Nickel and Diming Your GAN: A Dual-Method Approach to Enhancing GAN Efficiency via Knowledge Distillation

Sangyeop Yeo, Yoojin Jang, and Jaejun Yoo*

Laboratory of Advanced Imaging Technology (LAIT)
Ulsan National Institute of Science and Technology (UNIST)
sangyeop377@gmail.com, {softjin, jaejun.yoo}@unist.ac.kr
(*: corresponding author)

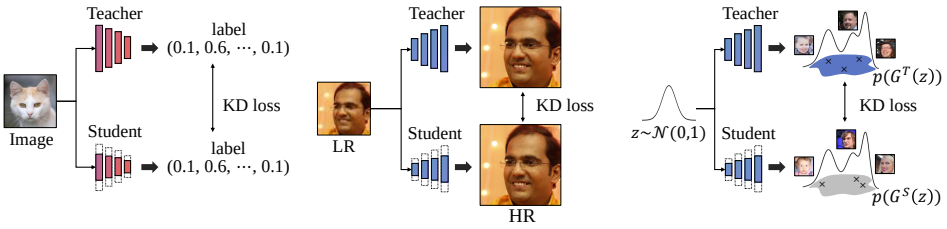
Abstract. In this paper, we address the challenge of compressing generative adversarial networks (GANs) for deployment in resource-constrained environments by proposing two novel methods: Distribution Matching for Efficient compression (`DiME`) and Network Interactive Compression via Knowledge Exchange and Learning (`NICKEL`). `DiME` employs foundation models as embedding kernels for efficient distribution matching, leveraging maximum mean discrepancy to facilitate effective knowledge distillation. `NICKEL` employs an interactive compression method that enhances the communication between the student generator and discriminator, achieving a balanced and stable compression process. Our comprehensive evaluation on the StyleGAN2 architecture with the FFHQ dataset shows the effectiveness of our approach, with `NICKEL & DiME` achieving FID scores of 10.45 and 15.93 at compression rates of 95.73% and 98.92%, respectively. Remarkably, our methods sustain generative quality even at an extreme compression rate of 99.69%, surpassing the previous state-of-the-art performance by a large margin. These findings not only show our methodologies' capacity to significantly lower GANs' computational demands but also pave the way for deploying high-quality GAN models in settings with limited resources. Our code is available at Nickel & Dime.

Keywords: Model compression · Generative models · Compact models

1 Introduction

Generative Adversarial Networks (GANs) have attracted significant popularity as one of the most promising generative models, alongside the diffusion models [6, 15, 49, 55], in various computer vision tasks such as super-resolution [32, 44, 58], image editing [12, 46, 53], and image generation [20, 24, 25]. Particularly, thanks to their fast inference speed compared to diffusion models, GANs offer significant advantages for real-time applications [18, 26, 48]. However, despite their outstanding performance, the application of state-of-the-art GANs [8,20,22, 24,25,51,52] on edge devices is constrained by their huge resource consumption.

Although compression methods have been extensively studied for classification tasks [11, 13, 31, 45, 56], their naïve application to generative models often



(a) KD for classifier **(b)** KD for conditional GAN **(c)** KD for unconditional GAN

Fig. 1: Comparison of knowledge distillation methods. (a) In classification tasks, the instance matching of output labels between the teacher and student is performed. Output labels are in low-dimensional space. Ideally, the outputs of the student and teacher are the same. (b) In conditional generative tasks, the instance matching of output images between the teacher and student is performed. Output images are in high dimensional space. The outputs of the student and teacher are similar (in terms of structure or background). (c) In unconditional generative tasks, the distribution matching of output images between the teacher and student is performed. There is no necessity for each input to have the same output.

leads to significant performance degradation [54,57]. As shown in Fig. 1, to distill the rich knowledge from the teacher to the student, simple label matching is performed in classification models, whereas high-dimensional output matching is required in generative models. Moreover, in GANs, achieving optimal performance requires a delicate balance between the generator and discriminator during adversarial training, which becomes more difficult between the pruned generator and discriminator (see Fig. 3a).

Recently, several GAN compression methods [16,17,21,35,36,39,57,59,60,62] have been proposed, but compressing unconditional GAN remains challenging. This is because conditional GANs require instance matching [59] as the teacher and student strive for similar outputs in a manner akin to classification tasks, whereas unconditional GAN compression demands distribution matching [21] (Fig. 1). There exist a few unconditional GAN compression studies [21,39,57,59], but they either still suffer from significant performance degradation [21,39,57,59] or require additional costs such as manual labeling [39] and MCMC sampling [21].

To address these problems, we first propose the Distribution Matching for Efficient compression (`DiME`). Most GAN compression methods utilized the embedding space (*e.g.*, perceptual [17, 39], frequency [62]) because directly matching high-dimensional output images leads to significant performance degradation. Similarly, we leverage the foundation models (*i.e.*, DINO [2, 43], CLIP [47]) as embedding kernels, which have shown successful applications with strong embedding power on various tasks [29, 38, 63]. Furthermore, Santos *et al.* and Yeo *et al.* [50, 60] have shown that neural networks can be considered as characteristic kernels to map into Reproducing Kernel Hilbert Space (RKHS), where matching the extracted features of two distributions is equivalent to matching the original distributions as the maximum mean discrepancy (MMD) critic [9, 10, 34, 37, 50].

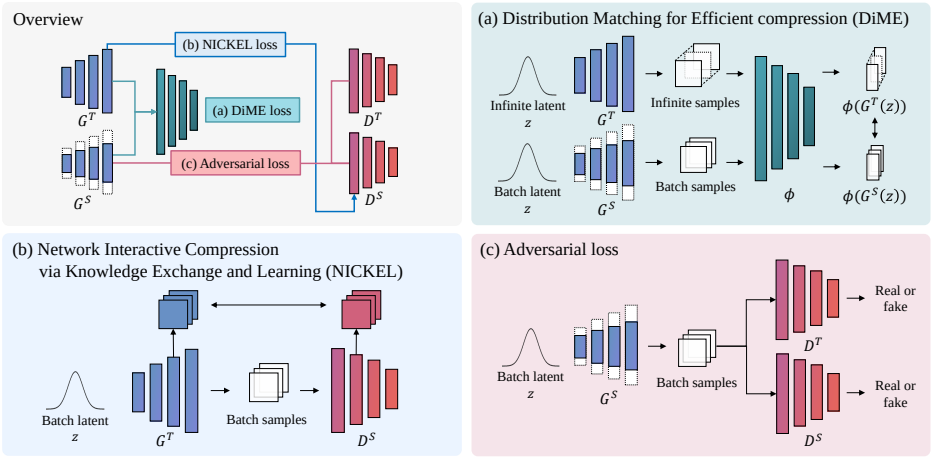
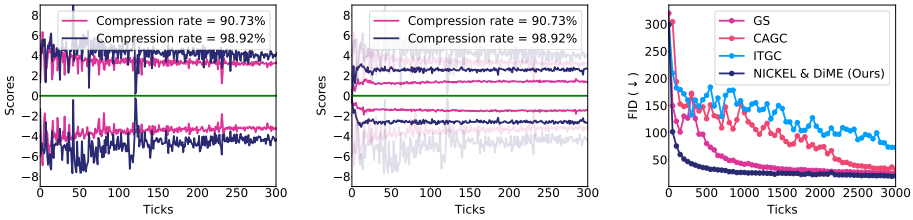


Fig. 2: A schematic overview of our method. Our method consists of (a) Distribution Matching for Efficient compression (`DiME`), (b) Network Interactive Compression via Knowledge Exchange and Learning (`NICKEL`), and (c) adversarial loss. (a) matches the outputs between the teacher generator (G^T) and the student generator (G^S) via the foundation model ϕ in the embedding space. (b) matches the intermediate features between the teacher generator and the student discriminator (D^S). (c) represents the adversarial loss between the student generator and both the teacher discriminator (D^T) and the student discriminator.

Additionally, we propose to utilize the global features of the teacher generator to reduce the sampling error. While we ideally hope for the matching of population distributions between the teacher generator (G^T) and the student generator (G^S) through knowledge distillation, in reality, there is a sampling error due to the matching between sample distributions. Since the distribution of the G^T is fixed, according to the law of large numbers, we can obtain nearly error-free statistics by precomputing a large number of samples from G^T. We provide detailed discussion in Sec. 4.5.

In addition to `DiME`, to exploit the characteristic of GAN that consists of a generator and a discriminator, we propose Network Interactive Compression via Knowledge Exchange and Learning (`NICKEL`). In GAN training, Lee *et al.* [33] has shown that the discriminator can provide more meaningful signals as feedback by learning the semantic knowledge of the generator. Inspired by Lee *et al.*, we not only distill knowledge directly between the generators (*i.e.*, `DiME`), but also distill knowledge from the more informative G^T to the student discriminator (D^S) by transmitting knowledge between generators via the discriminator indirectly. By utilizing G^T, we obtain two distinct advantages, Firstly, from the onset of training, D^S learns the rich semantic knowledge embedded within the G^T. Secondly, the G^T provides a wealth of knowledge surpassing that of G^S. Furthermore, we observe that `NICKEL` enhances the stability of GAN compression



(a) Logits of D on ITGC [21] **(b)** Logits of D on `NICKEL & DiME` **(c)** Convergence of FID

Fig. 3: Comparison of stability of ours and state-of-the-art compression methods. (a) indicates the logits of the discriminator for the pruned generator on ITGC [21]. The green solid line represents the ideal equilibrium state. When the compression rate is 98.92% (blue dash line), it shows a more severe imbalance state compared to when the compression rate is 90.73% (red dash line). (b) indicates the logits of the discriminator for the pruned generator on `NICKEL & DiME`. Our method mitigates the imbalance between the discriminator and the pruned generator. (c) indicates the FID convergence plot when the compression rate is 98.92%. `NICKEL & DiME` converges the most stably.

(see Fig. 3). To the best of our knowledge, `NICKEL` is the first method that distills the knowledge from G^T to G^S via the feedback of D^S for model compression.

Our experimental results show that `DiME` outperforms existing state-of-the-art compression methods through knowledge distillation between G^T and G^S. By applying `DiME` to StyleGAN2, which has a baseline FID of 4.02, resulted in FID scores of 11.25 and 18.32 at compression rates of 95.87% and 98.92%, respectively. This compares favorably to the state-of-the-art method [21], which achieves FID scores of 14.01 and 22.23 at the same compression rate. This demonstrates the power of using foundation models as embedding kernels for knowledge distillation. In addition, by using `NICKEL` with `DiME`, we further enhance the FID scores to 10.45 and 15.93 with improved stability, setting a new standard in GAN compression performance. It is worth to note that we achieve a reasonable performance with the FID of 29.38 at the extreme compression rate of 99.69%, surpassing the previous state-of-the-art performance by a significant margin. Our contributions can be summarized as follows:

- We propose `DiME`, an effective distillation method for GANs that ensures the matching output distributions between G^T and G^S via employing foundation models as kernels for MMD loss (Sec. 3.1). `DiME` outperforms existing GAN compression methods, achieving state-of-the-art performance in GAN compression at all compression rates.
- We propose `NICKEL` that further enhances the distillation capability by providing more meaningful feedback from D^S. We observe that `NICKEL` leads to the improvement of stability (Sec. 3.2).
- With `NICKEL & DiME`, our final model further raises the bar of the state-of-the-art. Our method shows a stable convergence with competitive performance, even at the extremely high compression rates of 99.69% (Sec. 4).
- Last but not least, we standardize and benchmark GAN compression methods using official codes, ensuring future compatibility and reproducibility.

2 Related Work

2.1 GAN Compression

Most GAN compression methods have been explored in conditional GAN settings [17, 35, 36, 62], which are unsuitable for distribution matching in unconditional GAN compression (see Fig. 1). Occasionally, to address this problem, GAN compression methods have been explored [21, 39, 59], proposing better embedding spaces or distance metrics between G^T and G^S. Wang *et al.* [57] proposed the GAN slimming (GS), a unified optimization framework, and emphasized that naïve application of compression methods leads to significant performance degradation due to the notorious instability of GANs. Liu *et al.* [39] proposed the content-aware GAN compression (CAGC) method, which focuses on only the contents of interest (*e.g.*, object, face) to distill the knowledge, but this method requires additional costs due to manual labeling of contents. Li *et al.* [36] proposed the generator-discriminator cooperative compression (GCC) to maintain the nash-equilibrium between G^S and D^S, but nash-equilibrium still cannot be maintained, in complex settings. Kang *et al.* [21] proposed the information-theoretic GAN compression (ITGC) by maximizing the mutual information between G^T and G^S. ITGC requires a lot of computational costs due to the energy-based model and MCMC sampling. Xu *et al.* [59] proposed the StyleKD that focuses on the mapping network to achieve consistent outputs between G^T and G^S. However, StyleKD can only be applied to networks based on StyleGAN. To address these issues, we propose Distribution Matching for Efficient compression (`DiME`), which matches the distribution between G^T and G^S via foundation kernels.

2.2 Discriminator Regularization

Generally, GAN compression methods are focused on the G^S, thus it is applied in the form of generator regularization for knowledge distillation. On the other hand, GCC [36] emphasized the importance of considering not only the generator but also the discriminator to maintain the Nash equilibrium state between the compressed generator and discriminator. Similar phenomena were observed by several studies [3,17,62]. To address this issue, GCC used the selective activation discriminator, which partially activates the channels of the discriminator by utilizing the capacity constraint to maintain the Nash equilibrium state. However, GCC still shows significant performance degradation due to instability. In GAN training, Lee *et al.* [33] proposed generator-guided discriminator regularization (GGDR). GGDR showed that the discriminator can learn the semantic knowledge from the generator and lead to performance improvement of the generator by providing more powerful adversarial loss as feedback. However, GGDR cannot inject meaningful knowledge of the generator into the discriminator in the early stage because the initial generator is close to being a randomly initialized generator. Inspired by GGDR, we propose the Network Interactive Compression via Knowledge Exchange and Learning (`NICKEL`), which distills the knowledge from G^T to D^S and encourages powerful feedback from D^S to G^S.

3 Method

3.1 Knowledge Distillation with Foundation Kernels MMD

Generally, knowledge distillation (KD) minimizes the distance d_{kd} (*e.g.*, wavelet loss [62]) between the outputs of G^T and G^S, encouraging G^S to mimic G^T. We can achieve more effective knowledge distillation by designing a better distance metric. In this paper, we propose Distribution Matching for Efficient compression (`DiME`), which matches the distributions between G^T and G^S in the space embedded by foundation kernels ϕ as distance d_{kd}:

$$\mathcal{L}_{KD} = d_{kd}(G^T(z), G^S(z)) = \mathbb{E}[||\phi(G^T(z)), \phi(G^S(z))||_1] \tag{1}$$

This is equivalent to using the MMD critic [9, 10, 34, 37, 50], a statistical method that matches two distributions in RKHS, assuming that the foundation kernels ϕ are characteristic kernels [50, 60].

While Eq. (1) generally shows good performance, we observe the tremendous performance degradation of all baselines (*i.e.*, GS, CAGC, ITGC, Eq. (1)) when G^S has extremely few parameters (see Fig. 3c). As shown in Fig. 3a, the Nash equilibrium breaks down when G^S has fewer parameters, which consequently leads to the performance degradation of adversarial loss. To improve the stability of KD loss in the early stage, we utilize the global features of G^T. The global features are computed by inferring over a multitude of images rather than batch images, enabling the calculation of popular distribution statistics. Utilizing the global features mitigates the sampling error induced by the batch size in KD, with detailed discussion included in Sec. 4.5.

3.2 Network Interactive Compression via Knowledge Exchange and Learning

GAN utilizes a discriminator, which is a learnable network as the loss during the training of the generator. The performance of the generator is heavily influenced by the quality of feedback provided by the discriminator. GGDR [33] showed that during GAN training, the discriminator can learn semantic knowledge from the generator. Subsequently, the discriminator provides better feedback to the generator, thus improving the performance of the generator. Inspired by GGDR, we propose `NICKEL`, which distills knowledge from G^T into D^S to provide more powerful feedback to G^S. `NICKEL` has advantages over simply applying GGDR to G^S for two reasons. First, GGDR may struggle to provide meaningful information when G^S resembles a random network during early training, whereas `NICKEL` can distill rich information from G^T from the outset. Second, in GAN compression, due to the smaller network structure of G^S, GGDR cannot provide knowledge as rich as G^T. Therefore, we propose fine-tuning D^S via `NICKEL` to learn information from G^T. However, fine-tuning D^S using the `NICKEL` loss alone is insufficient to fully leverage the information from the pre-trained discriminator. Therefore, for adversarial learning, both D^T and D^S are employed. The loss

function of NICKEL can be formulated as follows:

$$\mathcal{L}_{\texttt{NICKEL}} = \sum_{i=1}^{L} d_{\texttt{NICKEL}}(G_i^T(z), f_i(D_i^S(G^T(z)))), \tag{2}$$

where $D_i^S(G^T(z))$ and $G_i^T(z)$ represent the feature maps of the i-th layer of D^S and G^T, respectively. f_i is a linear transform to match the shape of feature maps. As Lee *et al.* [33] mentioned, the knowledge of the generator contains a lot of semantic information. Therefore, we utilize the wavelet loss [62] for $d_{\texttt{NICKEL}}$, which is good for matching semantic information.

3.3 Training Objective

In summary, our training loss for GAN compression is formulated as:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_{dino} \cdot \mathcal{L}_{dino} + \lambda_{clip} \cdot \mathcal{L}_{clip} + \lambda_{\texttt{NICKEL}} \cdot \mathcal{L}_{\texttt{NICKEL}}, \tag{3}$$

where λ_{dino} and λ_{clip} are the weights for the knowledge distillation, which utilizes the dino embedding and clip embedding, respectively. $\lambda_{\texttt{NICKEL}}$ is the weight for NICKEL loss $\mathcal{L}_{\texttt{NICKEL}}$ in Eq. (2). $\mathcal{L}_{adv}$ is the adversarial loss, which is the min-max objective function that includes both D^T and D^S. $\mathcal{L}_{dino}$ and $\mathcal{L}_{clip}$ are the knowledge distillation losses of DiME in Eq. (1).

4 Experiments

4.1 Setups

Implementation details. We set 20, 15, and 10 for λ_{dino}, λ_{clip}, and $\lambda_{\texttt{NICKEL}}$, respectively. We use the same pruned generator as CAGC, and for CLIP and DINO, we use the pretrained weights.[1] For NICKEL, we match feature maps in 1/4 resolution channels of the generator's output (e.g., 64x64 feature maps for 256 resolution) on LH, HL, and HH components of Haar wavelet. To obtain the global features, we conduct 20,000 model inferences with the batchsize of 256.

Datasets and Evaluation Metrics. We use CelebA [40], FFHQ [24], AFHQ [4], LSUN-Church and LSUN-CAT [61] datasets, and CIFAR-10 [28]. We use Fréchet Inception Distance (FID) [14] and Precision & Recall [27,30,42] to evaluate the performance of the GANs, measuring both the quality and diversity of the generated images. We note the average of the five best FID scores. Here, we use the `fvcore` library to measure FLOPs, ensuring consistency with reported FLOPs and enabling fair comparisons with existing works.[2]

[1] CLIP.pt and DINO.pt

[2] If you noticed any discrepancy in FLOPs, the discrepancy stems from (a) the typo in the original CAGC paper, where FLOPs were reported as 4.1M instead of 4.1B—a mistake recognized in subsequent papers; and (b) different libraries used. The reported FLOPs of 4.1B at 90.87% compression can be reproduced via `fvcore` library. In contrast, when used `torchprofile` library with the official NVIDIA StyleGAN2 code, it gives the FLOPs of 5.33B at 90.73%.

Table 1: Comparison of FID scores of our methods (`DiME` and `NICKEL & DiME`) and state-of-the-art compression methods on StyleGAN2 for various datasets. The dagger symbol indicates the performance of the official model provided on the original paper's GitHub repository. The right side of the arrow shows the results obtained using NVIDIA's official FID measurement code. Our reproduced models achieve higher performance compared to the official models.

Model	Dataset	Method	#params.	FLOPs	Compression rate	FID↓
StyleGAN2	FFHQ (256×256)	(1) Full model [23]	24.77M	14.90B	-	4.02
		(2) Full model†	30.03M	45.12B	-	4.5
		(3) CAGC† [39]	5.57M	4.12B	90.87%	7.9 → 10.82
		(4) DCP-GAN† [5]				6.35 → 8.93
		CAGC	8.72M	3.73B	74.96%	5.24
		ITGC [21]				5.27
		`DiME`				5.00
		`NICKEL & DiME`				**4.42**
		GS [57]	4.96M	1.38B	90.73%	10.26
		CAGC				10.06
		GCC [36]				11.19
		ITGC				10.02
		`DiME`				8.39
		`NICKEL & DiME`				**7.43**
		CAGC	2.69M	0.16B	98.92%	23.05
		ITGC				22.23
		`DiME`				18.32
		`NICKEL & DiME`				**15.93**
	FFHQ (1024×1024)	Full model	30.37M	74.27B	-	2.74
		Full model†	49.1M	74.3B	-	2.7
		(5) CAGC†	9.2M	7.0B	89.39%	7.6 → 7.53
		(6) DCP-GAN†				5.80 → 5.87
		`NICKEL & DiME`	5.65M	6.99B	90.59%	**6.41**
	LSUN Church (256×256)	Full model	30.03M	45.12B	-	3.97
		CAGC	5.57M	4.12B	90.87%	4.50
		StyleKD				4.47
		(7) DCP-GAN†				4.87 → 4.82
		`NICKEL & DiME`				**3.94**
DDPM	LSUN Church (256×256)	Full model†	113.7M	497.4B	-	10.6
		SPDM (t=100)† [7]	63.2M	277.6B	44.19%	13.9

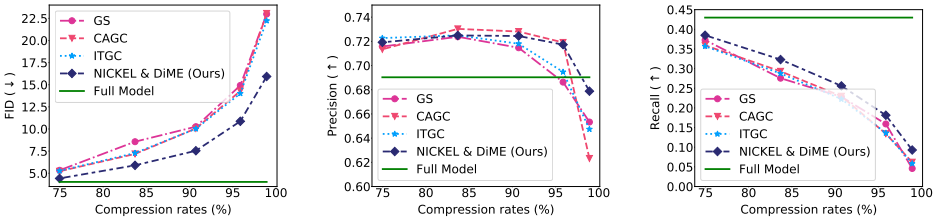
Baselines. We follow the architecture and training setups of StyleGAN2 [23], except for augmentation (not used). Our pruned generators are identical to CAGC [39]. The architecture and training setups for BigGAN and SNGAN follow Kang *et al*. [19]. We use the official code of StyleGAN2-ADA-PyTorch and StudioGAN. We also compare Structural Pruning for Diffusion Models (SPDM) [7].

Table 2: Comparison of FID scores of our methods (`DiME` and `NICKEL & DiME`) and state-of-the-art compression methods on various architectures for various datasets.

Model	Dataset	Method	#params.	FLOPs	Compression rate	FID↓
SNGAN	CIFAR-10 (32×32)	Full model [41]	4.28M	3.36B	-	17.71
		CAGC [39]	1.20M	0.85B	74.88%	40.45
		ITGC [21]				43.66
		`DiME`				31.98
		`NICKEL & DiME`				**23.11**
		CAGC	0.50M	0.31B	90.85%	51.93
		ITGC				59.99
		`DiME`				36.89
		`NICKEL & DiME`				**28.27**
BigGAN	CIFAR-10 (32×32)	Full model [1]	8.83M	3.83B	-	10.66
		CAGC	0.89M	0.37B	90.45%	**26.32**
		ITGC				27.78
		`NICKEL & DiME`				26.66
StyleGAN2	CelebA (128×128)	Full model [23]	24.53M	11.27B	-	2.70
		CAGC	3.08M	0.27B	97.60%	10.05
		ITGC				10.09
		StyleKD				10.41
		`NICKEL & DiME`				**7.56**
	AFHQ (512×512)	Full model	30.28M	59.67B	-	3.01
		`NICKEL & DiME`	10.30M	14.94B	74.96%	**3.17**
	LSUN CAT (256×256)	Full model	24.77M	14.90B	-	8.19
		GS	4.96M	1.38B	90.73%	17.11
		CAGC				12.31
		ITGC				12.06
		`DiME`				11.59
		`NICKEL & DiME`				**10.80**

4.2 Benchmarking and Reproducibility in GAN Compression

The performance metrics for previous GAN compression techniques were often based on outdated and unofficial code. To address this and for future compatibility and reproducibility in the field, we reimplement the state-of-the-art GAN compression methods using NVIDIA's official StyleGAN2-ADA code. The official code shows comparable performance to the unofficial code, but with more efficient FLOPs and fewer parameters (see (1) and (2) in Tab. 1). Additionally, we provide results re-evaluated using NVIDIA's official FID code, utilizing trained weights from the original GitHub repository of the baselines (indicated by the right of the arrow of (3)-(7) in Tab. 1). While the re-evaluated FID scores



(a) Comparison of FID **(b)** Comparison of Precision **(c)** Comparison of Recall

Fig. 4: Performance comparison as a function of compression rates on StyleGAN2 for FFHQ. (a) indicates a function showing how FID varies with compression rates. `NICKEL & DiME` consistently outperforms other state-of-the-art compression methods at various compression rates. At a compression rate of 74.96%, `NICKEL & DiME` shows only 9.68% performance degradation compared to the full model, and the performance degradation due to increasing compression rates occurs less than other state-of-the-art compression methods. (b) indicates a function showing how Precision varies with compression rates. `NICKEL & DiME` shows comparable fidelity scores to other methods. (c) indicates a function showing how Recall varies with compression rates. `NICKEL & DiME` shows better preservation of diversity compared to other methods, even with higher compression rates.

for 1024×1024 FFHQ and LSUN-Church datasets are similar to the reported FID, we find that the FID for the 256×256 FFHQ dataset shows a significant performance gap (see (3) and (4) in Tab. 1). This discrepancy arises because the reported FID scores were not calculated by computing the feature embedding directly but by using the feature embedding provided by CAGC's custom FID code. We hypothesize that the feature embedding provided by CAGC may be biased for the 256×256 FFHQ dataset.

4.3 Results

We first compare the knowledge distillation performance of `DiME`, as described in Eq. (1), with state-of-the-art GAN compression methods [21,36,39,57]. To distill the knowledge of G^T, `DiME` compares the outputs of G^T and G^S in the foundation embedding spaces (*i.e.*, DINO, CLIP). As shown in Tab. 1 and Tab. 2, `DiME` outperforms the previous compression methods on various GAN architectures and datasets. Particularly, `DiME` improves FID scores by 1.63 compared to the state-of-the-art GAN compression methods in a setting where it reduces the FLOPS of StyleGAN2 for FFHQ by 11 times, with a compression rate of 90.73%. Our experimental results show that `DiME` is highly effective for knowledge distillation.

Additionally, to investigate the effectiveness of distillation considering the characteristics of GANs (*i.e.*, `NICKEL`) beyond direct knowledge distillation (*i.e.*, `DiME`), we combine `NICKEL`, as described in Eq. (2), and `DiME`. Tab. 1 and Tab. 2 show that `NICKEL & DiME` further improves the performance over that of `DiME`. In Tab. 1, our method achieves the FID score of 15.93 by compressing StyleGAN2 93-fold. This compares to ITGC, which attains the FID score of 14.01 with a 24-

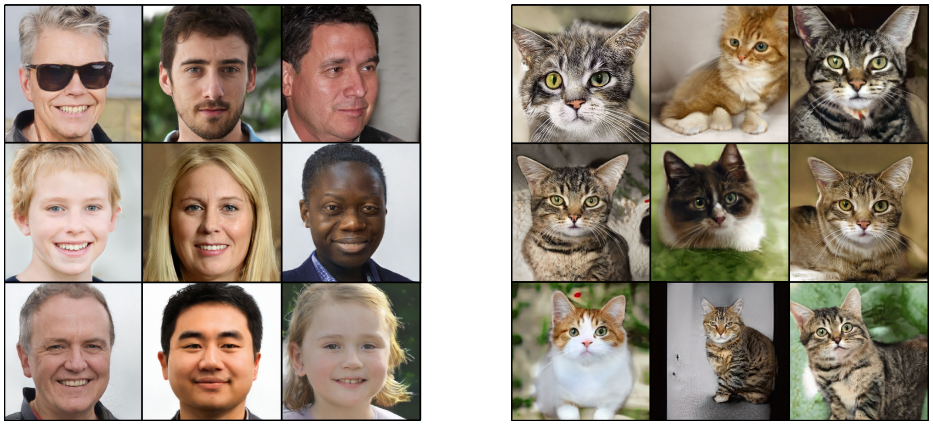
Table 3: Quantitative results of extremely compressed StyleGAN2. We compare the performance of various compression methods on StyleGAN2 for FFHQ at compression rate = 99.69%. Previous methods often suffer from severe performance degradation due to the imbalance between G^S and D^S when GAN is extremely compressed. On the other hand, `NICKEL & DiME` shows acceptable performance compared to other methods with high stability.

Model	Dataset	Method	#params.	FLOPs	Compression rate	FID↓
StyleGAN2	FFHQ (256×256)	Full model [23]	24.77M	14.90B	-	4.02
		GS [57]	2.35M	0.05B	99.69%	184.33
		CAGC [39]				186.61
		ITGC [21]				164.92
		`NICKEL & DiME`				**29.38**

fold compression. Furthermore, as shown in Tab. 1, our method not only obtains better computational efficiency but also shows superior performance compared to the state-of-the-art diffusion-model pruning method for the LSUN-Church dataset. These results indicate the continued significance of GAN compression research. As shown in Tab. 1 and Tab. 2, our method can be applied to various GAN architectures and show similar trends for various datasets.

For in-depth investigations, we compare the FID, Precision, and Recall performance of compression methods at various compression rates. Fig. 4 indicates the FID, Precision, and Recall scores at each compression rate for StyleGAN2 on FFHQ dataset. In Fig. 4a, `NICKEL & DiME` outperforms previous methods at all reported compression rates. Furthermore, at a compression rate of 74.96%, `NICKEL & DiME` shows only a 9.68% performance degradation (FID: 4.42), compared to the full model (FID:4.03). Remarkably, our method shows significant gaps with previous methods as compression rates increase, thanks to the improved stability. Fig. 4b shows precision scores, which indicate the fidelity of generated images. We observe that the precision scores of the compressed models are higher than those of the full model. While we observe a deterioration of precision scores with increasing compression rates, `NICKEL & DiME` maintains the precision scores comparable to the full model, even at high compression rates. Moreover, `NICKEL & DiME` shows precision scores comparable to the precision score of the full model up to a compression rate of 98.92%. Fig. 4c shows recall scores, indicating the diversity of the generated images. We observe that, unlike precision scores, the recall scores of the compressed models decrease significantly with increasing compression rates. Still, `NICKEL & DiME` maintains better diversity compared to the other compression methods. We provide performance comparison with recent various metrics in the supplementary for more comprehensive analysis.

As shown in Fig. 3, `NICKEL & DiME` mitigates the imbalance between G^S and D^S by considering D^S during knowledge distillation. Fig. 3a and Fig. 3b respectively show the logits of D^S for ITGC and `NICKEL & DiME`. In contrast to



(a) compression rate = 90.73% **(b)** compression rate = 90.73%

Fig. 5: Visualization of images generated by compressed StyleGAN2 on FFHQ and LSUN-CAT. (a) shows the visual quality of StyleGAN2 compressed by `NICKEL & DiME` on FFHQ at compression rate = 90.73%. (b) shows the visual quality of StyleGAN2 compressed by `NICKEL & DiME` on LSUN-CAT at compression rate = 90.73%.

the ideal training of GAN where the logits of the discriminator should be close to 0, ITGC shows significant performance degradation due to the imbalance between G^S and D^S during training. Particularly, as the compression on generator intensifies, the imbalance between G^S and D^S becomes more pronounced. On the other hand, `NICKEL & DiME` alleviates this imbalance. Even at a compression rate of 98.92%, our method maintains a better equilibrium compared to ITGC's at the compression rate of 90.73%. Fig. 3c shows the convergence of FID scores, indicating stable convergence of our method compared to the other alternatives. It is noteworthy that our method shows stable convergence even under extreme compression rates. As shown in Tab. 3, at an extreme compression rate of 99.69%, other methods fail to achieve stable learning due to the breakdown of Nash equilibrium between the highly compressed generator and discriminator. In contrast, our method not only shows stable convergence but also achieves reasonable performance, even with a 321-fold compression. Sec. 4.4 shows the visual quality of this scenario.

4.4 Visualization of a Compression Factor of 11, 92, and 321.

In Fig. 5, we show generated images for FFHQ and LSUN-CAT datasets using StyleGAN2 at a compression rate of 90.73%. Our method shows not only high visual quality but also the ability to generate diverse images. In Fig. 6, we visualize generated images at high compression rates. At a compression rate of 98.92%, our method shows visual quality that is not significantly degraded. Moreover, even at a compression rate of 99.69%, our method shows reasonable visual quality with diverse images.

(a) compression rate = 98.92% **(b)** compression rate = 99.69%

Fig. 6: Visualization of extremely compressed StyleGAN2 on FFHQ. (a) indicates the visual quality of StyleGAN2 compressed by `NICKEL & DiME` at compression rate = 98.92%. (b) indicates the visual quality of StyleGAN2 compressed by `NICKEL & DiME` at compression rate = 99.69%. `NICKEL & DiME` shows acceptable visual quality even at extreme compression rates.

4.5 Ablation Study

CLIP and DINO Embeddings. In Tab. 4, the CLIP embedding w/o global indicates using only CLIP as the embedding kernel for knowledge distillation. We observe significant challenges in achieving stable knowledge distillation when using only CLIP. In contrast, when using only DINO as the embedding kernel, DINO embedding w/o global, we observe stable convergence and achieve the FID of 20.75. In addition, we observe that although the CLIP embedding space may pose challenges in achieving stable knowledge distillation, combining it with the DINO embedding space could lead to slight performance improvements.

Utilization of Global Features. The objective of KD is to match the population distributions between G^T and G^S. However, due to the batch size, we can only match the sample distributions. Hence, a sampling error ϵ_{KD} may occur in the KD loss, which is bounded by the sum of the sampling errors of G^T and G^S:

$$\epsilon_{KD} < \epsilon_{teacher} + \epsilon_{student} \tag{4}$$

Fortunately, unlike G^S, the distribution of G^T is fixed. Therefore, by precomputing the statistics— referred to as global features —through infinite sampling, we can achieve an infinitesimal sampling error $\epsilon_{teacher}$. As shown in Tab. 4, we find that utilizing global features leads to performance enhancement. In fact, this resembles the MMD critic, which is a stable metric for learning the distribution. Santos *et al.* [50] and Yeo *et al.* [60] noted that pretrained neural networks can be considered as characteristic kernels, and reducing the discrepancy of the mean

Table 4: Ablation study results in NICKEL & DiME.

Name	Model		Global features	NICKEL	FID↓
	DINO	CLIP			
CLIP embedding w/o global		✓			152.15
DINO embedding w/o global	✓				20.75
DINO embedding	✓		✓		19.60
DiME	✓	✓	✓		18.32
NICKEL & DiME	✓	✓	✓	✓	**15.93**

between extracted features can be seen as the MMD critic. In this vein, DiME can be considered to stably match distributions between two generators.

5 Limitations

Our method shows excellent performance via distribution matching, yet it tends to focus on the fidelity of generated images. In fact, every method experiences significant degradation in recall performance, even at low compression rates (Fig. 4). Furthermore, there still remains the imbalance between the generator and discriminator at extreme compression rates, which incurs significant performance degradation. Thus, it is an interesting research direction to develop methods that are capable of maintaining diversity and stability when compressing generative models at extreme compression rates.

6 Conclusion

In this paper, we propose Distribution Matching for Efficient compression (DiME) and Network Interactive Compression via Knowledge Exchange and Learning (NICKEL) that set a new standard of the performance in GAN compression. DiME matches the distributions between the teacher generator and student generator by using the maximum mean discrepancy (MMD) as a loss function. For better matching, we harness the power of the pretrained foundation model and use it as embedding kernels in MMD loss for knowledge distillation. DiME can compress StyleGAN2 with the FID of 4.02 by 20 times while maintaining reasonable performance with the FID of 11.25, achieving the state-of-the-art performance in all compression rates. NICKEL further enhances the performance by providing better feedback to the student generator from the discriminator. Combining these two, NICKEL & DiME successfully compresses StyleGAN2 by 92 times while maintaining the FID score of 15.93. Thanks to its enhanced stability, NICKEL & DiME allows us to compress StyleGAN2 by up to 99.69% (321 times smaller) while maintaining reasonable performance, which is not possible for existing methods.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2022R1C1C1008496), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2020-II201336, Artificial Intelligence graduate school support (UNIST), No.RS-2021-II212068, Artificial Intelligence Innovation Hub, RS-2022-II220959, (Part 2) Few-Shot Learning of Causal Inference in Vision and Language for Decision Making, RS-2022-II220264, Comprehensive Video Understanding and Generation with Knowledge-based Deep Logic Neural Network). We also thank the supercomputing resources of the UNIST Supercomputing Center.

References

1. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
3. Chen, X., Zhang, Z., Sui, Y., Chen, T.: Gans can play lottery tickets too. arXiv preprint arXiv:2106.00134 (2021)
4. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8188–8197 (2020)
5. Chung, J., Hyun, S., Shim, S.H., Heo, J.P.: Diversity-aware channel pruning for stylegan compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7902–7911 (2024)
6. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
7. Fang, G., Ma, X., Wang, X.: Structural pruning for diffusion models. Advances in neural information processing systems **36** (2024)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
9. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: A kernel method for the two-sample-problem. Advances in neural information processing systems **19** (2006)
10. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research **13**(1), 723–773 (2012)
11. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. Advances in neural information processing systems **28** (2015)
12. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. Advances in neural information processing systems **33**, 9841–9850 (2020)
13. Hassibi, B., Stork, D.: Second order derivatives for network pruning: Optimal brain surgeon. Advances in neural information processing systems **5** (1992)

14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
16. Hou, L., Yuan, Z., Huang, L., Shen, H., Cheng, X., Wang, C.: Slimmable generative adversarial networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 7746–7753 (2021)
17. Hu, T., Lin, M., You, L., Chao, F., Ji, R.: Discriminator-cooperated feature map distillation for gan compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20351–20360 (2023)
18. Hu, X., Liu, X., Wang, Z., Li, X., Peng, W., Cheng, G.: Rtsrgan: Real-time super-resolution generative adversarial networks. In: 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD). pp. 321–326. IEEE (2019)
19. Kang, M., Shin, J., Park, J.: Studiogan: a taxonomy and benchmark of gans for image synthesis. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
20. Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10124–10134 (2023)
21. Kang, M., Yoo, H., Kang, E., Ki, S., Lee, H.E., Han, B.: Information-theoretic gan compression with variational energy-based model. Advances in Neural Information Processing Systems **35**, 18241–18255 (2022)
22. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
23. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. Advances in neural information processing systems **33**, 12104–12114 (2020)
24. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
25. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020)
26. Kim, H., Choi, Y., Kim, J., Yoo, S., Uh, Y.: Exploiting spatial dimensions of latent in gan for real-time image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 852–861 (2021)
27. Kim, P.J., Jang, Y., Kim, J., Yoo, J.: Topp&r: Robust support estimation approach for evaluating fidelity and diversity in generative models. Advances in Neural Information Processing Systems **36** (2024)
28. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
29. Kwon, G., Ye, J.C.: One-shot adaptation of gan in just one clip. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
30. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in neural information processing systems **32** (2019)
31. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. Advances in neural information processing systems **2** (1989)

32. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
33. Lee, G., Kim, H., Kim, J., Kim, S., Ha, J.W., Choi, Y.: Generator knows what discriminator should learn in unconditional gans. In: European Conference on Computer Vision. pp. 406–422. Springer (2022)
34. Li, C.L., Chang, W.C., Cheng, Y., Yang, Y., Póczos, B.: Mmd gan: Towards deeper understanding of moment matching network. Advances in neural information processing systems **30** (2017)
35. Li, M., Lin, J., Ding, Y., Liu, Z., Zhu, J.Y., Han, S.: Gan compression: Efficient architectures for interactive conditional gans. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5284–5294 (2020)
36. Li, S., Wu, J., Xiao, X., Chao, F., Mao, X., Ji, R.: Revisiting discriminator in gan compression: A generator-discriminator cooperative compression scheme. Advances in Neural Information Processing Systems **34**, 28560–28572 (2021)
37. Li, Y., Swersky, K., Zemel, R.: Generative moment matching networks. In: International conference on machine learning. pp. 1718–1727. PMLR (2015)
38. Lin, Y., Chen, M., Wang, W., Wu, B., Li, K., Lin, B., Liu, H., He, X.: Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15305–15314 (2023)
39. Liu, Y., Shu, Z., Li, Y., Lin, Z., Perazzi, F., Kung, S.Y.: Content-aware gan compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12156–12166 (2021)
40. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)
41. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
42. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable fidelity and diversity metrics for generative models. In: International Conference on Machine Learning. pp. 7176–7185. PMLR (2020)
43. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
44. Park, J., Son, S., Lee, K.M.: Content-aware local gan for photo-realistic super-resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10585–10594 (2023)
45. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3967–3976 (2019)
46. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2085–2094 (2021)
47. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
48. Rippel, O., Bourdev, L.: Real-time adaptive image compression. In: International Conference on Machine Learning. pp. 2922–2930. PMLR (2017)

49. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
50. Santos, C.N.d., Mroueh, Y., Padhi, I., Dognin, P.: Learning implicit generative models by matching perceptual features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4461–4470 (2019)
51. Sauer, A., Karras, T., Laine, S., Geiger, A., Aila, T.: Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. arXiv preprint arXiv:2301.09515 (2023)
52. Sauer, A., Schwarz, K., Geiger, A.: Stylegan-xl: Scaling stylegan to large diverse datasets. In: ACM SIGGRAPH 2022 conference proceedings. pp. 1–10 (2022)
53. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE transactions on pattern analysis and machine intelligence **44**(4), 2004–2018 (2020)
54. Shu, H., Wang, Y., Jia, X., Han, K., Chen, H., Xu, C., Tian, Q., Xu, C.: Co-evolutionary compression for unpaired image translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3235–3244 (2019)
55. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
56. Sreenivasan, K., Sohn, J.y., Yang, L., Grinde, M., Nagle, A., Wang, H., Xing, E., Lee, K., Papailiopoulos, D.: Rare gems: Finding lottery tickets at initialization. Advances in Neural Information Processing Systems **35**, 14529–14540 (2022)
57. Wang, H., Gui, S., Yang, H., Liu, J., Wang, Z.: Gan slimming: All-in-one gan compression by a unified optimization framework. In: European Conference on Computer Vision. pp. 54–73. Springer (2020)
58. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European conference on computer vision (ECCV) workshops. pp. 0–0 (2018)
59. Xu, G., Hou, Y., Liu, Z., Loy, C.C.: Mind the gap in distilling stylegans. In: European Conference on Computer Vision. pp. 423–439. Springer (2022)
60. Yeo, S., Jang, Y., Sohn, J.y., Han, D., Yoo, J.: Can we find strong lottery tickets in generative models? In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 3267–3275 (2023)
61. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
62. Zhang, L., Chen, X., Tu, X., Wan, P., Xu, N., Ma, K.: Wavelet knowledge distillation: Towards efficient image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12464–12474 (2022)
63. Zhang, R., Hu, X., Li, B., Huang, S., Deng, H., Qiao, Y., Gao, P., Li, H.: Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15211–15222 (2023)