Singular Value Scaling: Efficient Generative Model Compression via Pruned Weights Refinement

Hyeonjin Kim and Jaejun Yoo*

Graduate School of Artificial Intelligence
Ulsan National Institute of Science and Technology (UNIST)
{hyeonjin.kim, jaejun.yoo}@unist.ac.kr

Abstract

While pruning methods effectively maintain model performance without extra training costs, they often focus solely on preserving crucial connections, overlooking the impact of pruned weights on subsequent fine-tuning or distillation, leading to inefficiencies. Moreover, most compression techniques for generative models have been developed primarily for GANs, tailored to specific architectures like StyleGAN, and research into compressing Diffusion models has just begun. Even more, these methods are often applicable only to GANs or Diffusion models, highlighting the need for approaches that work across both model types. In this paper, we introduce Singular Value Scaling (SVS), a versatile technique for refining pruned weights, applicable to both model types. Our analysis reveals that pruned weights often exhibit dominant singular vectors, hindering fine-tuning efficiency and leading to suboptimal performance compared to random initialization. Our method enhances weight initialization by minimizing the disparities between singular values of pruned weights, thereby improving the fine-tuning process. This approach not only guides the compressed model toward superior solutions but also significantly speeds up fine-tuning. Extensive experiments on StyleGAN2, StyleGAN3 and DDPM demonstrate that SVS improves compression performance across model types without additional training costs.

Introduction

Generative models like Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) and Diffusion models (Ho, Jain, and Abbeel 2020) have achieved remarkable performance across various computer vision tasks such as image generation (Brock, Donahue, and Simonyan 2019; Karras et al. 2020; Ho, Jain, and Abbeel 2020; Wang et al. 2022), image editing (Baykal et al. 2023; Pehlivan, Dalva, and Dundar 2023; Kawar et al. 2023; Zhang et al. 2023), even video generation (Skorokhodov, Tulyakov, and Elhoseiny 2022; Ho et al. 2022; Blattmann et al. 2023) and 3D generation (Chan et al. 2022; Karnewar et al. 2023). The impressive performance of these generative models, however, often comes at the cost of high memory and computational demands, limiting their real-world applicability. To address these issues, several model compression technique for generative models have been proposed (Liu et al. 2021; Xu et al. 2022; Chung et al. 2024; Fang, Ma, and Wang 2023).

Model compression typically involves two steps: 1) pruning to reduce model size while retaining essential pre-trained knowledge, and 2) fine-tuning to restore performance. Among these, effective pruning has received significant attention for its ability to enhance following fine-tuning step. The retained pre-trained knowledge from pruning can enhance fine-tuning, leading to improved performance and faster convergence without additional training costs. While previous pruning methods (Chung et al. 2024; Fang, Ma, and Wang 2023) effectively maintain model performance, they often focus solely on preserving crucial connections in the pre-trained model, overlooking the impact of pruned weights on subsequent process, leading to inefficient fine-tuning or distillation. This issue becomes more severe as the model's capacity decreases, and in some cases, the pruned weights results in worse performance compared to random initialization; i.e. slow convergence speed and lower performance. Therefore, addressing these factors is essential for achieving more efficient generative models.

Our key observation is that pruned weights often exhibit dominant singular vectors, which results in a large disparity between the largest and smallest singular values. The presence of such dominant singular vectors significantly impacts the model's forward propagation, overshadowing the contributions of minor singular vectors, and the overall fine-tuning process is dominated by these few dominant singular vectors. We find that this could limit the exploration of diverse weight space. Therefore, ensuring a balanced contribution among the singular vectors within pruned weights at initialization may offer a potential solution to the inefficiencies observed in the fine-tuning process.

In this paper, we introduce a simple yet effective refinement technique called Singular Value Scaling (SVS) to enhance the efficiency of fine-tuning pruned weights. The dominant singular values is scaled down to reduce the disparity between the largest singular values and smallest singular values while preserving the relative order of the singular values at initialization. In this way, the refined pruned weights makes the fine-tuning process easier compared to directly using the pruned weights by ensuring all singular vectors contribute more evenly at the beginning of fine-tuning. Note that since our method focuses on the knowledge

*Corresponding author.

within each weight and is independent of specific architectures, it can be applied to both GANs and diffusion models.

We conduct extensive experiments with representative generative model architectures across various datasets, including StyleGAN2, StyleGAN3, and Denoising Diffusion Probabilistic Model (DDPM) on CIFAR10, CelebA-HQ, FFHQ, and LSUN-Church. The results demonstrate that our method enhances the fine-tuning process, leading to both faster convergence and improved solutions for the compressed model without additional training cost. Our contributions can be summarized as follows:

- We propose a simple yet effective method that enhances the efficacy of fine-tuning process of pruned generative models without additional training cost.
- By simply scaling down the singular values of pruned weights at initialization, our method, Singular Value Scaling (SVS) enables the compressed model to converge faster and achieve superior performance than using the existing baseline methods.
- We are the first to provide a general method for generative model compression that can be applied to both model types of GANs and Diffusion models.

Related Works

StyleGAN Compression

Recently, several studies (Liu et al. 2021; Xu et al. 2022; Chung et al. 2024) have been proposed for compressing unconditional GANs, particularly the StyleGAN family (Karras et al. 2020, 2021), which are the state-of-the-art models in the area. CAGAN (Liu et al. 2021) introduced the first framework for pruning pre-trained StyleGAN models and fine-tuning them via pixel-level and feature-level knowledge distillation. Following this, StyleKD (Xu et al. 2022) further enhanced the fine-tuning process by incorporating a specialized relation loss tailored for StyleGAN. In particular, CAGAN introduced a content-aware pruning technique that preserves connections in the pre-trained model crucial for the semantic part of generated images, while StyleKD inherited only the pre-trained model's mapping network and randomly initialized the synthesis network with a smaller size. More recently, DCP-GAN (Chung et al. 2024) proposed a diversity-aware channel pruning technique, which preserves connections in the synthesis network that contribute to sample diversity, significantly enhancing both the diversity of generated images and the training speed.

Diffusion Model Compression

Diffusion models have garnered significant attention for their stable training and impressive generative capabilities. However, their performance incurs high computational costs due to iterative sampling. To enhance efficiency, various sampling techniques have been developed to reduce the number of required iterations (Song, Meng, and Ermon 2021; Lu et al. 2022; Zheng et al. 2023). Orthogonal to the these sampling techniques, Diff-Prune (Fang, Ma, and Wang 2023) introduced a seminal work for reducing computational costs by compressing Diffusion models. This demonstrated that by primarily pruning connections involved in less important diffusion steps, a reduced-size Diffusion model with minimal performance loss can be achieved in significantly fewer training iterations compared to the original-sized model.

Weight Initialization and Trainability

Weight initialization is essential for efficient training in deep learning. Glorot (Glorot and Bengio 2010) and He initialization (He et al. 2015) are widely used to maintain activation variance and ensure stable optimization, particularly in feed-forward networks with ReLU activations. Orthogonal initialization (Saxe, McClelland, and Ganguli 2014), which ensures that all singular values of weights are 1, achieves dynamical isometry, the ideal state for trainability. In classifier pruning, TPP (Wang and Fu 2023) recently highlighted the importance of preserving trainability in pruned networks to support effective performance recovery.

Analysis

Suboptimal Results with Pruned Weights

Previous works in the field of generative model compression (Chung et al. 2024; Fang, Ma, and Wang 2023) have shown that well-designed pruning techniques can lead compressed models to better solutions compared to their randomly initialized counterparts, while reaching the same performance more quickly ($\times 2.5$ for StyleGAN2, $\times 8.0$ for DDPM). This is because pruned weights provide the compressed model with an initialization state that preserves the pre-trained model's performance, starting with better performance compared to the random counterpart. This initial gain continues throughout the fine-tuning process, ultimately resulting in a better-performing compressed model. However, we observe that as fine-tuning progresses, models initialized with pruned weights converge more slowly compared to those initialized randomly (see Figure 2: orange line (DCP-GAN) vs. blue line (StyleKD)). This issue worsens as the model size decreases, resulting in suboptimal solutions, which indicates that the pruned weights themselves contain factors that hinder fine-tuning efficiency.

Analyzing Pruned Weights with SVD

We analyze the learned prior of pruned weights inherited from the pre-trained model by employing Singular Value Decomposition (SVD), which is widely adopted for low-rank approximation by identifying important basis from the weights (Denton et al. 2014; Zhang et al. 2015; Girshick 2015; Yoo et al. 2019). For a weight matrix $W \in \mathcal{R}^{m \times n}$ $(m < n)$, we can decompose it using SVD:

$$W = U \Sigma V^T = \sum_{i=1}^{m} \sigma_i \vec{u}_i \vec{v}_i^T$$

where $U = [\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_m] \in \mathcal{R}^{m \times m}$ and $V = [\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n] \in \mathcal{R}^{n \times n}$ are orthogonal matrices and $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_m) \in \mathcal{R}^{m \times n}$ is a diagonal matrix with singular values on the diagonal. Here, for a fully connected layer, $W \in \mathcal{R}^{c_{out} \times c_{in}}$. For a convolutional layer, $W \in$

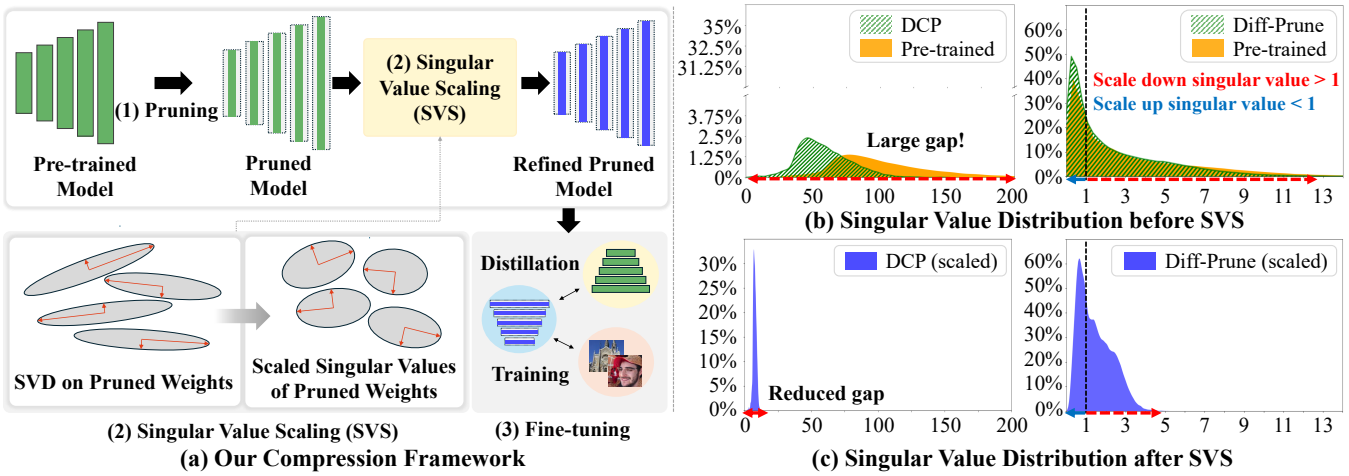

Figure 1: (a) A schematic overview of our compression framework. Unlike existing model compression scheme, we additionally perform pruned weights refining step for efficient model fine-tuning. First, we prune the pre-trained model. Next, we compute singular values of the weights of the pruned model. Then, we refine the pruned weights by scaling down the singular values with large magnitudes and scaling up the singular values with small magnitudes (*Singular Value Scaling, SVS*). Finally, the refined pruned model are fine-tuned. (b), (c) The singular value distribution before and after singular value scaling in the pruned weights by the method of DCP-GAN and Diff-Prune. The x-axis represents singular values and the y-axis represents the density. When applied SVS, the singular value disparity reduces, balancing the power of each singular vector in the pruned weights.

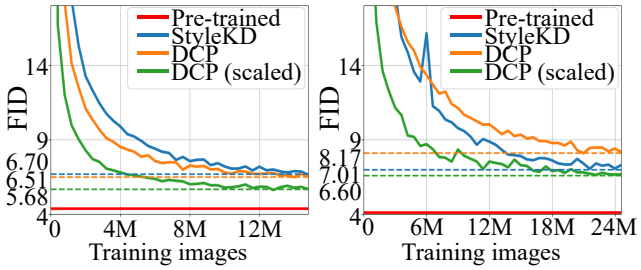

Figure 2: FID convergence graph in different StyleGAN2 Architectures compressed by different methods. The x-axis represents the number of images shown to the discriminator. Solid line represents FID with respect to the training images, while dashed line represents the best FID of the compressed model by the corresponding method. "scaled" means that our *Singular Value Scaling* is applied to the pruned weights.

$\mathcal{R}^{c_{out}\times(c_{in}\times k\times k)}$, where k is the kernel size. For a convolutional layer, we consider the weight is flattened. Each singular value represents the influence of its corresponding singular vector within the weight. The most notable observation is the large gap between the largest and smallest singular values of pruned weights ($\sim \times 100$). This implies that the forward and backward propagations of the weights are heavily influenced by these dominant singular vectors. This can potentially bias the compressed model towards these singular vectors during training, severely limiting diverse exploration in the weight space (Saxe, McClelland, and Ganguli 2014; Wang and Fu 2023).

Method

Scaling Singular Values of Pruned Weights

Based on our observation, we propose "Singular Value Scaling (SVS)", to refine pruned weights to enhance fine-tuning efficiency. Our primary goal is to reduce the gap between the singular values of pruned weights. In the pruned weights, dominant singular vectors tend to have significantly larger singular values compared to smaller ones, and this gap increases as the values grow. Since all bases in pruned weights contain important knowledge from the pre-trained model, we prevent any single basis from dominating to let these bases contribute equally at the beginning of training. To achieve this, we simply scale the singular values using the "square root function",

$$W_{scaled} = U\Sigma_{scaled}V^T = \sum_{i=1}^{m}\sqrt{\sigma_i}\vec{u}_i\vec{v}_i^T$$

Here, U and V remain unchanged, and $\Sigma_{scaled} = diag(\sqrt{\sigma_1}, \sqrt{\sigma_2}, \cdots, \sqrt{\sigma_m}) \in \mathcal{R}^{m\times n}$. Square root function has several good properties: 1) Since singular values are non-negative, it maps non-negative values to non-negative values, 2) For $\sigma_i < 1$, it increases the value, while for $\sigma_i > 1$, it decreases the value more as the value grows. 3) In the positive domain, it is monotonically increasing, thereby maintaining the relative order of singular values. By scaling the singular values of pruned weights in this manner, we can preserve the original bases while balancing their relative contributions. This balanced contribution helps the compressed model fully leverage the pre-trained model's knowledge, facilitating a more effective path to the optimal solution. Alternative functions, such as $\log(x+1)$ or $|\log(x)|$,

could also achieve similar effects. We provide ablation study on these functions in the Experiments section.

Scaling Bias with respect to the Scaled Weights

Since biases and weights are learned simultaneously, we need to consider the impact of scaling the singular values of the weights on the corresponding biases. Let us consider the following linear equation, $y = Wx + b$, where $W \in \mathcal{R}^{m \times n}$ is the weights, $x \in \mathcal{R}^n$ is the input and $b \in \mathcal{R}^m$ is the bias. We can factor the equation with respect to W as follows:

$$y = W(x + W^{\dagger}b) = W(x + V\Sigma^{-1}U^T b)$$

where $W^{\dagger} \in \mathcal{R}^{n \times m}$ is the Moore-Penrose inverse (Petersen 2012) and $\Sigma^{-1} = diag(\sigma_1^+, \sigma_2^+, \cdots, \sigma_m^+) \in \mathcal{R}^{n \times m}$ with

$$\sigma_i^+ = \begin{cases} \frac{1}{\sigma_i}, & \text{if } \sigma_i \neq 0 \\ 0, & \text{if } \sigma_i = 0 \end{cases}$$

By representing the bias b as $b = |b|\vec{\beta}$, where $|\cdot|$ is the vector norm and $\vec{\beta}$ is the normalized vector of b, it becomes:

$$y = W(x + V\Sigma^{-1}U^T b) = W(x + V(|b|\Sigma^{-1})U^T\vec{\beta})$$

where $|b|\Sigma^{-1} = diag(|b|\sigma_1^+, |b|\sigma_2^+, \cdots, |b|\sigma_m^+)$. Thus, for $\sigma_i > 0$, when scaling the biases, instead of only scaling σ_i, we must also scale b: $b_{scaled} = b/\sqrt{|b|}$, which leads to:

$$|b_{scaled}|/\sigma_{i,scaled} = \sqrt{|b|/\sigma_i}.$$

Experiments

Experimental Setup

Baselines. To evaluate our method, we conducted experiments on StyleGAN2, which is the most extensively studied model in the field of generative model compression. Additionally, to validate our method on generative models with different architectures, we conducted experiments on StyleGAN3 and the Denoising Diffusion Probabilistic Model (DDPM), which, along with StyleGAN2, are representative models in the field of generative modeling.

Implementation details. For the StyleGAN2 compression, we re-implement previous StyleGAN2 compression methods, which were implemented on the unofficial StyleGAN2 implementation, on the official StyleGAN2 implementation because the official implementation provides more optimized StyleGAN2 training settings and are easy to use. we apply our method to the weights pruned by DCP-GAN (Chung et al. 2024), which is the state-of-the-art pruning method in StyleGAN2 compression. We mainly compare our method against StyleKD (Xu et al. 2022) and DCP-GAN because they are the methods that demonstrate the best performance in StyleGAN2 compression. We use two StyleGAN2 architectures: the optimized architecture provided by the official implementation (denoted as "StyleGAN2 (small)") and the original structure used in previous works (denoted as "StyleGAN2 (base)"). Experiments are conducted on the FFHQ (Karras, Laine, and Aila 2019) and LSUN Church (Yu et al. 2015) datasets. Following DCP-GAN, we use a facial mask as the content mask for the FFHQ dataset and a uniform mask (all pixel values are assigned values 1) for LSUN Church dataset. For StyleGAN3 compression, we implement our method based on the DCP-GAN implementation. Similar to StyleGAN2 compression, we apply our method to the pruned weights using the DCP-GAN method for our experiments. We follow the experimental setup of DCP-GAN and train StyleGAN3 on the FFHQ dataset until the discriminator see 10 million images for both DCP-GAN and our method. For DDPM compression, we build upon Diff-Prune (Fang, Ma, and Wang 2023), a foundational work in Diffusion model pruning. We train pruned DDPMs using the CIFAR10 (Alex 2009) and CelebA-HQ (Liu et al. 2015) datasets, following the experimental setup of Diff-Prune. Unlike Diff-Prune, which tested only on a 30% channel sparsity, we explore more extreme levels of sparsity in our experiments, including 50% and 70% channel sparsity. We provide more detailed implementation in the supplementary material.

Evaluation metrics. For StyleGAN compression, we evaluate our method using the Fréchet Inception Distance (FID) (Heusel et al. 2017) and Precision and Recall (P&R) (Kynkäänniemi et al. 2019). We also measure Density and Coverage (D&C) (Naeem et al. 2020), which are more robust to outliers than Precision and Recall. Here, Precision and Density evaluate the fidelity of generated samples, while Recall and Coverage assess their diversity. To calculate FID, we use all real samples from each dataset and 50K fake samples. For P&R and D&C calculation, we use 50K fake samples and all real samples from FFHQ, and 50K real samples from LSUN Church due to computational costs. For DDPM compression, in addition to FID, we employ Structural Similarity (SSIM) (Wang et al. 2004) to evaluate consistency with the pre-trained model, following the Diff-Prune. The SSIM score is measured between images generated by the pre-trained model and the compressed model, given the identical noise inputs. We use a 100-step DDIM sampler (Song, Meng, and Ermon 2021) for sampling.

Analysis on Convergence Speed

Figure 2 visualizes the FID convergence graph for two different StyleGAN2 architectures (StyleGAN2 (base) and StyleGAN2 (small)) in the FFHQ dataset. As shown in Figure 2 (a) (StyleGAN2 (base)), although DCP-GAN converges to better performance than StyleKD, the FID convergence speed of DCP-GAN noticeably slows down in the later stages of training. This issue becomes even more severe as the model's capacity decreases. In Figure 2 (b), this slowdown allows StyleKD to surpass DCP-GAN early in fine-tuning, leading DCP-GAN to a suboptimal point. These results indicate that fine-tuning a model naively initialized with the pruned weights can lead to significantly poor outcomes, depending on the model's capacity. In contrast, when pruned weights are refined using SVS, the model is much easier to fine-tune. It not only fits faster across most intervals but also converges to a significantly better solution compared to previous methods.

Dataset	Method	Arch	Params ↓	FLOPs ↓	FID ↓	P ↑	R ↑	D ↑	C ↑
FFHQ	Teacher	base	30.0M	45.1B	4.29	0.762	0.561	1.061	0.845
	StyleKD (Xu et al. 2022)		5.6M	4.1B	6.70	0.711	0.551	0.879	0.786
	DCP-GAN (Chung et al. 2024)				6.51	0.712	**0.556**	0.884	0.785
	DCP-GAN (scaled)				**5.68**	**0.740**	0.530	**0.981**	**0.806**
	Teacher	small	24.7M	14.9B	4.02	0.769	0.555	1.095	0.854
	GS (Wang et al. 2020)		4.9M	1.3B	10.23	0.702	0.430	0.845	0.721
	CAGAN (Liu et al. 2021)				9.23	0.685	0.500	0.760	0.722
	StyleKD (Xu et al. 2022)				7.01	0.707	**0.543**	**0.900**	**0.783**
	DCP-GAN (Chung et al. 2024)				8.17	0.680	0.539	0.787	0.741
	DCP-GAN (scaled)				**6.60**	**0.719**	0.532	0.874	0.778
Church	Teacher	base	30.0M	45.1B	3.97	0.698	0.553	0.849	0.823
	StyleKD (Xu et al. 2022)		5.6M	4.1B	5.72	0.703	0.460	0.845	0.777
	DCP-GAN (Chung et al. 2024)				4.80	0.692	**0.505**	0.810	0.790
	DCP-GAN (scaled)				**4.62**	**0.716**	0.501	**0.878**	**0.812**
	Teacher	small	24.7M	14.9B	4.37	0.696	0.512	0.815	0.809
	StyleKD (Xu et al. 2022)		1.38B	1.38B	5.99	0.698	0.446	0.848	0.777
	DCP-GAN (Chung et al. 2024)				6.30	0.687	0.435	0.793	0.762
	DCP-GAN (scaled)				**5.18**	**0.719**	**0.464**	**0.934**	**0.805**

Table 1: Quantitative Results on StyleGAN2 compression. We report training results of previous methods on FFHQ-256 and LSUN Church-256 datasets. "Params" and "FLOPs" refer to the number of parameters in generator and floating-point operations respectively. "P" and "R" denote precision and recall metrics, while "D" and "C" represent density and coverage metrics. "base" and "small" denote the StyleGAN2 architecture with different capacity. "scaled" means that our *Singular Value Scaling* is applied to the pruned weights. All reported metrics are trained results with the official StyleGAN2 implementation.

Method	FID ↓	P ↑	R ↑	D ↑	C ↑
Teacher	4.76	0.736	0.593	0.955	0.817
DCP-GAN	9.88	0.694	0.484	0.780	0.720
DCP-GAN (scaled)	**8.38**	**0.702**	**0.531**	**0.808**	**0.748**

Table 2: Quantitative Results on StyleGAN3 Compression.

Quantitative Results on StyleGAN2 Compression

Table 1 shows results on StyleGAN2 compression on different StyleGAN2 architectures in FFHQ and LSUN Church datasets. For FID, our method outperforms previous compression methods with a clear margin in all experiments. It generally outperforms previous methods in Precision and Density metrics, and matches or exceeds their performance in Recall and Coverage metrics. These results show that our method enables the compressed model to better match the distribution compared to previous methods, leading to higher quality and more diverse samples. One notable observation is that when the capacity of StyleGAN2 is sufficient (base), DCP-GAN consistently outperforms StyleKD, and as the capacity decreases (small), DCP-GAN's performance lags behind that of StyleKD, indicating that the pruned weights from DCP-GAN are harder to fine-tune. In contrast, our method provides a more effective and refined initialization, resulting in improved final performance.

Quantitative Results on StyleGAN3 Compression

Table 2 presents the results of StyleGAN3 compression. The model initialized with our refined pruned weights outperforms the DCP-GAN across all metrics. This result shows that, similar to StyleGAN2, refining pruned weights facilitates easier fine-tuning of the model, enabling it to generate more plausible and diverse images.

Quantitative Results on DDPM Compression

Table 3 presents the results of DDPM compression. For DDPM compression, we compare our method with the best-performing baseline during fine-tuning. Diff-Prune achieves higher SSIM scores, indicating closer alignment with the pre-trained model. This is expected, as our method refines pruned weights to balance knowledge preservation with easier fine-tuning. Consequently, our method consistently achieves better FID scores across all pruning ratios, indicating more effective distribution matching. Additionally, the modest drop in SSIM indicates that our method still effectively retains contextual knowledge, even as it further relaxes the knowledge of the pre-trained model.

Qualitative Results on StyleGAN2 Compression

Figure 3 shows samples generated by StyleGAN2 (small) compressed using different methods from the same noise vector. Despite all compressed models being trained with the same loss, the characteristics of the generated samples vary depending on the weight initialization and frequently exhibit artifacts. For example, the model compressed by StyleKD often misses contextual details, such as face shape (second and fourth row). DCP-GAN, which prunes the pre-trained model to maximize diversity, tends to focus heavily on details like wrinkles and hair, while often failing to restore the overall appearance of the face. Additionally, as seen in the

DDPM CIFAR-10 32×32 (100 DDIM steps)										
Method	Channel Sparsity ↑	Params ↓	MACs ↓	FID ↓	SSIM ↑	P ↑	R ↑	D ↑	C ↑	Train Steps ↓
pre-trained	0%	35.7M	6.1G	4.19	1.000	0.672	0.759	0.762	0.897	800K
Diff-Prune†	30%	19.8M	3.4G	5.29	**0.932**	N/A	N/A	N/A	N/A	100K
Diff-Prune				5.49	0.931	**0.671**	**0.742**	**0.772**	0.888	160K
Diff-Prune (scaled)				**5.14**	0.923	0.668	0.740	0.768	**0.889**	
Diff-Prune	50%	8.96M	1.5G	7.77	**0.914**	0.670	**0.720**	0.805	0.861	360K
Diff-Prune (scaled)				**7.41**	0.912	**0.671**	0.718	**0.817**	**0.868**	
Diff-Prune	70%	5.12M	1.0G	9.87	0.906	0.665	**0.702**	0.818	0.836	600K
Diff-Prune (scaled)				**9.38**	**0.907**	**0.671**	0.699	**0.834**	**0.844**	
DDPM CelebA-HQ 64 × 64 (100 DDIM steps)										
Method	Channel Sparsity ↑	Params ↓	MACs ↓	FID ↓	SSIM ↑	P ↑	R ↑	D ↑	C ↑	Train Steps ↓
pre-trained	0%	78.7M	23.9G	6.48	1.000	0.587	0.812	0.488	0.755	500K
Diff-Prune†	30%	43.7M	13.3G	6.24	0.885	N/A	N/A	N/A	N/A	100K
Diff-Prune				6.17	**0.949**	0.591	**0.803**	0.502	0.760	100K
Diff-Prune (scaled)				**5.57**	0.938	**0.619**	0.793	**0.573**	**0.785**	
Diff-Prune	50%	19.7M	6.0G	5.32	**0.926**	0.618	**0.773**	0.594	0.793	200K
Diff-Prune (scaled)				**4.66**	0.921	**0.634**	0.766	**0.626**	**0.812**	
Diff-Prune	70%	9.25M	4.2G	6.30	**0.913**	0.584	**0.772**	0.524	0.761	240K
Diff-Prune (scaled)				**5.04**	0.905	**0.625**	0.752	**0.631**	**0.812**	

Table 3: Quantitative results on DDPM compression. "†" represents reported results in the paper. "N/A" indicates the inability to evaluate due to the inaccessibility of the weights used in the paper. "Channel Sparsity" refers the ratio of removed channels from the pre-trained model and "Params" and "MACs" are resulting number of parameters and multiply-accumulate of the compressed model. "SSIM" estimates the similarity between generated sample of pre-trained and compressed models given same noise. "P" and "R" denote precision and recall metrics, while "D" and "C" represent density and coverage metrics. "scaled" means that our *Singular Value Scaling* is applied to the pruned weights.

Method	FID ↓	P ↑	R ↑	D ↑	C ↑
CAGAN (Liu et al. 2021)	9.23	0.685	0.500	0.700	0.722
CAGAN (scaled)	**6.83**	**0.720**	**0.511**	**0.894**	**0.768**

Table 4: Ablation study on the CAGAN with FFHQ. "scaled" means that SVS is applied to the pruned weights.

third column, first row of LSUN Church, DCP-GAN tends to generate excessively repetitive patterns by focusing too much on restoring fine details, often neglecting the overall context. In contrast, models with refined pruned weights preserve the details of the pre-trained model while faithfully restoring its overall context. This result shows that, to achieve both diversity and fidelity in the compressed model, it is essential not only to preserve knowledge from the pre-trained model but also to refine the pruned weights for a proper balance with fine-tuning.

Ablation Study on Other Pruning Methods

There are only a few pruning methods for generative models, with CAGAN (Liu et al. 2021), DCP-GAN, and Diff-Prune considered key baselines. We extend our analysis to CAGAN to further validate the applicability of SVS beyond state-of-the-art methods. The pruned weights of CAGAN exhibit a similar singular value distribution to DCP-GAN, with existence of dominant singular vectors. As shown in Table 4, applying SVS to CAGAN consistently leads to substantial improvements in all metrics, confirming that applicability of our method.

Dataset	Scaling functions	FID ↓	P ↑	R ↑	D ↑	C ↑
FFHQ	$\log(x+1)$	**6.18**	**0.732**	0.513	0.960	0.789
	$\lvert\log(x)\rvert$	6.32	0.731	0.519	**0.975**	**0.796**
	$\sqrt{x}$	6.60	0.719	**0.532**	0.874	0.778
Church	$\log(x+1)$	5.30	**0.723**	0.458	0.933	**0.809**
	$\lvert\log(x)\rvert$	5.43	0.720	0.453	0.916	0.799
	$\sqrt{x}$	**5.18**	0.719	**0.464**	**0.934**	0.805

Table 5: Ablation on the DCP-GAN (StyleGAN2 (small)) with different scaling functions.

Ablation Study on Different Scaling Functions

While we primarily use the square root function, we also test alternative scaling functions, such as $\log(x+1)$ and $|\log(x)|$, for reducing singular value gaps. Table 5 shows similar performance across all functions, highlighting the importance of addressing singular value gaps. The square root function was chosen for its simplicity, with results suggesting that broader exploration of scaling strategies could further enhance compression performance.

Ablation Study on Different Scaling Methods

In Table 6, we explore additional approaches to refine singular values: (**a**) normalizing singular values (setting $\sigma_i = 1$), (**b**) normalizing with the spectral norm (dividing all singular values σ_i by the largest singular value σ_1), (**c**) Squaring singular values, (**d**) applying spectral normalization to the synthesis network (Miyato et al. 2018), and (**e**) iterative refining the generator's weights with our method. For (**e**), we perform iterative refining only for the first half of the training due to performance degradation. From (**a**) and (**b**), we find

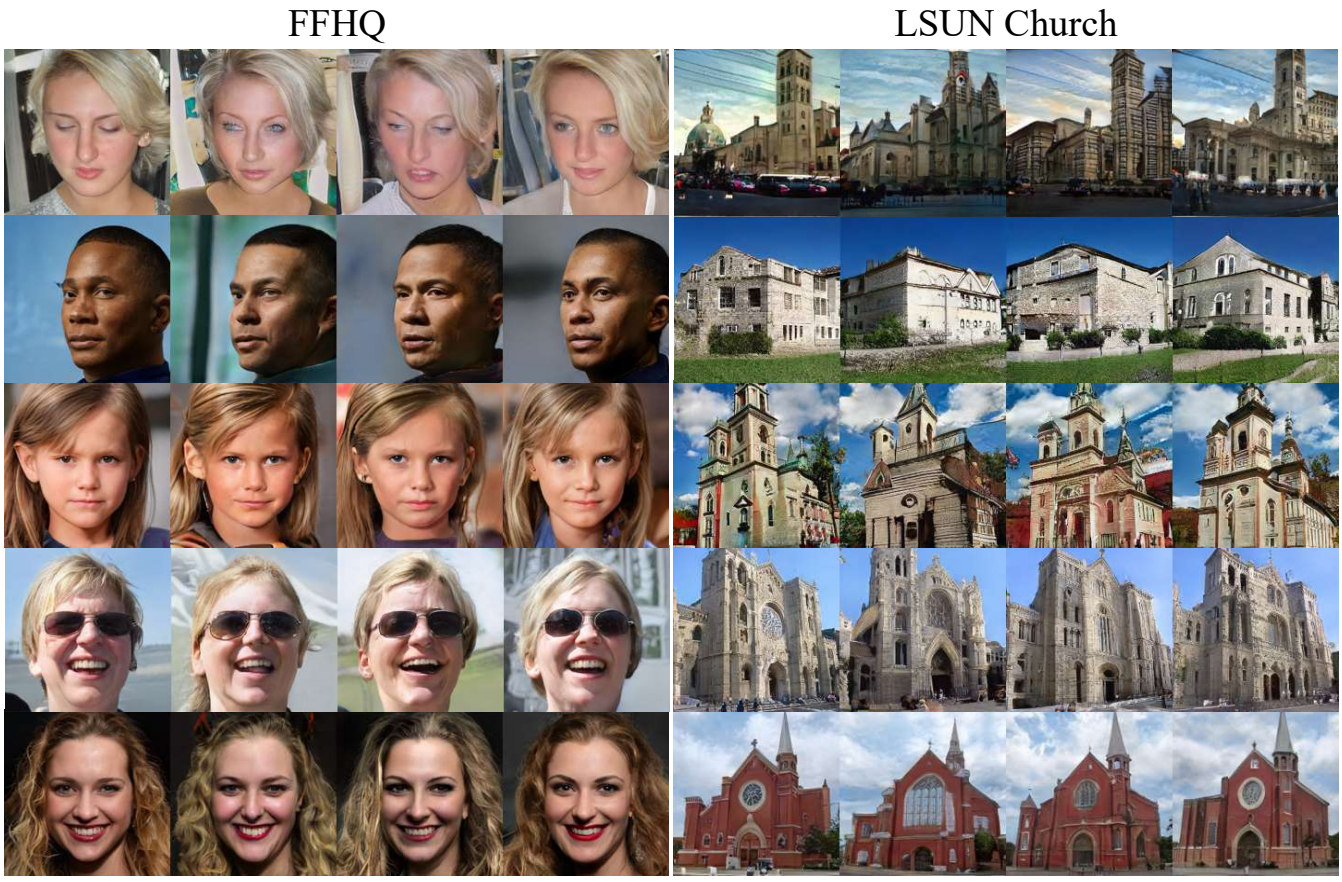

Figure 3: Qualitative results on FFHQ and LSUN Church datasets. Samples in each row are generated from same noise vector with StyleGAN2 (small), which is compressed using different compression methods with channel sparsity 70%. "Ours" denotes the compressed model with DCP-GAN refined by *Singular Value Scaling*.

Method (StyleGAN2 (small), FFHQ)	FID ↓
(**a**) Normalizing singular values	N/A
(**b**) Spectral normalizing singular values	N/A
(**c**) Squaring singular values	N/A
(**d**) Spectral normalization to the generator	8.00
(**e**) Iterative refining the generator's weights	8.40
Singular Value Scaling (SVS)	**6.60**

Table 6: Ablation on the different refinement methods. "N/A" indicates that the model diverges.

that setting singular values to 1 (where the weights are orthogonal) or spectral normalizing singular values cause the training to diverge, similar to previous observations (Saxe, McClelland, and Ganguli 2014; Wang and Fu 2023). From (**c**), we find that widening singular value gaps leads to early training divergence, highlighting the importance of minimizing these gaps for effective fine-tuning. From (**d**), we see that applying spectral normalization to the generator's weights makes it difficult for the generator to learn. Finally, from (**e**), we find that repeatedly refining the singular values of weights during training perturbs the learning process, depriving the generator of sufficient training time and leading to suboptimal results. From these results, we find that applying our method once at model initialization is not only efficient but also leads to more effective training.

Conclusion

In this paper, we propose "Singular Value Scaling" that refines pruned weights to achieve more effective and efficient generative model compression. Our method minimizes the disparities between singular values of pruned weights, allowing the pruned model to explore a broader weight space. Our method is not only simple and effective but also versatile, applicable across different model types, GANs and Diffusion models. Extensive experiments demonstrate that our method provides a more effective initialization state, leading to performance improvements across various metrics for both StyleGAN and Diffusion model compression.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST)), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2022R1C1C100849612), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00959, No.RS-2022-II220959 (Part 2) Few-Shot Learning of Causal Inference in Vision and Language for Decision Making), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-II220264, Comprehensive Video Understanding and Generation with Knowledge-based Deep Logic Neural Network), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2021-II212068, Artificial Intelligence Innovation Hub).

References

Alex, K. 2009. Learning multiple layers of features from tiny images. *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*.

Baykal, A. C.; Anees, A. B.; Ceylan, D.; Erdem, E.; Erdem, A.; and Yuret, D. 2023. CLIP-Guided StyleGAN Inversion for Text-Driven Real Image Editing. *ACM Trans. Graph.* Just Accepted.

Blattmann, A.; Rombach, R.; Ling, H.; Dockhorn, T.; Kim, S. W.; Fidler, S.; and Kreis, K. 2023. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations*.

Chan, E. R.; Lin, C. Z.; Chan, M. A.; Nagano, K.; Pan, B.; Mello, S. D.; Gallo, O.; Guibas, L.; Tremblay, J.; Khamis, S.; Karras, T.; and Wetzstein, G. 2022. Efficient Geometry-aware 3D Generative Adversarial Networks. In *CVPR*.

Chung, J.; Hyun, S.; Shim, S.-H.; and Heo, J.-P. 2024. Diversity-aware Channel Pruning for StyleGAN Compression. *arXiv preprint arXiv:2403.13548*.

Denton, E. L.; Zaremba, W.; Bruna, J.; LeCun, Y.; and Fergus, R. 2014. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27.

Fang, G.; Ma, X.; and Wang, X. 2023. Structural pruning for diffusion models. In *Advances in Neural Information Processing Systems*.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256. JMLR Workshop and Conference Proceedings.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

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. *Advances in neural information processing systems*, 30.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022. Video diffusion models. *Advances in Neural Information Processing Systems*, 35: 8633–8646.

Karnewar, A.; Vedaldi, A.; Novotny, D.; and Mitra, N. J. 2023. HOLODIFFUSION: Training a 3D Diffusion Model Using 2D Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18423–18433.

Karras, T.; Aittala, M.; Laine, S.; Härkönen, E.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2021. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34: 852–863.

Karras, T.; Laine, S.; and Aila, T. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4396–4405.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. CVPR*.

Kawar, B.; Zada, S.; Lang, O.; Tov, O.; Chang, H.; Dekel, T.; Mosseri, I.; and Irani, M. 2023. Imagic: Text-Based Real Image Editing with Diffusion Models. In *Conference on Computer Vision and Pattern Recognition 2023*.

Kynkäänniemi, T.; Karras, T.; Laine, S.; Lehtinen, J.; and Aila, T. 2019. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32.

Liu, Y.; Shu, Z.; Li, Y.; Lin, Z.; Perazzi, F.; and Kung, S.-Y. 2021. Content-Aware GAN Compression. In *CVPR*.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

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

Naeem, M. F.; Oh, S. J.; Uh, Y.; Choi, Y.; and Yoo, J. 2020. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, 7176–7185. PMLR.

Pehlivan, H.; Dalva, Y.; and Dundar, A. 2023. StyleRes: Transforming the Residuals for Real Image Editing With StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1828–1837.

Petersen, P. 2012. *Linear algebra*. Springer.

Saxe, A.; McClelland, J.; and Ganguli, S. 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of the International Conference on Learning Represenatations 2014*. International Conference on Learning Represenatations 2014.

Skorokhodov, I.; Tulyakov, S.; and Elhoseiny, M. 2022. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3626–3636.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Wang, H.; and Fu, Y. 2023. Trainability Preserving Neural Pruning. In *The Eleventh International Conference on Learning Representations*.

Wang, H.; Gui, S.; Yang, H.; Liu, J.; and Wang, Z. 2020. GAN Slimming: All-in-One GAN Compression by A Unified Optimization Framework. In *European Conference on Computer Vision*.

Wang, J.; Yang, C.; Xu, Y.; Shen, Y.; Li, H.; and Zhou, B. 2022. Improving GAN Equilibrium by Raising Spatial Awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11285–11293.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Xu, G.; Hou, Y.; Liu, Z.; and Loy, C. C. 2022. Mind the gap in distilling StyleGANs. In *European Conference on Computer Vision*, 423–439. Springer.

Yoo, J.; Uh, Y.; Chun, S.; Kang, B.; and Ha, J.-W. 2019. Photorealistic Style Transfer via Wavelet Transforms. In *International Conference on Computer Vision (ICCV)*.

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

Zhang, X.; Zou, J.; Ming, X.; He, K.; and Sun, J. 2015. Efficient and accurate approximations of nonlinear convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition*, 1984–1992.

Zhang, Z.; Han, L.; Ghosh, A.; Metaxas, D. N.; and Ren, J. 2023. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6027–6037.

Zheng, H.; Nie, W.; Vahdat, A.; Azizzadenesheli, K.; and Anandkumar, A. 2023. Fast sampling of diffusion models via operator learning. In *International conference on machine learning*, 42390–42402. PMLR.