The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)
February 20–27, 2024, Vancouver, Canada

RG-GAN: Dynamic Regenerative Pruning for Data-Efficient Generative Adversarial Networks

Divya Saxena[1], **Jiannong Cao**[1], **Jiahao Xu**[2], **Tarun Kulshrestha**[3]
[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]Department of Computer Science and Engineering, University of Nevada, Reno, USA
[3]University Research Facility of Big Data Analytics, The Hong Kong Polytechnic University, Hong Kong
divsaxen@comp.polyu.edu.hk, csjcao@comp.polyu.edu.hk, jiahaoxxuu@gmail.com, tarun.kulshrestha@polyu.edu.hk

Abstract

Training Generative Adversarial Networks (GAN) to generate high-quality images typically requires large datasets. Network pruning during training has recently emerged as a significant advancement for data-efficient GAN. However, simple and straightforward pruning can lead to the risk of losing key information, resulting in suboptimal results due to GAN's competitive dynamics between generator (G) and discriminator (D). Addressing this, we present RG-GAN, a novel approach that marks the first incorporation of dynamic weight regeneration and pruning in GAN training to improve the quality of the generated samples, even with limited data. Specifically, RG-GAN initiates layer-wise dynamic pruning by removing less important weights to the quality of the generated images. While pruning enhances efficiency, excessive sparsity within layers can pose a risk of model collapse. To mitigate this issue, RG-GAN applies a dynamic regeneration method to reintroduce specific weights when they become important, ensuring a balance between sparsity and image quality. Though effective, the sparse network achieved through this process might eliminate some weights important to the combined G and D performance, a crucial aspect for achieving stable and effective GAN training. RG-GAN addresses this loss of weights by integrating learned sparse network weights back into the dense network at the previous stage during a follow-up regeneration step. Our results consistently demonstrate RG-GAN's robust performance across a variety of scenarios, including different GAN architectures, datasets, and degrees of data scarcity, reinforcing its value as a generic training methodology. Results also show that data augmentation exhibits improved performance in conjunction with RG-GAN. Furthermore, RG-GAN can achieve fewer parameters without compromising, and even enhancing, the quality of the generated samples. Code can be found at this link: https://github.com/IntellicentAI-Lab/RG-GAN

Introduction

Generative Adversarial Networks (GAN) (Goodfellow, Mirza, et al. 2014; Saxena et al. 2021) have achieved remarkable progress in generating high-quality images. They have become essential in various visual tasks, such as domain adaptation (Hoffman et al. 2018; Hsu et al. 2020), image painting (Cheng et al. 2022), and image-to-image translation (Lee et al. 2018; Zhu et al. 2017). However, the effectiveness of GAN heavily relies on having large and diverse datasets but gathering such data can be time-intensive and difficult (Kalibhat, Balaji, and Feizi 2021). Figure 1 shows StyleGAN2 shows artifacts with limited

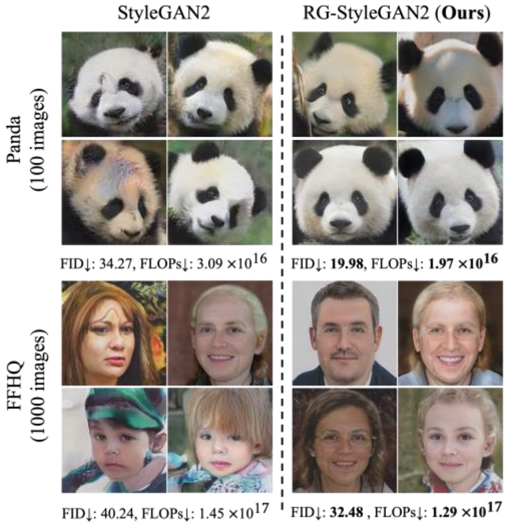

StyleGAN2
RG-StyleGAN2 (**Ours**)
Panda
(100 images)
FID↓: 34.27, FLOPs↓: 3.09 ×10^16
FID↓: **19.98**, FLOPs↓: **1.97 ×10^16**
FFHQ
(1000 images)
FID↓: 40.24, FLOPs↓: 1.45 ×10^17
FID↓: **32.48** , FLOPs↓: **1.29 ×10^17**

Figure 1: Comparative results of StyleGAN2 and our method (RG-StyleGAN2) on two datasets – Panda of 256×256 resolution (trained only on 100 images) and FFHQ of 1024×1024 resolution (trained only on 1000 images). While StyleGAN2 shows artifacts with limited data, our RG-StyleGAN2 leverages dynamic regeneration with pruning to boost image quality, improve data-efficiency, and reduce computational costs (best viewed in color).

data. To address the challenges posed by limited data, numerous techniques (Karras et al. 2018; Karras, Laine, and Aila 2019) have been introduced. Dynamic data augmentation (Zhao et al. 2020; Jiang et al. 2021) serves as a stabilizing mechanism for GAN training with limited data. A recent addition is the lottery ticket hypothesis (LTH) for GAN, termed "GAN tickets", serves as a complementary strategy to existing augmentation approaches. LTH shows that identified sparse sub-networks, i.e., winning tickets, when trained separately can match or even surpass the performance of unpruned models. However, identifying these GAN tickets (T. Chen et al. 2021; X. Chen et al. 2021) necessitates repeated, resource-intensive train-prune-retrain cycles, leading to longer training time and increased floating-point operations (FLOPs) compared to conventional dense GAN training.

Recently, pruning during training has been introduced (Saxena et al. 2023) to investigate sparse sub-networks for data-efficient GAN. However, simple and straightforward pruning can lead to risk of losing key information, resulting in suboptimal results due to GAN's competitive dynamics. Unlike traditional networks, GAN have an evolving training landscape due to adversarial training: weights deemed not important early in the training may become vital later, potentially being prematurely pruned. Different from existing works, we ask a question:

Can we achieve enhanced stability and efficiency in GAN training for high-quality generation, even with limited data, by strategically regenerating prematurely pruned weights?

Addressing this, in this paper, we propose, RG-GAN, a novel approach that marks the first incorporation of dynamic weight regeneration and pruning in *GAN training* to improve the quality of the generated samples even with limited data. Here, *regeneration* refers to the process of reintroducing or regenerating previously pruned weights in the GAN training process. This allows the network to retain important structural information that might have been lost during the pruning process. This is of utmost need in data-limited situations as data is not diverse and if key features are mistakenly pruned, they can be recovered through regeneration.

RG-GAN employs layer-wise dynamic pruning of weights (J. Liu et al. 2020), which dynamically removes weights based on their importance to the quality of the generated images. While pruning helps in making the model more efficient, it can also create a problem. If too many weights are removed, layers may become too sparse, which can lead to training instability or, in the worst case, a model collapse. To tackle this risk, regeneration allows for the recovery of pruned weights within the specific sparse layer and makes them part of the training process once again if they become important, named *recurrent regeneration*. By regenerating pruned weights, network is able to retain some of the original structure, which prevents network from collapsing or diverging during training. In this way, the model remains effective while also being efficient.

While effective and efficient, the sparse network achieved through dynamic pruning and recurrent regeneration process might eliminate weights that are essential to the collaborative performance of Generator (G) and Discriminator (D), a balance that is crucial for effective and robust GAN training. RG-GAN addresses this loss of weights by integrating learned sparse network weights back into the dense network at the previous stage during a follow-up regeneration step, named *augmented regeneration*. This procedure ensures that key network weights are not permanently lost, preserving their collective importance to the GAN's performance. Our contributions can be summarized as follows:

- We propose an improved GAN training methodology, RG-GAN, that fuses dynamic pruning and regeneration during the training process. This innovation facilitates the optimization of network structures, resulting in high-quality image generation, even in data-limited situations.
- RG-GAN applies layer-wise dynamic pruning, considering each weight's contribution to image quality. To prevent model collapse from over-pruning, a regeneration step is implemented to reintroduce some weights if they gain importance. Furthermore, to maintain equilibrium between G and D, an additional regeneration step is proposed to reintegrate learned sparse network weights back into its prior dense network.
- We conduct extensive experiments to demonstrate the **four merits** of the proposed dynamic pruning and regeneration-based GAN training methodology. **First**, our method is robust, working well with a wide range of GAN architectures, and datasets of different resolutions (32×32, 64×64, 256×256, and 1024×1024) and data constraints (ranging from 10% to 100% of training data, and many few-shot datasets), reinforcing its value as a generic training methodology. **Second**, generating higher-quality samples both in regular and low-data regime setups. **Third**, our method provides an alternative to the GAN tickets and progressive growing method. **Finally**, removing unimportant weights through pruning also leads to reduced number of parameters.

Methodology

Figure 2 illustrates the training process. The key features of our training process include, (1) dynamic pruning followed by recurrent regeneration; and (2) augmented regeneration.

Dynamic Pruning and Recurrent Regeneration (RR). As shown in Algorithm 1, we start from a randomly initialized dense G and D structures and prune the networks by identifying and eliminating less important parameters. This dynamic, layer-wise pruning adjusts the pruning threshold based on the network's performance and gradients. We represent the parameters of G and D with a set $\mathbf{W}i$, where i ranges from 1 to $\mathbf{C}$, representing the layer in the network. Pruning applies a binary mask $\mathbf{M}$ to each parameter $\mathbf{W}$, giving rise to the sparse structure of the network. This is done by defining a trainable pruning threshold vector $\mathbf{t} \in \mathbb{R}c_o$ for each parameter matrix $\mathbf{W} \in \mathbb{R}c_{o\times}c_i$, and then using a unit step function ($\mathbf{S}(\mathbf{x})$) to calculate the masks based on the magnitudes of the parameters and their respective thresholds. $\mathbf{S}(\mathbf{x})$ maps positive values (active weights > $\boldsymbol{t}$) to 1, and non- positive values to 0. Each network layer undergoes this process independently, learning individual thresholds, allowing dynamic, task-specific sparsity level adjustments.

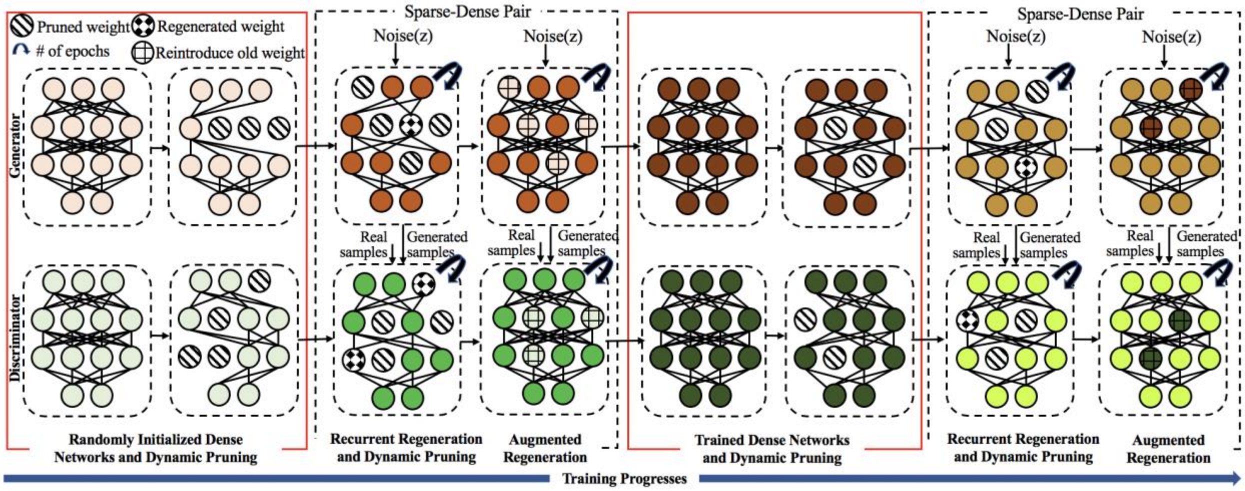


Figure 2: Overview of the RG-GAN training methodology. Best viewed in color.

$$\mathbf{M}ij = \mathbf{S}(|\mathbf{W}_{ij}| - \boldsymbol{t}_i), \text{ for } 1 \le i \le c_o,\ 1 \le j \le c_i \qquad (1)$$

$\mathbf{S}(\mathbf{x})$ is non-differentiable and unsuitable for gradient computation, we use a differentiable approximation function, denoted as $\mathbf{H}(\mathbf{x})$ (Xu and Cheung 2020). This function permits gradient computations for pruned weights, enabling their potential reactivation if found beneficial for network performance. During the backward pass, all pruned weights are reactivated for gradient computation. A masking operation multiplies the computed gradients with the mask $\mathbf{M}$, generated during the forward pass for the gradient update, i.e., no gradient update for reactivated weights in this pass.

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} \odot \mathbf{M}_{ij} \qquad (2)$$

In the subsequent forward pass, reactivated weights may gain importance based on gradients computed in the previous backward pass, leading to their inclusion in the training, i.e., recurrent regeneration.

This process is cyclical, where weights can be pruned, and subsequently reinstated multiple times based on their evolving contribution to the network's performance. This dynamic, recurring cycle fosters an adaptable network structure capable of self-optimizing for a specific task.

We incorporate a sparse regularization term $\mathbf{R}$ in the training loss to encourage a higher level of sparsity in mask $\mathbf{M}$ by penalizing low threshold values. For a trainable masked layer with a threshold $\boldsymbol{t} \in \mathbb{R}c_o$, the regularization term is $\mathbf{R} = \sum_{i=1}^{Co} exp(-\boldsymbol{t}_i)$. The RG-GAN objective is as follows:

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{x\sim p_{data}}[f_D(D(x, \theta_D \odot M_D))]$$
$$+ \mathbb{E}_{z\sim p_z}[f_G(D(G(z, \theta_G \odot M_G)))] + \lambda \sum_{i=1}^{C} R_i \qquad (3)$$

where, $\boldsymbol{\theta}_\mathbf{G}$, $\mathbf{M}_\mathbf{G}$, $\boldsymbol{\theta}_\mathbf{D}$, and $\mathbf{M}_\mathbf{D}$ represent the weights and masks for the parameter space of G and D. $\odot$ denotes Hadamard product and $\boldsymbol{\lambda}$ is the penalty term promoting sparsity. The notations f_D and f_G represent the mapping functions from which various GAN losses can be derived.

RG-GAN employing dynamic pruning and recurrent regeneration, yields a sparse yet efficient model of G and D networks. This process focuses on preserving important weights, enhancing the network's performance without compromising efficiency. Although sparse model demonstrates efficiency, it may be restrictive in terms of its capacity to represent complex features and maintain equilibrium during the training. In order to address this potential limitation, we introduce the process of augmented regeneration.

Augmented Regeneration. The process of augmented regeneration involves determining (1) *when to apply augmented regeneration during the training?* and (2) *what values should be assigned to the regenerated weights in order to regenerate a dense neural network of G and D?*

This phase effectively restores all previously pruned connections, marking the transition from a pruned to a regrown dense network ready for further training. Augmented regeneration is applied when current epoch's FID is greater than the weighted average FID value calculated over the previous n (n=5 in all experiments) epochs. This increase in the FID value indicates a decline in the quality of the generated images, suggesting the network may have pruned some important connections. In this case, pruned weights are reactivated with their old weights which provides a common starting point for both G and D. This can increase the likelihood of finding the equilibrium point again and results in more stable training.

Algorithm 1: RG-GAN training

Input: Discriminator $D(x, \theta_D)$ with threshold t_D; Generator $G(z, \theta_G)$ with threshold t_G; training epochs K; sparse training mode indicator I; evaluate frequency e.
Output: Converged G and D

```
Randomly initialize weights $\theta_G \in R^N, \theta_D \in R^N$, and initialize $t_G$, $t_D$ to zero
$I \leftarrow$ True
for t = 1, 2, ..., $K$ do
  if $I$ then
    $\{\hat{\theta}_G, \hat{\theta}_D\} \leftarrow$ Dynamic pruning($\{\theta_G, \theta_D\}$) by Eq 1 & 2
    Compute loss by Eq (3) with $\{\hat{\theta}_G, \hat{\theta}_D\}$
    Update $\{\hat{\theta}_G, \hat{\theta}_D\}$, and $\{t_G, t_D\}$
    $\{\hat{\theta}_G, \hat{\theta}_D\} \leftarrow$ Recurrent regeneration($\{\hat{\theta}_G, \hat{\theta}_D\}$,$\{t_G, t_D\}$)
  else
    Compute loss by Eq 3 with $\{\hat{\theta}_G, \hat{\theta}_D\}$ & update $\{\theta_G, \theta_D\}$
  end if
  if mod ($t$, $e$) = 0 then
    Calculate current FID $F_c$, weighted FID $F_w$
    if $F_c > F_w$ then
      $I \leftarrow$ False                #activates dense training
      $\{\theta_G, \theta_D\} \leftarrow$ Augmented regeneration ($\{\hat{\theta}_G, \hat{\theta}_D\}$)
    else
      $I \leftarrow$ True
    end if
  end if
end for
```

Experiments

In this section, we extensively evaluate the effectiveness of our regeneration-based GAN training for image generation and further analyze our method. For 1024×1024 resolution, experiments are performed on NVIDIA RTX A6000 GPU (48GB), while for remaining resolutions, NVIDIA RTX 3090 GPU (24GB) is used.

Base Models. We apply our method on the unconditional **ProGAN** (Karras et al. 2018), **StyleGAN2** (Karras et al. 2020), **SNGAN** (Miyato et al. 2018), **FastGAN** (B. Liu et al. 2021), and **GAN tickets** (T. Chen et al. 2021). We experiment with three distinct loss functions for the d_{critic} function namely, WGAN-GP (Gulrajani et al. 2017) used by ProGAN, Hinge loss used by SNGAN, and non-saturating GAN loss with 1-sided GP (Goodfellow, Pouget-Abadie, et al. 2014) used by StyleGAN2. These models were chosen to compare with RG-GAN due to their diversity in structures, SOTA performance, the utilization of different loss functions, and their broad use-cases. This comparison allows for a comprehensive evaluation of RG-GAN's adaptability and robustness, across varying architectures and loss functions.

Datasets: RG-GAN's effectiveness is exhaustively tested through evaluation on a wide spectrum of datasets, encompassing various resolutions 32×32, 64×64, 256×256 and 1024×1024. These datasets range from the commonly used **CIFAR-10** (32×32) and **Tiny-ImageNet** (64×64) (on 10%, 20%, 50% and 100% of training dataset), to specific **animal face (AF) datasets** (256×256)(Si and Zhu 2011) (**Dog** (389 images) and **Cat** (160 images)), **100-shot datasets** (256×256) (Zhao et al. 2020) (**Obama**, **Panda**, and **Grumpy-Cat**), and the varying size of high-resolution **FFHQ** (1024×1024) (Karras, Laine, and Aila 2019). This broad testing landscape underscores RG-GAN's ability to perform consistently across different domains and scales.

Evaluation Metrics. We use the widely adopted metric, **Frechet Inception Distance (FID)** (Heusel et al. 2017) which assesses generated image quality by comparing feature statistics from real and generated data, using an intermediate layer of a pre-trained Inception network. FID is consistent with human judgment, and a low FID indicates higher-quality images. We also provide the **number of real images (#RI)**(in Million (M) or in Thousand (K)) (Karras et al. 2018; Miyato et al. 2018) and **FLOPs** to show the efficiency of our model. If model *A* is taking less number of RIs and achieving comparable or better results than model *B*, it shows that model *A* is time-efficient. To compute results, we take an average over three runs.

Image Generation

In this section, we evaluate our proposed RG-GAN's capability in handling distinct types of diversity present in the image datasets: inter-class and intra-class diversity.

Evaluation on Inter-Class Diversity. Our RG-GAN performance was tested on CIFAR-10 and Tiny-ImageNet datasets, which embodies significant inter-class diversity due to the presence of multiple distinct categories. Notably, our model showed considerable improvements across varying data availability levels (100%, 50%, 20%, and 10%) when compared to ProGAN and StyleGAN2 models at 32×32 and 64×64 resolutions. As depicted in Table 1, these improvements were particularly notable in only 10% data scenarios, with gains of 5.78% and 2.08% in FID on CIFAR-10, and 7.54% and 18.30% in FID on Tiny-ImageNet for RG-ProGAN and RG-StyleGAN2, respectively. This improvement can be attributed to our method's unique capability of pruning and regenerating weights dynamically, enabling the model to adapt to new data and progressively refine its understanding of the task.

Evaluation on Intra-Class Diversity. To further assess the method's robustness, we evaluate its performance on the FFHQ dataset. This dataset offers considerable intra-class diversity, as it comprises various human faces, each with distinct features, yet all belonging to the same class-human faces. This evaluation provides a test of our model's capacity to handle diversity within a single class at high-resolution. Furthermore, RG-GAN superior performance on FFHQ is evident in the improved FID across all data sizes, i.e., 1k, 5k, 10k, and 70k (Table 2). This further attests to the effectiveness and robustness of RG-GAN when dealing with intra-class diversity and high-resolution data.

Dataset	Models	100% data		50% data		20% data		10% data	
		#RI (M) ↓	FID↓	#RI (M) ↓	FID↓	#RI (M) ↓	FID↓	#RI (M) ↓	FID↓
CIFAR-10 (32×32)	ProGAN	19.5	26.14	10.5	28.33	7.2	28.64	5.6	30.08
	RG-ProGAN	**15.9**	**24.61**	**9.4**	**25.20**	**5.9**	**26.50**	**5.1**	**28.34**
	StyleGAN2	17.7	13.19	6.4	17.22	4.1	29.77	3.2	36.45
	RG-StyleGAN2	**11.3**	**11.97**	**4.7**	**16.50**	**3.3**	**26.32**	**1.3**	**31.69**
Tiny-ImageNet (64×64)	ProGAN	9.4	40.27	9.6	49.0	8.0	77.86	7.7	83.27
	RG-ProGAN	**9.1**	**37.15**	**10.4**	**44.90**	**7.4**	**73.55**	**7.4**	**76.99**
	StyleGAN2	19.2	20.95	14.5	30.03	5.1	65.26	3.2	84.86
	RG-StyleGAN2	**16.5**	**20.07**	**11.0**	**26.14**	**4.6**	**53.61**	**2.7**	**69.33**

Table 1. FID comparison on datasets at 32×32 and 64×64 resolution. FID is calculated using 50k randomly generated samples, with the test data (10k) serving as the reference distribution.

Models	70k	10k	5k	1k
StyleGAN2	4.35	13.06	21.76	40.24
RG-StyleGAN2	**4.12**	**11.64**	**18.33**	**32.48**

Table 2. FID↓ on FFHQ dataset across varying size at 1024×1024 resolution. FID is calculated using 50k randomly generated samples, with training data (70k) serving as the reference distribution.

Few-Shot Generation

Results from Table 3 show that RG-GAN models consistently outperform their counterparts (ProGAN and StyleGAN2) across different few-shot datasets, indicating RG-GAN's superior image generation capabilities. For instance, on the 100-shot Obama dataset, RG-StyleGAN2 achieved a significantly lower FID (71.09) than StyleGAN2 (86.67). We further diversify the data through an advanced augmentation technique, Differentiable Augmentation (DA) (Zhao et al. 2020). This combined with RG-GAN's ability to dynamically adapt network structure through regeneration, led to improved performance. It indicates RG-GAN's complementary nature with such augmentation techniques.

We also compare with FastGAN which is recognized for its stability and speed in few-shot image synthesis (see Table 4). Notably, RG-FastGAN manages to produce better image quality (FID↓) with fewer FLOPs↓ and #RI shown to D↓, as observed across all few-shot datasets. This shows that RG-GAN can offer a better balance of efficiency and performance for few-shot image synthesis at high resolutions.

GAN Tickets

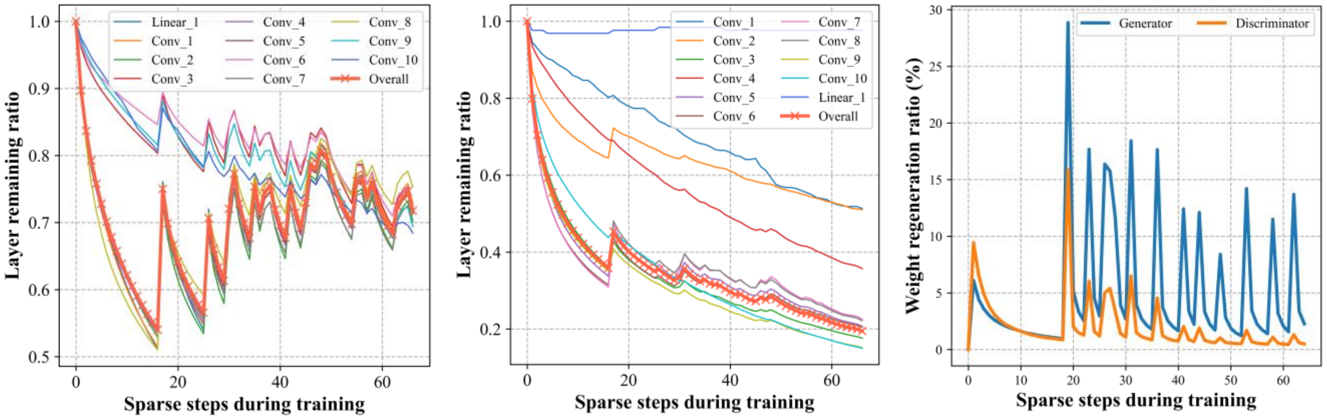
Recently, Iterative Magnitude Pruning (IMP) (T. Chen et al. 2021) has demonstrated its effectiveness in identifying "lottery tickets" in GAN compared to other pruning methods (Han, Mao, and Dally 2015; Z. Liu et al. 2017). We use IMP at 20% and 46% pruning ratio on the full CIFAR-10 data on SNGAN, i.e., ST_SNGAN@20% and ST_SNGAN@46%, respectively. The results are summarized in Table 5. Results show that RG-GAN surpasses GAN tickets in terms of FID. Results also show that discovering GAN tickets can be time-consuming (high #RI). Furthermore, RG-GAN consumes fewer FLOPs during training as compared to both GAN tickets and base SNGAN model.

Models	Obama	G-Cat	Panda	Animal Face	
				Cat	Dog
# of images	100	100	100	160	389
ProGAN	129.5	135.8	235.8	289.7	259.3
+RG	**108.6**	**123.2**	**152.5**	**279.8**	**225.3**
StyleGAN2	86.6	51.3	34.2	81.9	157.6
+RG	**71.0**	**36.3**	**19.9**	**71.2**	**130.3**
+DA	46.3	28.6	12.8	42.8	58.4
+DA+RG	**44.8**	**27.3**	**12.2**	**42.1**	**56.7**

Table 3. FID on few-shot datasets at 256×256 resolution. FID is calculated using 5k randomly generated samples, with the training data serving as the reference distribution. +RG represents RG-GAN

Dataset	Models	#RI (in K)↓	FLOPs ×10^{15}↓	FID↓
Obama	FastGAN	360	7.14	40.96
	RG-FastGAN	**328**	**6.34**	**35.81**
G-Cat	FastGAN	400	7.93	25.71
	RG-FastGAN	**360**	**6.98**	**24.31**
Panda	FastGAN	400	7.93	10.58
	RG-FastGAN	**328**	**6.34**	**9.87**
AF-Cat	FastGAN	400	7.93	35.54
	RG-FastGAN	**264**	**5.07**	**33.01**
AF-Dog	FastGAN	720	14.27	53.28
	RG-FastGAN	**680**	**13.32**	**52.04**

Table 4. FID comparison with state-of-the-art on few-shot datasets at 256×256 resolution.



(a) Generator (G)

(b) Discriminator (D)

(c) Weight Regeneration@RR

Figure 3: Patterns of pruning in G and D at (a) and (b)), and weight regeneration@RR in G and D at (c). The results also show that dynamic regeneration with pruning in RG-GAN enables balanced and simultaneous pruning and regeneration at G and D, mitigating the risk of losing key features and maintaining competitive dynamics essential for effective GAN.

Models	#RI (M)↓	FLOPs (×10^{17})↓	FID↓
SNGAN@0%	40.0	1.35	17.74
ST_SNGAN@20%	92.5	2.60	18.96
ST_SNGAN@46%	150.0	3.46	18.22
Re-SNGAN	31.5	1.03	17.87
Sparse_RG-SNGAN	31.5	1.06	**15.64**
Dense_RG-SNGAN	36.0	1.21	17.24

Table 5. Comparison to GAN tickets for full CIFAR-10.

We also compare with Re-GAN (Saxena et al. 2023) to see the benefit of regeneration in the GAN training. Results show that Sparse_RG-SNGAN (model output is sparse) and Dense_RG-SNGAN (model output is dense) outperform the Re-GAN. Notably, RG-SNGAN achieved the lowest FID, underscoring the effectiveness of incorporating dynamic regeneration in the GAN training process.

Method Analysis

In-Depth Analysis of Pruning Dynamics. In our analysis, we delve into the patterns of pruning within the G and D during the training process of RG-GAN as shown in Figure 3 (a) and (b). By plotting the level of pruning achieved during the sparse phase, we aim to better understand how our model adapts and evolves throughout training. The observed pruning patterns in RG-GAN directly tie back to its design principles. As we noted, the early stages of training display an increased degree of pruning, a critical step towards the shedding of redundant connections and maximizing model efficiency. As the training advances, this aggressive pruning diminishes, indicating that our method appreciates the growing significance of weights to the model's performance.

The differential pruning patterns across the layers of G and D also underline the model's adaptability and strategic approach. For instance, the higher pruning in G's early layers and D's later layers reflects a reduction of redundancy or less complex feature representations. Pruning across layers emphasizes the model's ability to dynamically assess and optimize the distribution of complexity across layers. Figure 3 (c) shows that the regeneration activity is more volatile and significant in the G, particularly at the beginning and in certain subsequent spikes. This could imply that the G's structure is being refined more aggressively or that it needs more adjustment as it learns to generate data.

As the training progresses to its end stages, we note that all layers of G converge to a similar level of sparsity. This demonstrates our design principle of achieving a balanced trade-off between model complexity and efficiency. While

D's ongoing increase in sparsity signifies model's commitment to continual refinement and efficiency enhancement.

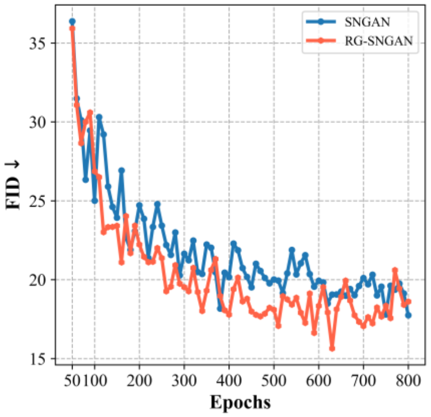


Figure 4: FID vs Epochs

These observations not only attest to the success of RG-GAN model's design but also offer significant insights for the future development of effective yet efficient GAN architectures. Figure 4 demonstrates that RG-SNGAN, utilizing dynamic pruning and weight regeneration, consistently surpasses SNGAN in terms of FID across all epochs, and reaches optimal performance faster, thereby demonstrating its improved performance and training efficiency.

Impact of Regeneration. In order to better understand the relative contributions of each component of RG-GAN, we conduct a study on the CIFAR-10 dataset (see Figure 5). We investigate the effects of adding dynamic pruning (DP), augmented regeneration (AR), and recurrent regeneration (RR) to the base SNGAN model, and further examine the impact of these additions when applied to either the G, D, or both networks (B) simultaneously. The analysis reveals how each component enhances RG-GAN's overall performance.

The addition of RR to DP in SNGAN greatly enhances its performance, emphasizing the key role of weight regeneration in maintaining learned knowledge and boosting model efficacy. By incorporating AR in both G and D, we witness a significant boost in performance over using DP and RR alone. We set sparsity penalty (λ) 5e-12 for all experiments.

Related Works

Stabilizing GAN Training. Recent innovations have proposed various loss functions (Arjovsky, Chintala, and Bottou 2017; Deshpande, Zhang, and Schwing 2018; Berthelot, Schumm, and Metz 2017), regularizations (Miyato et al. 2018; Zhang et al. 2020), and architectural modifications (Radford, Metz, and Chintala 2016; Song et al. 2021; Karras et al. 2020) to enhance GAN (Goodfellow, Pouget-Abadie, et al. 2014). Among these, state-of-the-art models like StyleGAN (Karras, Laine, and Aila 2019; Karras et al. 2020) emphasize deeper and wider networks, leading to improved generalization but longer training durations. Deep models, having more parameters are more challenging to train due to weaker gradient flow (Karras et al. 2018; B. Liu et al. 2021; Karnewar and Wang 2020). Techniques like Progressive GAN (ProGAN) (Karras et al. 2018) and MSG-GAN (Karnewar and Wang 2020) address these challenges. However, with limited data, these models face degraded performance and higher resource demands. While dynamic data augmentation (Zhao et al. 2020; Jiang et al. 2021) has been introduced to stabilize training, our approach uniquely focuses on a dynamic network architecture to enhance both stability and efficiency. Additionally, while sparsity in training has shown promise, there is a risk of losing of key features.

Lottery Ticket Hypothesis (LTH). Recent findings identified lottery tickets or winning tickets (Frankle and Carbin 2019) in GAN (X. Chen et al. 2021; Kalibhat, Balaji, and Feizi 2021) that can be independently trained to match or even outperform dense networks. Some efforts scaled LTH (T. Chen et al. 2021), but these do not address the premature pruning of essential connections. Our method, in contrast, can restore these prematurely pruned connections, preserving model capacity. On the other hand, locating and training an LTH consumes significantly more FLOPs than training a dense one (Saxena et al. 2023). With the increasing complexity of state-of-the-art models like StyleGAN2, such resource demands could lead to financial and environmental challenges (Schwartz et al. 2020).

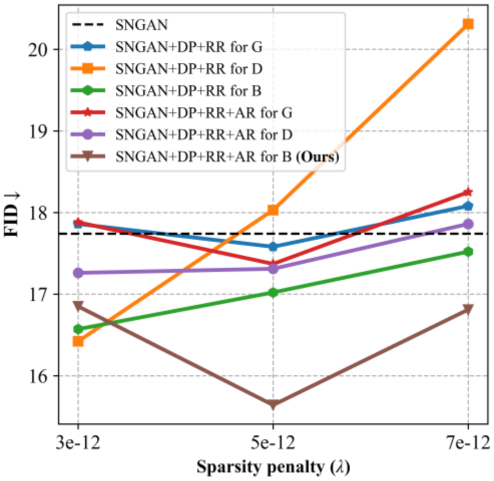


Figure 5: Impact of Regeneration

While these methods offer distinct and complementary approaches to ours, our achievements in high-quality image generation at reduced computational costs pave the way for us to train larger or more complex models for further enhancements in image quality.

Conclusion

This paper presents RG-GAN, a novel methodology introducing dynamic weight regeneration with pruning, as a new perspective to improve GAN training, particularly under limited data condition. Our extensive experiments confirm RG-GAN's robustness, improved sample quality, and efficiency, outperforming existing approaches. This success drives our motivation to train even larger or more complex models for further quality advancements. While this work offers a practical approach using weight regeneration, future work could focus on the development of a theoretical framework around the concept of regeneration in GAN. This could offer new insights and guide further improvements in the methodology. RG-GAN could be extended to multimodal GAN to handle more complex real-world scenarios in a more efficient and stable manner.

Acknowledgements

This work was supported by PolyU Start-up Fund (P0038876), Hong Kong Jockey Club Charities Trust (2021-0369), HK RGC Research Impact Fund (R5034-18), and Research Institute for Artificial Intelligence of Things, The Hong Kong Polytechnic University, Hong Kong.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, 214–23.

Berthelot, D.; Schumm, T.; and Metz, L. 2017. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *ArXiv Preprint ArXiv:1703.10717*. https://doi.org/10.1109/ACCESS.2018.2804278.

Chen, T.; Chen, Y.; Gan, Z.; Liu, J.; and Wang, Z. 2021. Data-Efficient GAN Training Beyond (Just) Augmentations: A Lottery Ticket Perspective. In *Advances in Neural Information Processing Systems*, 25:20941–55.

Chen, X.; Zhang, Z.; Sui, Y.; and Chen, T. 2021. GANs Can Play Lottery Tickets Too. *ArXiv Preprint ArXiv:2106.00134*. http://arxiv.org/abs/2106.00134.

Cheng, Y.-C.; Lin, C. H.; Lee, H.-Y.; Ren, J.; Tulyakov, S.; and Yang, M.-H. 2022. InOut: Diverse Image Outpainting via GAN Inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11431–40.

Deshpande, I.; Zhang, Z.; and Schwing, A. G. 2018. Generative Modeling Using the Sliced Wasserstein Distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3483–91.

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

García-Martín, E.; Rodrigues, C. F.; Riley, G.; and Grahn, H. 2019. Estimation of Energy Consumption in Machine Learning. *Journal of Parallel and Distributed Computing*, 134: 75–88. https://doi.org/10.1016/j.jpdc.2019.07.007.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, 3:2672–80. https://doi.org/10.3156/jsoft.29.5_177_2.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, 5768–78. http://arxiv.org/abs/1704.00028.

Han, S.; Mao, H.; and Dally, W. J. 2015. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *ArXiv Preprint ArXiv:1510.00149*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, 6626—6637.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-Consistent Adversarial Domain Adaptation. In *International Conference on Machine Learning*, 1989–98.

Hsu, H.-K.; Yao, C.-H.; Tsai, Y.-H.; Hung, W.-C.; Tseng, H.-Y.; Singh, M.; and Yang, M.-H. 2020. Progressive Domain Adaptation for Object Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 749–57.

Jiang, L.; Dai, B.; Wu, W.; and Loy, C. C. 2021. Deceive D: Adaptive Pseudo Augmentation for GAN Training with Limited Data. In *Advances in Neural Information Processing Systems*, 34:21655–67.

Kalibhat, N. M.; Balaji, Y.; and Feizi, S. 2021. Winning Lottery Tickets in Deep Generative Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:8038–46.

Karnewar, A.; and Wang, O. 2020. MSG-GAN: Multi-Scale Gradients for Generative Adversarial Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 7796–7805. https://doi.org/10.1109/CVPR42600.2020.00782.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *6th International Conference on Learning Representations*.

Karras, T.; Laine, S.; and Aila, T. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings* *of the IEEE Conference on Computer Vision and Pattern Recognition*, 4401–10. http://arxiv.org/abs/1812.04948.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and Improving the Image Quality of Stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8110–19.

Lee, H.-Y.; Tseng, H.-Y.; Huang, J.-B.; Singh, M.; and Yang, M.-H. 2018. Diverse Image-to-Image Translation via Disentangled Representations. In *Proceedings of the European conference on computer vision*, 35–51.

Liu, B.; Zhu, Y.; Song, K.; and Elgammal, A. 2021. Towards Faster and Stabilized Gan Training for High-Fidelity Few-Shot Image Synthesis. In *International Conference on Learning Representations.*

Liu, J.; Xu, Z.; Shi, R.; Cheung, R. C. C.; and So, H. K. H. 2020. Dynamic Sparse Training: Find Efficient Sparse Network From Scratch With Trainable Masked Layers. *ArXiv Preprint ArXiv:2005.06870*. http://arxiv.org/abs/2005.06870.

Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; and Zhang, C. 2017. Learning Efficient Convolutional Networks through Network Slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:2755–63. https://doi.org/10.1109/ICCV.2017.298.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral Normalization for Generative Adversarial Networks. In *6th International Conference on Learning Representations*.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations*, ICLR. http://arxiv.org/abs/1511.06434.

Saxena, D.; and Cao, J. 2021. Generative Adversarial Networks (GANs) Challenges, Solutions, and Future Directions. *ACM Computing Surveys (CSUR)*, 54 (3): 1–42.

Saxena, D.; Cao, J.; Xu, J.; and Kulshrestha, T. 2023. Re-GAN: Data-Efficient GANs Training via Architectural Reconfiguration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16230–40.

Schwartz, R.; Dodge, J.; Smith, N. A.; and Etzioni, O. 2020. Green AI. *Communications of the ACM*, 63 (12): 54–63.

Si, Z.; and Zhu, S.-C. 2011. Learning Hybrid Image Templates (HIT) by Information Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (7): 1354–67.

Song, X.; Chen, Y.; Feng, Z. H.; Hu, G.; Yu, D. J.; and Wu, X. J. 2021. SP-GAN: Self-Growing and Pruning Generative Adversarial Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32 (6): 2458–69. https://doi.org/10.1109/TNNLS.2020.3005574.

Strubell, E.; Ganesh, A.; and McCallum, A. 2019. Energy and Policy Considerations for Deep Learning in NLP. *ArXiv Preprint ArXiv:1906.02243.*

Xu, Z.; and Cheung, R. C. C. 2020. Accurate and Compact Convolutional Neural Networks with Trained Binarization. *30th British Machine Vision Conference*, BMVC.

Zhang, H.; Zhang, Z.; Odena, A.; and Lee, H. 2020. Consistency Regularization for Generative Adversarial Networks. *8th International Conference on Learning Representations*, ICLR. http://arxiv.org/abs/1910.12027.

Zhao, S.; Liu, Z.; Lin, J.; Zhu, J. Y.; and Han, S. 2020. Differentiable Augmentation for Data-Efficient GAN Training. In *Advances in Neural Information Processing Systems*, 2020-Decem:7559–70.

Zhu, J. Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017. Toward Multimodal Image-to-Image Translation. In *Advances in Neural Information Processing Systems*, 2017-Decem (1): 466–77.