achieve better results on unconditional generation with setting  $A_{128-128}$  and  $D_{128-256}$ . While dynamic compensations behave better in the rest tasks. Therefore, no convincing evidence shows dynamic compensation is better than static compensation, or the other way around. Our suggestion is that, for image generation on datasets of low resolutions and few categories, static compensation may works better. For image generation on datasets of high resolutions and more categories, dynamic compensation is better.

## V. CONCLUSIONS

In this paper, we monitor spectral distributions of the discriminator's weight matrices in SN-GANs. We discover that when mode collapse occurs to a SN-GAN, a large number of its weight matrices singular values will drop to very small values, and we introduce the concept of spectral collapse to describe this phenomenon. We have provided strong evidence to link mode collapse with spectral collapse. Based on such link, we have successfully developed a spectral regularization technique for training GANs. We show that by compensating the spectral distributions of the weight matrices, we can successfully prevent spectral collapse which in turn can successfully prevent mode collapse. In a series of extensive experiments, we have successfully demonstrated that preventing spectral collapse can not only avoid mode collapse but also can improve GANs performances.

## APPENDIX A PROOF OF COROLLARY 1

**Corollary 1.** If a linear function  $f = Wx$  satisfies Lipschitz constraint:  $\|f(x_1) - f(x_2)\| \leq \|x_1 - x_2\|$ , then the supremum

of the Lipschitz constraint is obtained when all the singular values of the weight matrix  $W$  are 1.

**Proof:** Because  $f$  is a linear function:  $f(x) = Wx$ . The 1-Lipschitz constraint for  $f$  can be expressed as:

$$\|Wx\| \leq \|x\| \quad (12)$$

Equation 12 is equivalent to:

$$\|Wx\|^2 \leq \|x\|^2 \quad (13)$$

and,

$$\|Wx\|^2 = x^T W^T W x = x^T V \Sigma V^T x \quad (14)$$

where columns of  $V$ ,  $[v_1, \dots, v_n]$  are eigenvectors of  $W^T W$ , and diagonal entries of diagonal matrix  $\Sigma$  are eigenvalues of  $W^T W$ .

Taking  $y = V^T x$ , then

$$\|Wx\|^2 = y^T \Sigma y = \lambda_1 y_1^2 + \dots + \lambda_n y_n^2 \quad (15)$$

where  $\lambda_i$  is the  $i$ -th eigenvalue, and  $y_i$  is the  $i$ -th element of  $y$ .

Because  $W^T W$  is symmetric,  $V^T = V^{-1}$ , then

$$\|y\|^2 = y^T y = x^T V V^T x = x^T x = \|x\|^2 \quad (16)$$

Finally,  $\|Wx\|^2 \leq \|x\|^2$  is equivalent to  $\lambda_1 y_1^2 + \dots + \lambda_n y_n^2 \leq y_1^2 + \dots + y_n^2$ . We can see that the upper bound of 1-Lipschitz constraint can be obtained only when all eigenvalues of  $W^T W$  are 1. In other words, all the singular values of  $W$  are 1.

## APPENDIX B ARCHITECTURE AND OPTIMIZATION SETTINGS

In this paper, we employ SN-GAN architecture for image generation task. To better illustrate how we change the channel size in the discriminator architecture, we show the architecture details in Figure 13. The weight in the convolutional layer is in the format  $[out, in, h, w]$ , where  $out$  is the output channel,  $in$  represents the input channel,  $h$  and  $w$  are kernel sizes.

Particularly, there are 10 convolutional layers ( $layer\_0 \sim layer\_9$ ) in  $D$  network for image generation on CIFAR-10 and STL-10, and 17 convolutional layers ( $layer\_0 \sim layer\_16$ ) in  $D$  network for image generation on ImageNet. ch in Figure 13 corresponds to channel size of discriminator function in main text, where extensive experiments are conducted with different settings of ch. All the experiments are conducted based on the following architecture. Image generation on STL-10 shares the

same architecture with that on CIFAR-10. Thus, images in STL-10 are compressed to  $32 \times 32$  pixels, identical to the resolution of images in CIFAR-10. For conditional generation on ImageNet, images are compressed to  $128 \times 128$  pixels.

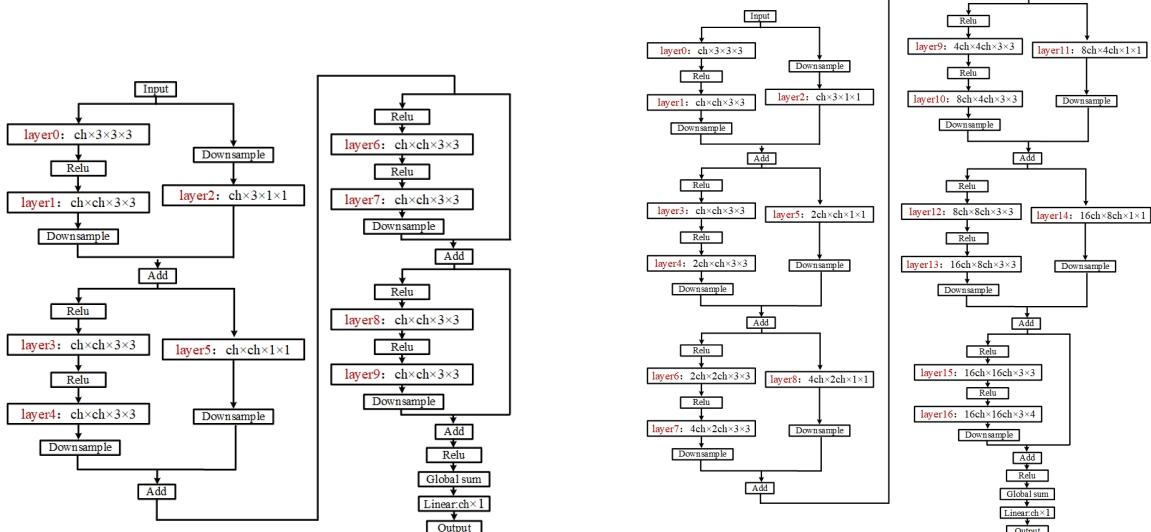
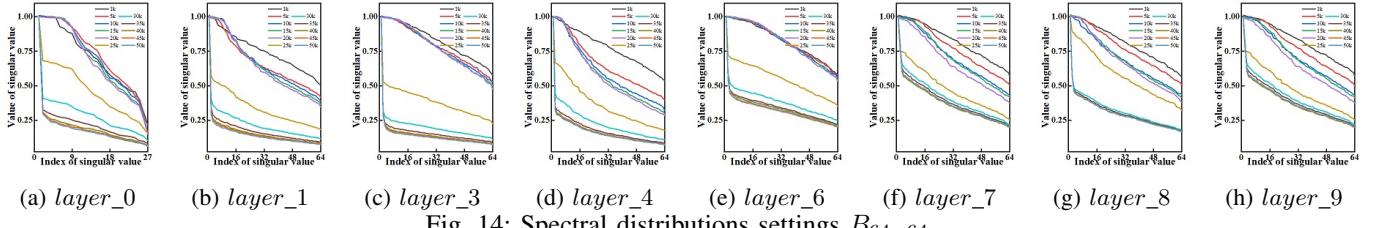
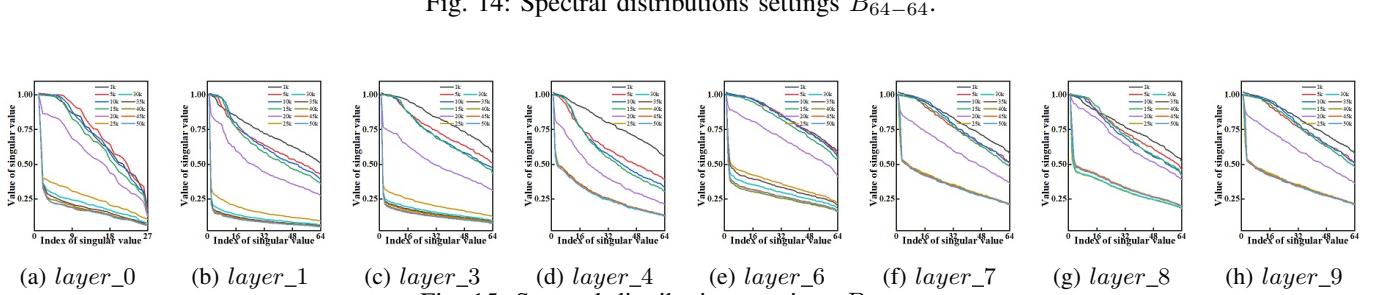
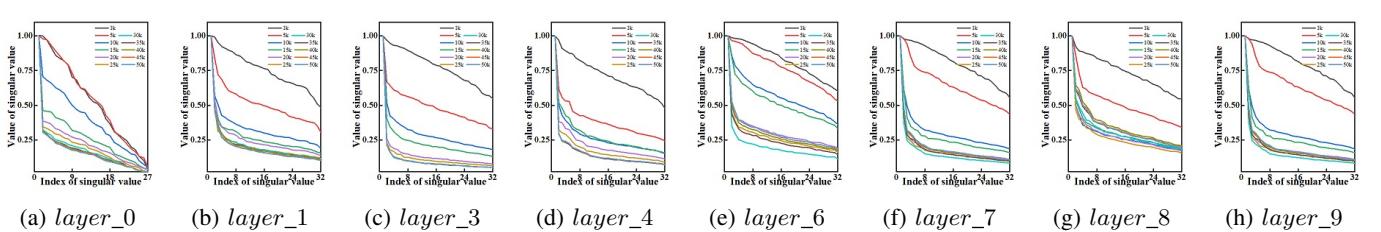
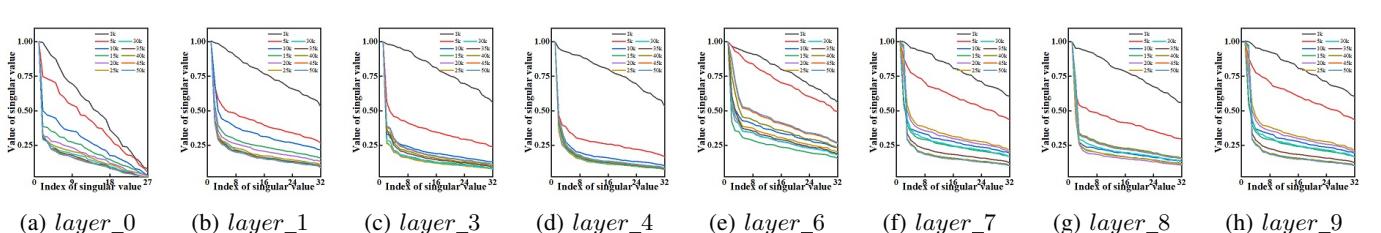
The optimization settings follow SN-GANs. To be specific, for image generation on CIFAR-10 and STL-10, the learning rate is taken as 0.0002, the number of updates of the discriminator per one update of the generator  $n_{critic}$  is 5, and Adam optimizer is used as the optimization with the first and second order momentum parameters as 0 and 0.9, respectively. For image generation on ImageNet, the learning rate for  $G$  and  $D$  is taken as 0.0001 and 0.0004, respectively, the number of updates of the discriminator per one update of the generator  $n_{critic}$  is 1. Adam optimizer is used as the optimization with the first and second order momentum parameters as 0 and 0.9, respectively. And, spectral normalization is applied in  $G$  architecture. To alleviate the huge demand for computational facility, we apply the trick of gradient accumulation, which is proposed in the implementation of BigGANs.

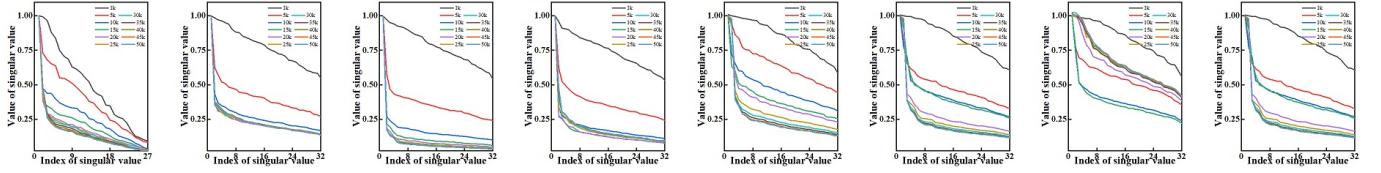
## APPENDIX C SPECTRAL DISTRIBUTION

In Figure 15 ~ Figure 22, we show the spectral distributions for settings, which suffer from mode collapse. We can see that spectral collapse and mode collapse always go side by side. For image generation on CIFAR-10 and STL-10, spectral collapse is observed in all layers except *layer\_2* and *layer\_5*, which act as the role of skip connection, as shown in Figure 13. While for image generation on ImageNet, spectral collapse is observed in all layers except *lay\_2*.

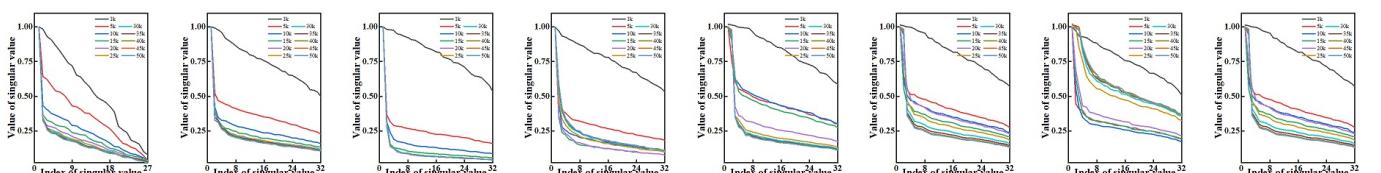
## REFERENCES

- [1] I. Goodfellow, J. Pouget-Abadie, and M. Mirza, “Generative adversarial nets,” *Advances in neural information processing systems*, pp. 2672–2680, 2014.
- [2] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv*, p. 1511.06434, 2015.
- [3] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” *arXiv preprint arXiv*, p. 1701.07875, 2017.
- [4] J. Wu, Z. Huang, and J. Thoma, “Energy-relaxed wasserstein gans (energywgan): Towards more stable and high resolution image generation,” *arXiv preprint arXiv*, p. 1712.01026, 2017.
- [5] D. Berthelot, T. Schumm, and L. Metz, “Began: Boundary equilibrium generative adversarial networks,” *arXiv preprint arXiv*, p. 1703.10717, 2017.
- [6] X. Mao, Q. Li, and H. Xie, “Least squares generative adversarial networks,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821, 2017.
- [7] B. Andrew, J. Donahue, and K. Simonyan, “Large scale gan training for high fidelity natural image synthesis,” *arXiv preprint arXiv*, 2018.
- [8] I. Gulrajani, F. Ahmed, and M. Arjovsky, “Improved training of wasserstein gans,” *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017.
- [9] G. Qi, “Loss-sensitive generative adversarial networks on lipschitz densities,” *arXiv preprint arXiv*, p. 1701.06264, 2017.
- [10] T. Miyato, T. Kataoka, and M. Koyama, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv*, p. 1802.05957, 2018.
- [11] T. Salimans and D. Kingma, “Weight normalization: A simple reparameterization to accelerate training of deep neural networks,” *Advances in Neural Information Processing Systems*, pp. 901–909, 2016.
- [12] J. Heinonen, “Lectures on lipschitz analysis,” *University of Jyvaskyla*, 2005.
- [13] A. Brock, T. Lim, and J. M. Ritchie, “Neural photo editing with introspective adversarial networks,” *arXiv preprint arXiv*, p. 1609.07093, 2016.
- [14] A. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for non-parametric object and scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 901–909, 2008.
- [15] M. Heusel, H. Ramsauer, and T. Unterthiner, “Gans trained by a two time-scale update rule converge to a nash equilibrium,” *arXiv preprint arXiv*, p. 1706.08500, 2017.
- [16] J. Deng, W. Dong, R. S. L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.
- [17] T. Miyato and M. Koyama, “cgans with projection discriminator,” *arXiv preprint arXiv*, p. 1808.05637, 2018.
- [18] T. Salimans, I. Goodfellow, and W. Zaremba, “Improved techniques for training gans,” *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

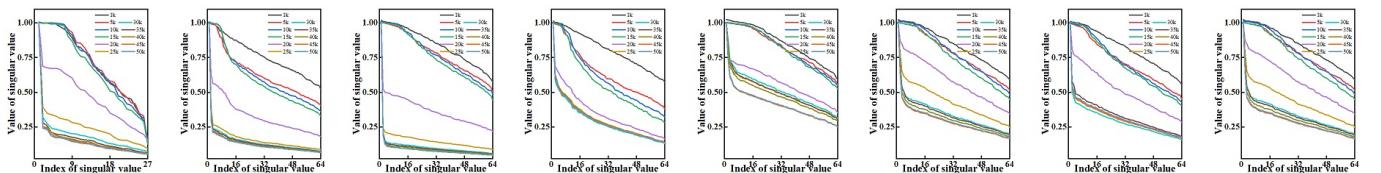
(a)  $D$  architecture for CIFAR-10 and STL-10Fig. 13: Architecture of  $D$ .(a)  $layer_0$ (b)  $layer_1$ (c)  $layer_3$ (d)  $layer_4$ (e)  $layer_6$ (f)  $layer_7$ (g)  $layer_8$ (h)  $layer_9$ (a)  $layer_0$ (b)  $layer_1$ (c)  $layer_3$ (d)  $layer_4$ (e)  $layer_6$ (f)  $layer_7$ (g)  $layer_8$ (h)  $layer_9$ (a)  $layer_0$ (b)  $layer_1$ (c)  $layer_3$ (d)  $layer_4$ (e)  $layer_6$ (f)  $layer_7$ (g)  $layer_8$ (h)  $layer_9$ (a)  $layer_0$ (b)  $layer_1$ (c)  $layer_3$ (d)  $layer_4$ (e)  $layer_6$ (f)  $layer_7$ (g)  $layer_8$ (h)  $layer_9$ Fig. 17: Spectral distributions settings  $C_{16-32}$ .



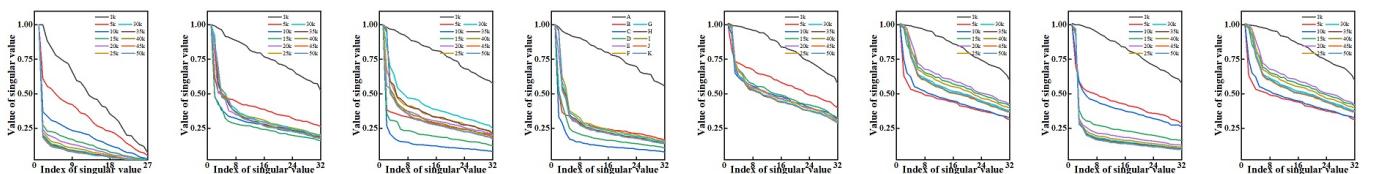
(a) layer\_0 (b) layer\_1 (c) layer\_3 (d) layer\_4 (e) layer\_6 (f) layer\_7 (g) layer\_8 (h) layer\_9

Fig. 18: Spectral distributions settings  $C_{32-32}$ .

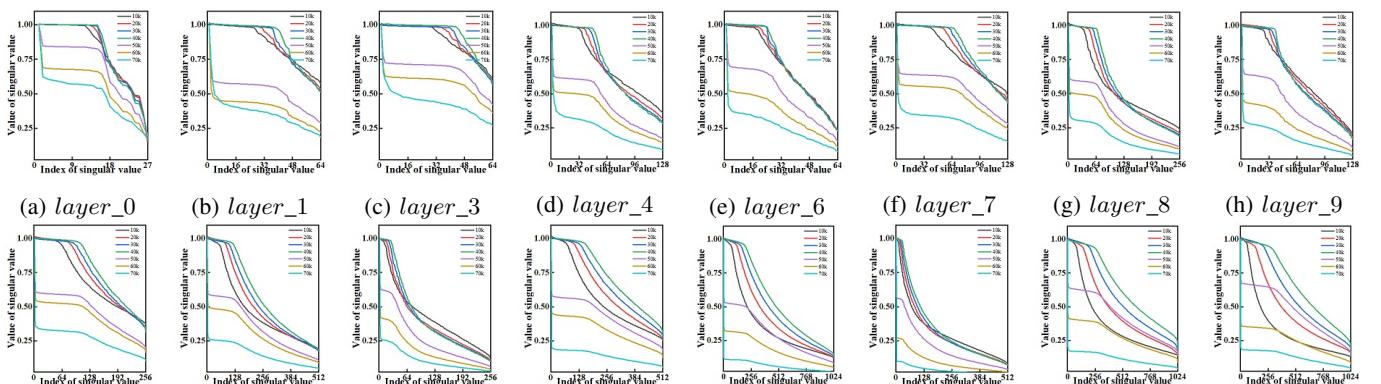
(a) layer\_0 (b) layer\_1 (c) layer\_3 (d) layer\_4 (e) layer\_6 (f) layer\_7 (g) layer\_8 (h) layer\_9

Fig. 19: Spectral distributions settings  $C_{64-32}$ .

(a) layer\_0 (b) layer\_1 (c) layer\_3 (d) layer\_4 (e) layer\_6 (f) layer\_7 (g) layer\_8 (h) layer\_9

Fig. 20: Spectral distributions settings  $E_{256-64}$ .

(a) layer\_0 (b) layer\_1 (c) layer\_3 (d) layer\_4 (e) layer\_6 (f) layer\_7 (g) layer\_8 (h) layer\_9

Fig. 21: Spectral distributions settings  $E_{256-32}$ .

(i) layer\_0 (j) layer\_1 (k) layer\_3 (l) layer\_4 (m) layer\_6 (n) layer\_7 (o) layer\_8 (p) layer\_9

Fig. 22: Spectral distributions settings  $E_{2048-64}$ .