Information-Theoretic GAN Compression with Variational Energy-based Model

Minsoo Kang[1] **Hyewon Yoo**[2] **Eunhee Kang**[3]

Sehwan Ki[3] **Hyong-Euk Lee**[3] **Bohyung Han**[1, 2]

[1] ECE, [2] IPAI, Seoul National University
[3] Samsung Advanced Institute of Technology (SAIT)
{kminsoo, yoohyewony, bhhan}@snu.ac.kr
{eunhee.kang, sh1004.ki, hyongeuk.lee}@samsung.com

Abstract

We propose an information-theoretic knowledge distillation approach for the compression of generative adversarial networks, which aims to maximize the mutual information between teacher and student networks via a variational optimization based on an energy-based model. Because the direct computation of the mutual information in continuous domains is intractable, our approach alternatively optimizes the student network by maximizing the variational lower bound of the mutual information. To achieve a tight lower bound, we introduce an energy-based model relying on a deep neural network to represent a flexible variational distribution that deals with high-dimensional images and consider spatial dependencies between pixels, effectively. Since the proposed method is a generic optimization algorithm, it can be conveniently incorporated into arbitrary generative adversarial networks and even dense prediction networks, e.g., image enhancement models. We demonstrate that the proposed algorithm achieves outstanding performance in model compression of generative adversarial networks consistently when combined with several existing models.

1 Introduction

Generative adversarial networks (GANs) [1] have accomplished impressive achievements on various computer vision tasks including image synthesis [2–5], image-to-image translation [6–9], video prediction [10–12], and many others. However, despite the outstanding performance, their applicability to resource-hungry systems, *e.g.*, edge or mobile devices, is limited due to high computational costs. To deal with this issue, GAN compression, a task reducing the size of generator networks, has recently drawn the attention of many machine learning researchers.

The straightforward way to compress generators is to directly employ traditional network compression techniques such as low-rank approximation [13–15], network quantization [16–19], pruning [20–25], and knowledge distillation [26–30]. However, the compression of generative models typically involves a higher-dimensional output space and requires matching the output distributions between the original and compressed models instead of the output instances given by the two models. Hence, as reported in [31–33], the synthesized images obtained from the compressed generators via the channel pruning [34, 23, 24] are often suboptimal—producing images with artifacts and suffering from image quality degradation, which implies that a naïve application of the methods designed for discriminative models is not effective for GAN compressions.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Existing approaches for compressing GANs [31, 35, 32, 36–38, 33, 39] often utilize knowledge distillation and manage to achieve competitive accuracy of student models. Specifically, these methods transfer the knowledge of a teacher network to a student model via intermediate representations, final outputs, and embedded features [40, 41]. To this end, the existing algorithms employ ℓ_1 or ℓ_2 norms to compute the difference between the representations of the two networks, but they are not necessarily effective for minimizing the distance between the two distributions.

We propose to compress GAN models by maximizing the mutual information for knowledge distillation between the outputs of a teacher and a student. For the compression of generative models, the use of the mutual information between two random variables makes more sense than the direct pixel-wise comparison between two outputs, which is the standard way to transfer knowledge from a teacher to a student in discriminative models. However, because the computation of the mutual information is generally intractable in the continuous domain, we employ a variational approach [42], where the lower bound of the mutual information is alternatively optimized using a variational distribution, an approximation for the conditional distribution of the teacher output given the student output. We further adopt an energy-based model for the flexible representation of the variational distribution and tighten the lower bound for effective knowledge transfer. Note that the proposed energy-based model is a convenient way to deal with high-dimensional data (images) and an effective tool to handle spatial dependency between pixels within an image. Combined with the existing approaches, our framework achieves substantial performance improvement on GAN compression tasks. The main contributions of our work are summarized below:

- We propose a novel energy-based framework of GAN compression for knowledge distillation, which maximizes the mutual information between a teacher and a student.
- The proposed energy-based approach is effective for representing a flexible variational distribution and can be incorporated into many GAN compression methods thanks to its generality.
- Experimental results verify that the proposed method improves performance consistently when combined with the existing methods.

The rest of the paper is organized as follows. Section 2 discusses the related work to GAN compression. The details of our approach are described in Section 3 and 4, and the experimental results are presented in Section 5. Finally, we conclude this paper in Section 6.

2 Related Work

2.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [1] consist of two components, a generator and a discriminator, and train the models using an adversarial loss; the generator aims to fool the discriminator by synthesizing realistic images from random noise samples while the discriminator attempts to distinguish the real data from the fake ones. Unlike the vanilla GANs, conditional GANs [43, 6, 7] can control the image generation conditioned on inputs such as a class label [43] or an image translated into another image [6, 7] (*e.g.*, Horse $\rightarrow$ Zebra). The conditional GANs are not confined to the conditional image generation tasks, but applied to various areas such as super-resolution [44], text-to-image [45], and image inpainting [46]. Since the size of the conditional GANs is typically large in terms of FLOPs, the proposed GAN compression algorithm is mainly tested on them but we also verify the effectiveness of our approach on the unconditional counterpart.

2.2 Knowledge Distillation

Knowledge distillation techniques aim to transfer knowledge from a powerful network (*i.e.*, teacher) to a compact network (*i.e.*, student) by matching logits [47], output distributions [26], intermediate activations [48], or attention maps [27]. Specifically, Hinton *et al*. [26] learn the representation of the student by minimizing the relative entropy from its normalized softmax output to the teacher's one while [47, 48] minimize the mean squared error between the representations of the teacher and student networks. Both Variational Information Distillation (VID) [49] and Contrastive Representation Distillation (CRD) [50] employ the mutual information between the teacher and student to estimate the similarity of the representations from the two networks. The former relies on a variational

approach [42] for the tractable computation of mutual information between the two networks. On the other hand, the latter derives an InfoNCE [51]-like loss function from the lower bound of the mutual information to learn the representation of the student. However, VID maximizes the variational lower bound using a strong assumption about the variational distribution, a fully factorized Gaussian distribution, and CRD is difficult to be employed to generative models due to its contrastive learning objective, which is prone to be trivial in image generation tasks without semantic attribute annotations, *e.g.*, class labels. Our algorithm is related to VID and CRD since they all attempt to maximize the mutual information between the teacher and student networks, but has clear advantage over them in terms of the flexibility of the variational distribution and the applicability to generative models.

2.3 GAN Compression

GAN compression algorithms typically target to reduce the size of the generator without considering that of the discriminator since it is not employed for inference. Co-Evolution [31] utilizes an evolutionary algorithm [52] to reduce the channel size in the generator of CycleGAN [7] based on a fitness formulated based on FLOPs and the mean squared reconstruction error between the original uncompressed and pruned generators. GAN-Compression [35] employs a once-for-all network training scheme [53] combined with the reconstruction loss as well as the intermediate feature distillation loss [48], and selects the best performing subnetwork given a target budget. Compression and Teaching (CAT) [38] designs an efficient building block for the generator, and prunes the network using the magnitude-based channel pruning method [24] while maximizing the similarity between the two networks based on centered kernel alignment [54]. Content-Aware GAN Compression (CAGC) [33] uses the feature distillation and reconstruction loss based only on the object part of the generated image while additionally considering LPIPS [41] as a perceptual distance for the synthesized one. On the contrary, Online Multi-Granularity Distillation (OMGD) [39] iteratively trains teacher networks with original conditional GAN objectives and a student network using the total variation loss [55] for spatial smoothness and a distillation loss that makes the student mimic the final outputs and channel-wise attention weights for the intermediate feature maps of the teacher networks. Generator-discriminator Cooperative Compression (GCC) [56] also adopts the online distillation scheme while selecting channels in the original discriminator to match the reduced capacity of the student generator at each iteration. To validate the effectiveness of our framework, we present how the proposed algorithm can improve performance via the combinations with existing methods such as GCC, OMGD, CAGC, CAT, and GAN-Compression.

3 Variational GAN Compression with Energy-based Model

Our goal is to compress generators in GANs through knowledge distillation. Let S and T be random variables of the outputs from a student, $G^s(\cdot;\phi^s) : \mathcal{X} \rightarrow \mathcal{Y}$ and a teacher, $G^t(\cdot;\phi^t) : \mathcal{X} \rightarrow \mathcal{Y}$, respectively, where ϕ^s and ϕ^t are their model parameters. For knowledge transfer from the teacher to the student, we maximize the mutual information between T and S, $I(T;S)$, which is given by

$$I(T;S) \equiv D_{\mathrm{KL}}\Big(p(\boldsymbol{t},\boldsymbol{s}) \,||\, p(\boldsymbol{t}) \cdot p(\boldsymbol{s})\Big) = H(T) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log p(\boldsymbol{t}|\boldsymbol{s})], \tag{1}$$

where $D_{\mathrm{KL}}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) divergence and $H(\cdot)$ means the entropy function. Note that all the expectations are taken over the joint data distribution, $p(\boldsymbol{t},\boldsymbol{s})$, where $\boldsymbol{t}$ and $\boldsymbol{s}$ indicate realizations of the teacher and student output distributions, respectively.

A practical challenge in this objective is that the computation of the mutual information for multivariate continuous random variables is typically intractable. To tackle this issue, we adopt a variational method [42], where we derive the lower bound of the mutual information, $\tilde{I}(T;S)$, using a variational distribution, $q(\boldsymbol{t}|\boldsymbol{s})$, which is given by

$$\begin{aligned} I(T;S) &= H(T) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log q(\boldsymbol{t}|\boldsymbol{s})] + \mathbb{E}_{\boldsymbol{s}}[D_{\mathrm{KL}}(p(\boldsymbol{t}|\boldsymbol{s})||q(\boldsymbol{t}|\boldsymbol{s}))] \\ &\geq H(T) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log q(\boldsymbol{t}|\boldsymbol{s})] \\ &:= \tilde{I}(T;S). \end{aligned} \tag{2}$$

To maximize the mutual information, we increase the lower bound by minimizing the KL-divergence between the true and variational distributions, which is defined as

$$\triangle I(T;S) := I(T;S) - \tilde{I}(T;S) = \mathbb{E}_{\boldsymbol{s}}[D_{\mathrm{KL}}(p(\boldsymbol{t}|\boldsymbol{s})||q(\boldsymbol{t}|\boldsymbol{s}))]. \tag{3}$$

Variational Information Distillation (VID) [49] optimizes the variational lower bound by employing the fully factorized Gaussian distribution as the variational distribution $q(\boldsymbol{t}|\boldsymbol{s})$. However, as noted in [57], the statistics of the natural images usually exhibit non-Gaussian properties. Furthermore, optimizing the lower bound using the fully factorized distribution encourages the student to generate blurry outputs [58, 46, 59]. To the contrary, we employ an energy-based model as a flexible distribution for the variational distribution, which is given by

$$q_\theta(\boldsymbol{t}|\boldsymbol{s}) = \frac{1}{Z_\theta(\boldsymbol{s})} \cdot \exp\Big(- E_\theta\big(\boldsymbol{t}, \boldsymbol{s}\big)\Big), \tag{4}$$

where $E_\theta(\boldsymbol{t}, \boldsymbol{s}) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the energy function defined by a deep neural network parametrized with θ and $Z_\theta(\boldsymbol{s})$ is the partition function. The variational distribution given by the energy-based model is effective for the student model to maximize the variational lower bound. In addition, differentiated from VID [49], our approach employs the variational distribution that considers the spatial dependency among pixels in $\boldsymbol{t}$ and $\boldsymbol{s}$.

Overall, we optimize the energy-based model, the student network, and the teacher network, which is achieved by the following iterative procedure:

1. **Fix $\{\phi^s, \phi^t\}$ and optimize θ.**
   Under the fixed parameters $\{\phi^s, \phi^t\}$, minimize the objective in (3) with respect to θ.
2. **Fix $\{\theta, \phi^t\}$ and optimize ϕ^s.**
   Under the fixed parameters $\{\theta, \phi^t\}$, maximize the objective in (2) while minimizing the standard knowledge distillation losses with respect to ϕ^s.
3. **Optimize ϕ^t.**
   Minimize the standard conditional GAN loss [6, 7] with respect to ϕ^t using a training dataset. This step is only required for online knowledge distillation. Otherwise, the teacher model should be learned offline in advance.

These three steps are repeated until convergence.

4 Training Procedures

This section describes the optimization procedure of our approach for knowledge distillation, which iteratively maximizes the mutual information between a teacher and a student using the energy-based model.

4.1 Optimization of Energy-based Model

The first step of the iterative procedure is the optimization of the energy-based model for minimizing $\mathbb{E}_{\boldsymbol{s}}[D_{\mathrm{KL}}(p(\boldsymbol{t}|\boldsymbol{s})||q_\theta(\boldsymbol{t}|\boldsymbol{s}))]$ with respect to θ given the fixed model parameters $\{\phi^s, \phi^t\}$.

4.1.1 Objective

Since the conditional data distribution $p(\boldsymbol{t}|\boldsymbol{s})$ is constant with respect to the model parameter θ, the energy-based model minimizes the following objective:

$$\min_\theta \triangle I(T; S) \iff \min_\theta \mathbb{E}_{\boldsymbol{s},\boldsymbol{t}}[-\log q_\theta(\boldsymbol{t}|\boldsymbol{s})]. \tag{5}$$

To optimize the energy-based model using the standard stochastic gradient descent method, we derive the gradient of the objective in (5), which is given by

$$\nabla_\theta \mathbb{E}_{\boldsymbol{s},\boldsymbol{t}}[-\log q_\theta(\boldsymbol{t}|\boldsymbol{s})] = \mathbb{E}_{\boldsymbol{s},\boldsymbol{t}}\Big[\nabla_\theta E_\theta(\boldsymbol{t}, \boldsymbol{s})\Big] - \mathbb{E}_{\boldsymbol{s}}\mathbb{E}_{\tilde{\boldsymbol{t}} \sim q_\theta(\boldsymbol{t}|\boldsymbol{s})}\Big[\nabla_\theta E_\theta(\tilde{\boldsymbol{t}}, \boldsymbol{s})\Big], \tag{6}$$

where the second expectation of the right-hand side requires sampling from $q_\theta(\boldsymbol{t}|\boldsymbol{s})$ via a Markov Chain Monte Carlo (MCMC) method, for example, Gibbs sampling.

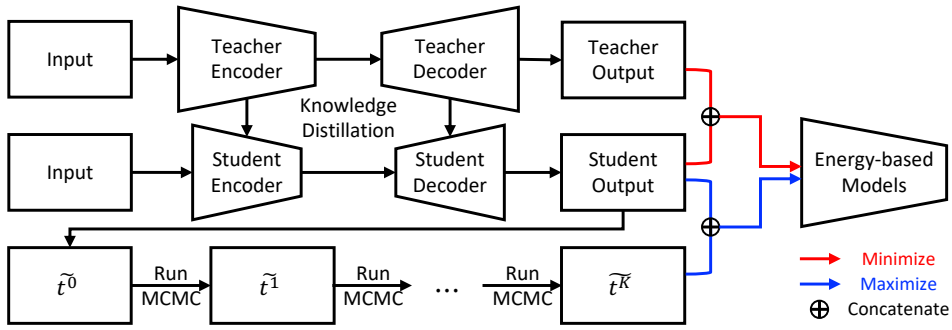

Figure 1: Overview of our framework for maximizing the variational lower bound with respect to the student model.

4.1.2 Improved MCMC sampling

The main drawback of the MCMC methods is the high computational cost, so we rely on the Langevin dynamics, which leads to the following recursive sample generation:

$$\tilde{\boldsymbol{t}}^k = \tilde{\boldsymbol{t}}^{k-1} - \frac{\lambda}{2}\nabla_{\boldsymbol{t}} E_\theta(\tilde{\boldsymbol{t}}^{k-1}, \boldsymbol{s}) + w^k, \tag{7}$$

where w^k is a sample drawn from $\mathcal{N}(0, \lambda I)$ and λ is a fixed step size. Theoretically, $\tilde{\boldsymbol{t}}^k$ becomes identical to a sample drawn from $q_\theta(\boldsymbol{t}|\boldsymbol{s})$ as $k \to \infty$ and $\lambda \to 0$ [60]. Following Short-Run MCMC [61], we also run K-step MCMC starting from an initial distribution, q_0, to approximate samples from $q_\theta(\boldsymbol{t}|\boldsymbol{s})$, where K is the number of MCMC steps.

4.1.3 Initial distribution

We initialize $\tilde{\boldsymbol{t}}^0$ as the output of the student network for the acceleration of the MCMC optimization via the Langevin sampling process. This initialization is a good approximation to the model distribution; intuitively, since the student model is supposed to mimic the samples drawn from the data distribution, the optimization of $q_\theta(\boldsymbol{t}|\boldsymbol{s})$ towards the data distribution $p(\boldsymbol{t}|\boldsymbol{s})$ from the current student output would be more efficient. As a result, the proposed initialization strategy often results in improved performance of student models compared to the existing methods [61–65] even with a small number of MCMC steps. On the contrary, we can naturally adopt the data distributions of teacher networks, which are static in Contrastive Divergence (CD) [62] and dynamic in Persistent Contrastive Divergence (PCD) [63, 64], for setting q_0 while Short-Run MCMC [61] can initialize the variational distribution using the uniform noise distribution. Since the data distribution is prone to induce bias in the CD estimate [65] and the random uniform distribution is far from the data distribution, the existing methods require many steps of MCMC and consequently increase training time [61]. In addition, PCD updates all samples in the memory buffer, which leads to significantly increased memory overhead especially for generation tasks.

4.1.4 Inference

In contrast to the previous approaches for energy-based models [61, 65, 64], we do not require any MCMC sampling for inference but generate images via a single forward process of the student generator. Therefore, the proposed algorithm incurs no extra computational cost at inference time.

4.2 Optimization of Student Generator

4.2.1 Loss for mutual information

The student generator learns to maximize the variational lower bound $\tilde{I}(T;S)$ in (2) with respect to ϕ_s, where the objective function is formally given by

$$\max_{\phi^s} \tilde{I}(T;S) \iff \max_{\phi^s} H(T) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log q_\theta(\boldsymbol{t}|\boldsymbol{s})] \iff \max_{\phi^s} \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log q_\theta(\boldsymbol{t}|\boldsymbol{s})]. \tag{8}$$

Table 1: Performance of VEM in comparison with the state-of-the-art methods for Pix2Pix model using the U-Net baseline on the Edges $\rightarrow$ Shoes and Cityscapes datasets. Methods with an asterisk (*) are our reproduction given by running the official codes.

Dataset	Method	MACs	# of Parameters	FID (↓)	mIoU (↑)
Edges $\rightarrow$ Shoes	Original [6]	18.60G (1.0×)	54.40M (1.0×)	34.31	-
	DMAD [37]	2.99G (6.2×)	2.13M (25.5×)	46.95	-
	OMGD* [39]	1.22G (15.3×)	3.40M (16.0×)	27.39	-
	OMGD* [39] + VID [49]	1.22G (15.3×)	3.40M (16.0×)	28.02	-
	OMGD* [39] + VEM (Ours)	1.22G (15.3×)	3.40M (16.0×)	**24.59**	-
Cityscapes	Original [6]	18.60G (1.0×)	54.40M (1.0×)	-	42.71
	DMAD [37]	3.96G (4.7×)	1.73M (31.4×)	-	40.53
	GCC [56]	3.09G (6.0×)	-	-	42.88
	OMGD* [39]	1.22G (15.3×)	3.40M (16.0×)	-	47.52
	OMGD* [39] + VID [49]	1.22G (15.3×)	3.40M (16.0×)	-	47.79
	OMGD* [39] + VEM (Ours)	1.22G (15.3×)	3.40M (16.0×)	-	**49.39**

Note that the entropy, $H(T)$, is independent of the parameters in the student network and the objective is equivalent to maximize the expected log likelihood as shown in the above equation. Since the joint distribution $p(\boldsymbol{t}, \boldsymbol{s})$ depends on ϕ_s, we employ a reparameterization trick using the function G^s. Then, the gradient of the objective function in (8) with respect to ϕ_s is derived as follows:

$$\begin{aligned}&\nabla_{\phi^s}\mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[-\log q_\theta(t|s)]\\&\quad=\mathbb{E}_{\boldsymbol{x}}\Big[\nabla_{\phi^s}E_\theta\big(G^t(\boldsymbol{x};\phi^t),G^s(\boldsymbol{x};\phi^s)\big)\Big]-\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\tilde{\boldsymbol{t}}\sim q_\theta\left(\boldsymbol{t}|G^s(x;\phi^s)\right)}\Big[\nabla_{\phi^s}E_\theta(\tilde{\boldsymbol{t}},G^s\big(\boldsymbol{x};\phi^s)\big)\Big].\end{aligned} \tag{9}$$

Note that we do not need to run MCMC additionally to obtain the samples from $q_\theta(\boldsymbol{t}|G^s_{\phi_s}(x))$ because they are already available from the training procedure for the energy-based model. Figure 1 illustrates the overview of our framework for maximizing the mutual information.

4.2.2 Algorithm-specific loss

The proposed variational lower bound maximization framework using the energy-based model is incorporated into knowledge distillation methods such as OMGD [39], GCC [56], GAN-Compression [35], CAT [38], and CAGC [33]. The brief descriptions about these methods are presented in Section 2.3, and we discuss the details of the algorithm-specific losses in the supplementary document.

4.2.3 Combined loss

The full objective function for the student network is given by

$$\mathcal{L}^s=\mathcal{L}_{\text{algo}}-\lambda_{\text{MI}}\cdot\tilde{I}(T;S), \tag{10}$$

where $\mathcal{L}_{\text{algo}}$ is the algorithm-specific loss and $\tilde{I}(T;S)$ is the variational lower bound defined in (8) while λ_{MI} is a hyperparameter balancing the two terms.

4.3 Optimization of Teacher Generator

The teacher generator is trained by the standard method of the corresponding GAN algorithm, *e.g.*, Pix2Pix [6] and CycleGAN [7]. Optionally, the teacher model learns to minimize the ℓ_1 distance between the generated image and the ground-truth when the ground-truth is available. In the case of the unsupervised image-to-image translation tasks, the cycle consistency loss [7] based on the ℓ_1 distance may enforce the instance mapping between two domains. To sum up, the teacher network is optimized by the adversarial loss, denoted by the GAN loss, $\mathcal{L}_{\text{GAN}}$, and the reconstruction loss, $\mathcal{L}_1$, as follows:

$$\min_{G^t}\max_{D^t}\mathcal{L}_{\text{GAN}}+\lambda_1\mathcal{L}_1, \tag{11}$$

where λ_1 is a hyperparameter to balance the two loss terms. Note that, for unconditional GANs, there is nothing to reconstruct and we train the network based only on the GAN loss.

Table 2: Performance comparison between VEM and the state-of-the-art methods in CycleGAN model based on the ResNet backbones.

Dataset	Method	MACs	# of Parameters	FID (↓)
Horse → Zebra	Original [7]	56.80G (1.0×)	11.30M (1.0×)	61.53
	Co-Evolution [31]	13.40G (4.2×)	-	96.15
	GAN-Slimming [36]	11.30G (23.6×)	-	86.09
	Auto-GAN-Distiller [32]	6.39G (8.9×)	-	83.60
	GAN-Compression [35]	2.67G (21.3×)	0.34M (33.2×)	64.95
	DMAD [37]	2.41G (23.6×)	0.20M (40.0×)	62.96
	CAT [38]	2.55G (22.3×)	-	60.18
	GCC [56]	2.40G (23.7×)	-	59.31
	OMGD* [39]	1.41G (40.3×)	0.14M (82.5×)	57.14
	OMGD* [39] + VEM (Ours)	1.41G (40.3×)	0.14M (82.5×)	**50.83**
Summer → Winter	Original [7]	56.80G (1.0×)	11.30M (1.0×)	79.12
	Co-Evolution [31]	11.10G (5.1×)	-	78.58
	Auto-GAN-Distiller [32]	4.34G (13.1×)	-	78.33
	DMAD [37]	3.18G (17.9×)	0.30M (37.7×)	78.24
	OMGD* [39]	1.41G (40.3×)	0.14M (82.5×)	75.20
	OMGD* [39] + VEM (Ours)	1.41G (40.3×)	0.14M (82.5×)	**74.04**

Table 3: Performance comparison between VEM and the existing compression algorithms for unconditional GANs on the CelebA and FFHQ datasets using SAGAN and StyleGAN2, respectively.

Model	Dataset	Method	MACs	Compression Ratio	FID (↓)
SAGAN [4]	CelebA	Original	23.45M	-	24.87
		Slimming [24]	15.45M	34.12%	36.60
		GCC* [56]	15.45M	34.12%	27.91
		GCC* [56] + VEM (Ours)	15.45M	34.12%	**25.27**
StyleGAN2 [66]	FFHQ	Original	45.1B	-	4.50
		GAN-Slimming [36]	5.0B	88.91%	12.40
		CAGC [33]	4.1B	90.91%	7.90
		CAGC [33] + VEM (Ours)	4.1B	90.91%	**7.48**

5 Experiments

To validate the effectiveness of the proposed compression algorithm based on the variational energy-based model, denoted by VEM, we conduct extensive experiments on the standard datasets using various architectures.

5.1 Datasets and Models

Image-to-image translation We adopt Pix2Pix [6] on Edges → Shoes [67] and Cityscapes [68] datasets while taking CycleGAN [7] on Horse → Zebra [7] and Summer → Winter [7]. Note that Pix2Pix is the image-to-image translation network for paired datasets while CycleGAN is mainly proposed for unpaired image-to-image translation tasks. Following the original models [6, 7], we employ a U-Net-based generator [69] for Pix2Pix and a ResNet-based generator [70] for CycleGAN. For all the datasets, images are resized to 256×256 before feeding them into the models.

Image generation To evaluate the performance of our approach for unconditional GANs, we employ StyleGAN2 [66] on Flickr-Faces-HQ (FFHQ) [71] and Self-Attention GAN (SAGAN) [4] on CelebA [72]. For image preprocessing, we resize images to 64×64 for CelebA and 256×256 for FFHQ.

5.2 Evaluation Metrics

Following the protocol of the previous approaches, we report Fréchet Inception Distance (FID) [73] scores on Edges → Shoes, Horse → Zebra, Summer→Winter, CelebA, and FFHQ while the mean Intersection over Union (mIoU) is selected for Cityscapes. The FID score is widely utilized for

assessing the quality of the generated images by measuring the distance between the distributions of embedding features extracted from generated images and real images, which are given by the Inception V3 model [74] pretrained on ImageNet. In the case of the Cityscapes, we adopt DRN-D-105 [75] as a segmentation model to derive the mIoU scores for the generated images.

5.3 Implementation Details

The proposed algorithm is implemented in PyTorch [76] based on the publicly available code[1]. Additionally, we incorporate the proposed mutual information maximization framework into existing approaches by using the official codes of GAN-Compression[2] [35], CAT[3] [38], CAGC (Content-Aware GAN Compression)[4] [33], and GCC (Generator-discriminator Cooperative Compression)[5] [56] to validate its benefit and generality. For fair comparisons, we do not modify either experimental settings or network structures of the compared algorithms. Specifically, we set the batch size to 4 for Pix2Pix, 1 for CycleGAN, 64 for SAGAN, and 16 for StyleGAN2 while we adopt the Adam optimizer with an initial learning rate of 0.0002, which is linearly decayed to zero.

When we train our energy-based model, we employ the Adam optimizer with a learning rate of 0.0001. Note that we only run 10 steps of Langevin dynamics to draw samples from the energy-based model to save training time. We set the standard deviation of random noise to 0.005 and use a step size of 100 for each gradient step of Langevin dynamics for all datasets except for the Cityscapes and FFHQ datasets. For these two datasets, we use a step size of 50 instead. Details about the hyperparameter settings are provided in the supplementary document. In the case of the paired datasets such as Edges $\rightarrow$ Shoes and Cityscapes, we maximize the mutual information between the outputs given by the student model and the real outputs instead of the teacher outputs.

5.4 Results

Pix2Pix Table 1 presents the results from Pix2Pix on Edges $\rightarrow$ Shoes and Cityscapes by the proposed algorithm using the U-Net backbone in comparison to the original network and recent methods including OMGD [39] and OMGD [39] + VID [49]. As shown in the table, OMGD + VEM achieves the best performance in terms of FID and MAC among all compared algorithms on Edges $\rightarrow$ Shoes. On Cityscapes, OMGD + VEM presents the highest mIoU by large margins even with the highest compression rate. Note that we report the results from OMGD based on our reproduction using the official code, which are denoted by asterisks (*) in all tables.

CycleGAN We test our compression algorithm using CycleGAN on Horse $\rightarrow$ Zebra and Summer $\rightarrow$ Winter. As shown in Table 2, OMGD + VEM outperforms all the compared methods on the both datasets, especially by large margins in Horse $\rightarrow$ Zebra.

Unconditional GANs We compress SAGAN and StyleGAN2 using GCC [56] and CAGC [33], respectively. Table 3 illustrates the proposed method is also effective to improve performance of the unconditional GAN compression techniques. In particular, our approach also works well with a large-scale generator such as StyleGAN2.

5.5 Ablation Study about Mutual Information Maximization Techniques

For maximizing mutual information between the teacher and student networks, one can use a fully-factorized Gaussian distribution for the variational distribution followed by VID [49] or CRD [50] employing InfoNCE-like contrastive loss. To validate the effectiveness of our energy-based models compared with VID and CRD, we perform an ablation study to combine with the state-of-the-art GAN compression approaches such as OMGD [39], GAN-Compression [35], CAT [38], GCC [56] and CAGC [33].

Table 4 demonstrates that VEM consistently outperforms VID and CRD in all the tested algorithms. This is because, compared to VID, we derive a tighter lower bound of the mutual information

[1] `https://github.com/bytedance/OMGD`

[2] `https://github.com/mit-han-lab/gan-compression`

[3] `https://github.com/snap-research/CAT`

[4] `https://github.com/lychenyoko/content-aware-gan-compression`

[5] `https://github.com/sjleo/gcc`

Table 4: Ablation study about the approaches maximizing mutual information. We compare the methods relying on the proposed energy-based model (VEM) with the approaches based on the fully-factorized Gaussian distribution (VID) and the InfoNCE-like contrastive-loss (CRD). This experiment is conducted on the Horse → Zebra, Summer → Winter, CelebA, and FFHQ datasets.

Model	Dataset	Method	MACs	FID (↓)
CycleGAN [7]	Horse → Zebra	OMGD* [39]	1.41G	57.14
		OMGD* [39] + VID [49]	1.41G	65.73
		OMGD* [39] + CRD [50]	1.41G	70.59
		OMGD* [39] + VEM (Ours)	1.41G	**50.83**
		CAT* [38]	2.56G	64.79
		CAT* [38] + VID [49]	2.56G	68.71
		CAT* [38] + CRD [50]	2.56G	67.69
		CAT* [38] + VEM (Ours)	2.56G	**52.44**
		GAN-Compression* [35]	2.55G	59.13
		GAN-Compression* [35] + VID [49]	2.55G	62.36
		GAN-Compression* [35] + CRD [50]	2.55G	60.27
		GAN-Compression* [35] + VEM (Ours)	2.55G	**50.01**
	Summer → Winter	OMGD* [39]	1.41G	75.20
		OMGD* [39] + VID [49]	1.41G	74.39
		OMGD* [39] + CRD [50]	1.41G	74.54
		OMGD* [39] + VEM (Ours)	1.41G	**74.04**
SAGAN [4]	CelebA	GCC* [56]	15.45M	27.91
		GCC* [56] + VID [49]	15.45M	26.57
		GCC* [56] + CRD [50]	15.45M	30.93
		GCC* [56] + VEM (Ours)	15.45M	**25.27**
StyleGAN2 [66]	FFHQ	CAGC [33]	4.1B	7.90
		CAGC [33] + VID [49]	4.1B	7.51
		CAGC [33] + CRD [50]	4.1B	7.65
		CAGC [33] + VEM (Ours)	4.1B	**7.48**

Table 5: Sensitivity analysis for λ_{MI} on the Cityscapes dataset using Pix2Pix under the U-Net backbone when combined with OMGD.

λ_{MI}	OMGD [39] (0.00)	0.05	0.10	0.20
mIoU	47.52	47.56	48.90	**49.39**

theoretically and consider spatial dependencies among pixels thanks to the more flexible variational distribution based on the energy function. Also, note that the application of CRD to generative models is not effective since the instance-level contrastive loss is prone to be trivial due to large gaps between positive and negative pairs.

5.6 Effect of λ_{MI} for Mutual Information

We analyze the effect of balancing term λ_{MI} for maximizing the variational lower bound on mutual information on the Cityscapes dataset using Pix2Pix under U-Net backbone by varying λ_{MI}. As shown in Table 5, VEM is not sensitive to the hyperparameter.

5.7 Qualitative Results

Figure 2 visualizes generated images given by the original uncompressed backbones and the compressed counterparts given by OMGD and OMGD + VEM for Pix2Pix and CycleGAN on Horse → Zebra, Summer → Winter, Edges → Shoes, and Cityscapes datasets. The presented images imply that our approach is effective to enhance the generation quality. For example, OMGD often produces unusual artifacts in the background in Cityscapes while OMGD + VEM is successful. Figure 3 depicts generation results given by unconditional GANs on CelebA and FFHQ datasets. Compared with the uncompressed versions, the proposed method preserves or often improves the generation

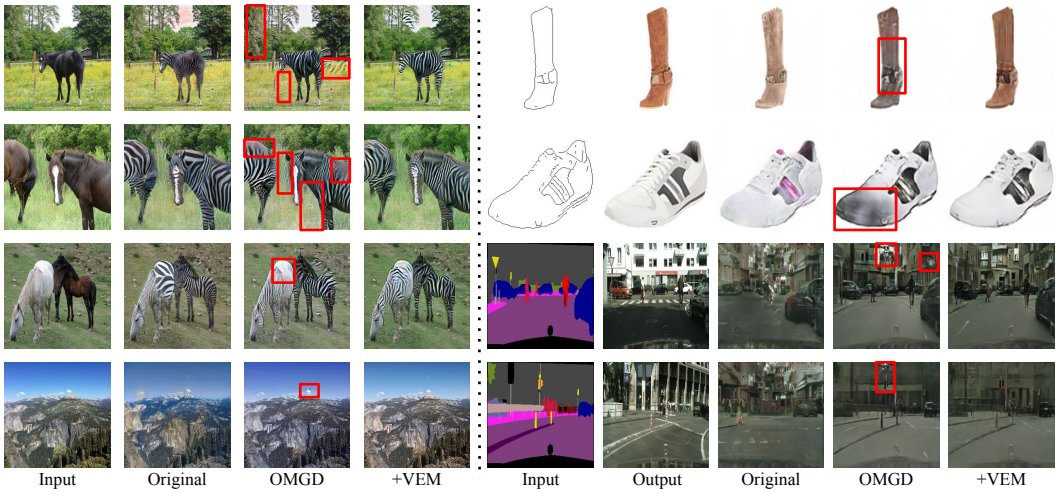

Figure 2: Qualitative results of CycleGAN and Pix2Pix on Horse → Zebra, Summer → Winter, Edges → Shoes, and Cityscapes datasets. Note that “output" denotes the ground-truth and “original" represents the generator without compression for Pix2Pix and CycleGAN.

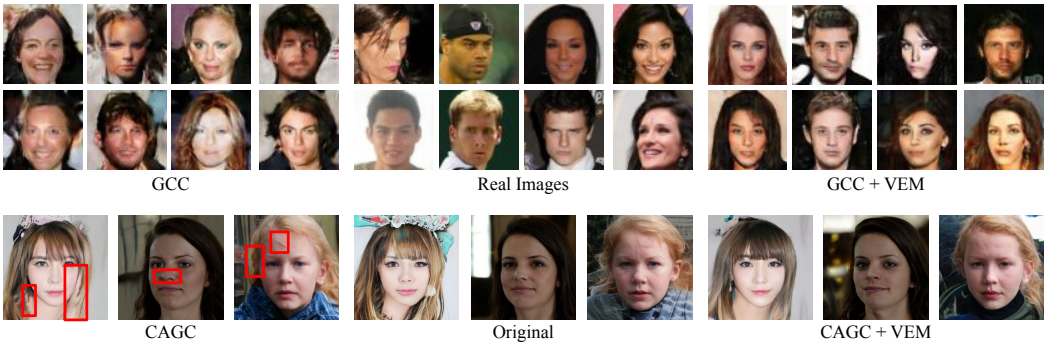

Figure 3: Qualitative results of SAGAN on the CelebA dataset (top) and StyleGAN2 on the FFHQ dataset (bottom). Note that “real images" denotes data samples in CelebA while “original" represents the images generated by uncompressed StyleGAN2.

quality even with large compression ratios. Also, our approach synthesizes visually improved images compared to the previous GAN compression methods such as GCC and CAGC when combined with them.

6 Conclusion

We presented a novel GAN compression approach via knowledge distillation based on the mutual information, which maximizes the variational lower bound by incorporating an energy-based variational distribution. Our energy-based model is straightforward to be incorporated into various existing model compression algorithms for generative models. It provides a more flexible variational distribution, which leads to theoretically tighter lower bounds and facilitates the consideration of the dependency between pixels. The proposed algorithm demonstrates outstanding performance in various GAN compression scenarios on multiple datasets compared to the state-of-the-art algorithms.

Acknowledgments This work was supported by SAIT, Samsung Electronics Co., Ltd, the Bio & Medical Technology Development Program of the National Research Foundation (NRF) funded by the Korea government (MSIT) [No. 2021M3A9E4080782], and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [No. 2021-0-01343, No. 2022-0-00959].

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In NIPS. (2014)

[2] Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein Generative Adversarial Networks. In ICML. (2017)

[3] Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral Normalization for Generative Adversarial Networks. In ICLR. (2018)

[4] Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-Attention Generative Adversarial Networks. In ICML. (2019)

[5] Brock, A., Donahue, J., Simonyan, K.: Large Scale GAN Training for High Fidelity Natural Image Synthesis. In ICLR. (2019)

[6] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-Image Translation with Conditional Adversarial Networks. In CVPR. (2017)

[7] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In ICCV. (2017)

[8] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. TPAMI (2018)

[9] Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic Image Synthesis with Spatially-Adaptive Normalization. In CVPR. (2019)

[10] Vondrick, C., Pirsiavash, H., Torralba, A.: Generating Videos with Scene Dynamics. In NIPS. (2016)

[11] Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: MoCoGAN: Decomposing Motion and Content for Video Generation. In CVPR. (2018)

[12] Xu, T., Wenliang, L.K., Munn, M., Acciaio, B.: COT-GAN: Generating Sequential Data via Causal Optimal Transport. In NeurIPS. (2020)

[13] Denton, E.L., Zaremba, W., Bruna, J., LeCun, Y., Fergus, R.: Exploiting Linear Structure within Convolutional Networks for Efficient Evaluation. In NIPS. (2014)

[14] Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up Convolutional Neural Networks with Low Rank Expansions. In BMVC. (2014)

[15] Tai, C., Xiao, T., Zhang, Y., Wang, X., et al.: Convolutional Neural Networks with Low-Rank Regularization. In ICLR. (2016)

[16] Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: Training Deep Neural Networks with Binary Weights during Propagations. In NIPS. (2015)

[17] Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In ECCV. (2016)

[18] Zhu, C., Han, S., Mao, H., Dally, W.J.: Trained Ternary Quantization. In ICLR. (2017)

[19] Polino, A., Pascanu, R., Alistarh, D.: Model Compression via Distillation and Quantization. In ICLR. (2018)

[20] Han, S., Pool, J., Tran, J., Dally, W.: Learning both Weights and Connections for Efficient Neural Network. In NIPS. (2015)

[21] Kang, M., Han, B.: Operation-aware soft channel pruning using differentiable masks. In ICML. (2020)

[22] Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning Structured Sparsity in Deep Neural Networks. In NIPS. (2016)

[23] Li, Hao and Kadav, Asim and Durdanovic, Igor and Samet, Hanan and Graf, Hans Peter: Pruning Filters for Efficient ConvNets. In ICLR. (2017)

[24] Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning Efficient Convolutional Networks through Network Slimming. In ICCV. (2017)

[25] He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration. In CVPR. (2019)

[26] Hinton, G., Vinyals, O., Dean, J.: Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531 (2015)

[27] Zagoruyko, S., Komodakis, N.: Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In ICLR. (2017)

[28] Passalis, N., Tefas, A.: Learning Deep Representations with Probabilistic Knowledge Transfer. In ECCV. (2018)

[29] Tian, Y., Krishnan, D., Isola, P.: Contrastive Representation Distillation. In ICLR. (2019)

[30] Chen, L., Wang, D., Gan, Z., Liu, J., Henao, R., Carin, L.: Wasserstein Contrastive Representation Distillation. In CVPR. (2021)

[31] Shu, H., Wang, Y., Jia, X., Han, K., Chen, H., Xu, C., Tian, Q., Xu, C.: Co-Evolutionary Compression for Unpaired Image Translation. In ICCV. (2019)

[32] Fu, Y., Chen, W., Wang, H., Li, H., Lin, Y., Wang, Z.: AutoGAN-Distiller: Searching to Compress Generative Adversarial Networks. In ICML. (2020)

[33] Liu, Y., Shu, Z., Li, Y., Lin, Z., Perazzi, F., Kung, S.Y.: Content-Aware GAN Compression. In CVPR. (2021)

[34] Hu, H., Peng, R., Tai, Y.W., Tang, C.K.: Network trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures. arXiv preprint arXiv:1607.03250 (2016)

[35] Li, M., Lin, J., Ding, Y., Liu, Z., Zhu, J.Y., Han, S.: Gan Compression: Efficient Architectures for Interactive Conditional GANs. In CVPR. (2020)

[36] Wang, H., Gui, S., Yang, H., Liu, J., Wang, Z.: GAN Slimming: All-in-One GAN Compression by A Unified Optimization Framework. In ECCV. (2020)

[37] Li, S., Lin, M., Wang, Y., Chao, F., Mao, X., Xu, M., Wu, Y., Huang, F., Shao, L., Ji, R.: Learning Efficient GANs for Image Translation via Differentiable Masks and co-Attention Distillation. arXiv preprint arXiv:2011.08382 (2020)

[38] Jin, Q., Ren, J., Woodford, O.J., Wang, J., Yuan, G., Wang, Y., Tulyakov, S.: Teachers Do More Than Teach: Compressing Image-to-Image Models. In CVPR. (2021)

[39] Ren, Y., Wu, J., Xiao, X., Yang, J.: Online Multi-Granularity Distillation for GAN Compression. In ICCV. (2021)

[40] Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In ECCV. (2016)

[41] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In CVPR. (2018)

[42] Barber, D., Agakov, F.: The im algorithm: A Variational Approach to Information Maximization. In NIPS. (2004)

[43] Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)

[44] 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 CVPR. (2017)

[45] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative Adversarial Text to Image Synthesis. In ICML. (2016)

[46] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context Encoders: Feature Learning by Inpainting. In CVPR. (2016)

[47] Ba, J., Caruana, R.: Do Deep Nets really Need to be Deep? In NIPS. (2014)

[48] Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for Thin Deep Nets. In ICLR. (2015)

[49] Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational Information Distillation for Knowledge Transfer. In CVPR. (2019)

[50] Tian, Y., Krishnan, D., Isola, P.: Contrasive representation distillation. In ICLR. (2020)

[51] Oord, A.v.d., Li, Y., Vinyals, O.: Representation Learning with Contrastive Predictive Coding. arXiv preprint arXiv:1807.03748 (2018)

[52] Wang, Y., Xu, C., Qiu, J., Xu, C., Tao, D.: Towards evolutionary compression. In KDD. (2018)

[53] Cai, H., Gan, C., Wang, T., Zhang, Z., Han, S.: Once-for-All: Train One Network and Specialize it for Efficient Deployment. In ICLR. (2019)

[54] Cortes, C., Mohri, M., Rostamizadeh, A.: Algorithms for Learning Kernels based on Centered Alignment. JMLR (2012)

[55] Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear Total Variation based Noise Removal Algorithms. Physica D: nonlinear phenomena (1992)

[56] Li, S., Wu, J., Xiao, X., Chao, F., Mao, X., Ji, R.: Revisiting Discriminator in GAN Compression: A Generator-discriminator Cooperative Compression Scheme. In NeurIPS. (2021)

[57] Wainwright, M.J., Simoncelli, E.P.: Scale Mixtures of Gaussians and the Statistics of Natural Images. In NIPS. (1999)

[58] Vemulapalli, R., Tuzel, O., Liu, M.Y.: Deep Gaussian Conditional Random Field Network: A Model-based Deep Network for Discriminative Denoising. In CVPR. (2016)

[59] Zhang, R., Isola, P., Efros, A.A.: Colorful Image Colorization. In ECCV. (2016)

[60] Welling, M., Teh, Y.W.: Bayesian Learning via Stochastic Gradient Langevin Dynamics. In ICML. (2011)

[61] Nijkamp, E., Hill, M., Zhu, S.C., Wu, Y.N.: Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model. In NeurIPS. (2019)

[62] Hinton, G.E.: Training Products of Experts by Minimizing Contrastive Divergence. Neural computation (2002)

[63] Tieleman, T.: Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. In ICML. (2008)

[64] Du, Y., Mordatch, I.: Implicit Generation and Modeling with Energy Based Models. In NeurIPS. (2019)

[65] Gao, R., Lu, Y., Zhou, J., Zhu, S.C., Wu, Y.N.: Learning Generative Convnets via Multi-grid Modeling and Sampling. In CVPR. (2018)

[66] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and Improving the Image Quality of StyleGAN. In CVPR. (2020)

[67] Yu, A., Grauman, K.: Fine-Grained Visual Comparisons with Local Learning. In CVPR. (2014)

[68] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In CVPR. (2016)

[69] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In MICCAI. (2015)

[70] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In CVPR. (2016)

[71] Karras, T., Laine, S., Aila, T.: A Style-based Generator Architecture for Generative Adversarial Networks. In CVPR. (2019)

[72] Liu, Z., Luo, P., Wang, X., Tang, X.: Deep Learning Face Attributes in the Wild. In ICCV. (2015)

[73] 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. In NIPS. (2017)

[74] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. In CVPR. (2016)

[75] Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In CVPR. (2017)

[76] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In NeurIPS. (2019)

[77] Hu, J., Shen, L., Sun, G.: Squeeze-and-Excitation Networks. In CVPR. (2018)

[78] Zhou, Z., Zhuge, C., Guan, X., Liu, W.: Channel Distillation: Channel-Wise Attention for Knowledge Distillation. arXiv preprint arXiv:2006.01683 (2020)

[79] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: from Error Visibility to Structural Similarity. TIP (2004)

[80] Chen, H., Wang, Y., Shu, H., Wen, C., Xu, C., Shi, B., Xu, C., Xu, C.: Distilling Portable Generative Adversarial Networks for Image Translation. In AAAI. (2020)

[81] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In ICLR. (2015)

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See the abstract and Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Section A.3.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] we have read the guidelines and ensured that our paper conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We attach the code and instructions in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We did not report them.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We did not include them.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5.
   (b) Did you mention the license of the assets? [No] We did not mention the license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] No we did not introduce any new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Section 5.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] No we did not.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

This document firstly describes the objective function for knowledge distillation employed in GAN compression algorithms [35, 38, 33, 39, 56] which VEM is combined to validate its benefit and generality. Secondly, we explain the implementation details of VEM about the architecture design of energy-based models. Finally, we present the additional qualitative results of VEM compared with the state-of-the-art methods.

A.1 GAN Compression Baseline Algorithms

To validate the benefit and generality of VEM, we optimize the student generator using the algorithm-specific objective function, $\mathcal{L}_{\text{algo}}$, depending on the previous works while we train the network by maximizing the mutual information. In the previous methods, OMGD [39] and GCC [56] are online distillation algorithms while GAN-Compression [35], CAT [38], and CAGC [33] are offline ones.

A.1.1 OMGD [39]

$\mathcal{L}_{\text{algo}}$ is defined as follows:

$$\mathcal{L}_{\text{algo}} = \mathcal{L}_{\text{OMGD-KD}} + \lambda_{\text{CD}}\mathcal{L}_{\text{CD}} + \lambda_{\text{TV}}\mathcal{L}_{\text{TV}}, \tag{12}$$

where λ_{CD}, λ_{TV} are hyperparameters while $\mathcal{L}_{\text{TV}}$, $\mathcal{L}_{\text{OMGD-KD}}$, and $\mathcal{L}_{\text{CD}}$ are the total variation loss and distillation losses for intermediate and final outputs, respectively.

OMGD [39] employs the channel distillation loss [77, 78] for the intermediate feature maps, which encourages the student network to mimic the channel-wise attention maps in the intermediate layers of the teacher. To accomplish this goal, the channel distillation loss, $\mathcal{L}_{\text{CD}}$, is given by

$$\mathcal{L}_{\text{CD}} = \sum_{\ell=1}^{L} \mathbb{E}_{\boldsymbol{x}}\Big[\mathcal{L}_{\text{atten}}\big(G_\ell^t(\boldsymbol{x};\phi^t), f_\ell(G_\ell^s(\boldsymbol{x};\phi^s))\big)\Big], \tag{13}$$

where $G_\ell^t(\cdot;\phi^t)$ and $G_\ell^s(\cdot;\phi^s)$ are the intermediate feature maps of the ℓth selected layer in the teacher and student networks, and $f_\ell(\cdot)$ denotes the 1x1 convolution operator to match the dimensionality of the two feature maps. The attention loss, $\mathcal{L}_{\text{atten}}(\cdot,\cdot)$, is given by

$$\mathcal{L}_{\text{atten}}(\boldsymbol{p},\boldsymbol{q}) = \frac{1}{C}\|\text{GAP}(\boldsymbol{p}) - \text{GAP}(\boldsymbol{q})\|_2^2, \tag{14}$$

where $\text{GAP}(\cdot)$ is an average pooling function over the spatial dimension and C is the number of channels in p and q, which should be same.

For the output distillation loss, OMGD [39] defines $\mathcal{L}_{\text{OMGD-KD}}$ as

$$\mathcal{L}_{\text{OMGD-KD}} = \lambda_{\text{MSSIM}}\mathcal{L}_{\text{MSSIM}} + \lambda_{\text{PL}}\mathcal{L}_{\text{PL}}, \tag{15}$$

where $\mathcal{L}_{\text{MSSIM}}$ and $\mathcal{L}_{\text{PL}}$ denote the mean of structural similarity loss (SSIM) [79] and the perceptual loss [40], respectively, while λ_{SSIM} and λ_{PL} are hyperparameters to control the balance between the two terms. $\mathcal{L}_{\text{OMGD-KD}}$ is commonly used in the previous approaches for GAN compression [80, 32, 36, 37, 39, 56] to quantify the perceptual difference between two outputs given by the teacher and student. Specifically, the mean structure similarity loss is an average of $\mathcal{L}_{\text{SSIM}}$, the structure similarity between $\boldsymbol{t}_i$ and $\boldsymbol{s}_i$, i^{th} patches in $\boldsymbol{t}$ and $\boldsymbol{s}$, over all local windows in the images, which is denoted using constant C_1 and C_2 for numerical stability by

$$\mathcal{L}_{\text{SSIM}}(\boldsymbol{t_i}, \boldsymbol{s}_i) = \frac{(2\mu_{\boldsymbol{t}_i}\mu_{\boldsymbol{s}_i} + C_1)(2\sigma_{\boldsymbol{t}_i\boldsymbol{s}_i} + C_2)}{(\mu^2_{\boldsymbol{t}_i} + \mu^2_{\boldsymbol{s}_i} + C_1)(\sigma^2_{\boldsymbol{t}_i} + \sigma^2_{\boldsymbol{s}_i} + C_2)}. \tag{16}$$

In Eq. (16), $\mu_{\boldsymbol{x}}$ and $\sigma^2_{\boldsymbol{x}}$ is a mean and variance to measure for luminance and contrast while $\sigma_{\boldsymbol{x},\boldsymbol{y}}$ is a covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$ used for structural comparison. Therefore, the mean structure similarity loss is defined by

$$\mathcal{L}_{\text{MSSIM}} = \frac{1}{M}\sum_{j=1}^{M}\mathcal{L}_{\text{SSIM}}(\boldsymbol{t}_j, \boldsymbol{s}_j), \tag{17}$$

where M is the number of local windows in the output.

The perceptual loss penalizes the feature reconstruction loss, $\mathcal{L}_{\text{feat}}$, and the style loss, $\mathcal{L}_{\text{style}}$, which encourage the student output to mimic the content and the style (*e.g.*, color, textures, and common patterns) of the teacher output using a pre-trained 16-layer of VGGNet [81], $F(\cdot)$. $\mathcal{L}_{\text{feat}}$ is calculated as follows:

$$\mathcal{L}_{\text{feat}} = \frac{1}{N}\sum_{i=1}^{N}|F^i_j(\boldsymbol{t}) - F^i_j(\boldsymbol{s})|, \tag{18}$$

where $F^i_j(\cdot)$ and N are the ith element for the activations of the jth layer in the VGGNet and the number of the elements in the activations, respectively. Also, $\mathcal{L}_{\text{style}}$ is denoted by

$$\mathcal{L}_{\text{style}} = \|\text{Gram}\Big(F^i_j(\boldsymbol{t})\Big) - \text{Gram}\Big(F^i_j(\boldsymbol{s})\Big)\|_{1,1}, \tag{19}$$

where $\text{Gram}(X)$ and $\|Y\|_{1,1}$ are the Gram matrix of X and the ℓ_1 norm of vectorized of Y.

A.1.2 GCC [56]

$\mathcal{L}_{\text{algo}}$ is defined as follows:

$$\mathcal{L}_{\text{algo}} = \mathcal{L}_{\text{cGAN}} + \lambda_{\text{KD}}\mathcal{L}_{\text{GCC-KD}}, \tag{20}$$

where $\mathcal{L}_{\text{cGAN}}$ is the adversarial loss for generators and discriminators while $\mathcal{L}_{\text{GCC-KD}}$ is a distillation loss for generated and intermediate outputs, which is given by

$$\mathcal{L}_{\text{GCC-KD}} = \sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{x}}d(D^t_k(G^s(\boldsymbol{x};\phi^s);\psi^t), D^t_k(G^t(\boldsymbol{x};\phi^t));\psi^t) + \sum_{\ell=1}^{L}\mathbb{E}_{\boldsymbol{x}}d(f_\ell(G^s_\ell(\boldsymbol{x};\phi^s)), G^t_\ell(\boldsymbol{x};\phi^t)) \tag{21}$$

Here, $D^t_k(\cdot;\psi^t)$ are the intermediate feature maps of the kth selected layer in the teacher discriminator and $d(\cdot,\cdot)$ is a combination of the squared Frobenius norm of the two inputs and the Gram matrices of them.

A.1.3 GAN-Compression [35]

$\mathcal{L}_{\text{algo}}$ is defined as follows:

$$\mathcal{L}_{\text{algo}} = \mathcal{L}_{\text{cGAN}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}}, \tag{22}$$

where λ_{recon} and λ_{distill} are hyperparameters while $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{distill}}$ are defined as follows:

$$\mathcal{L}_{\text{recon}} = \begin{cases} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|G^s(\boldsymbol{x};\phi^s) - \boldsymbol{y}\|_{1,1} & \text{for paired cGANs,} \\ \mathbb{E}_{\boldsymbol{x}} \|G^s(\boldsymbol{x};\phi^s) - G^t(\boldsymbol{x};\phi^t)\|_{1,1} & \text{for unpaired cGANs,} \end{cases} \tag{23}$$

$$\mathcal{L}_{\text{distill}} = \sum_{\ell=1}^{L} \mathbb{E}_{\boldsymbol{x}} \|f_\ell(G^s_\ell(\boldsymbol{x};\phi^s)) - G^t_\ell(\boldsymbol{x};\phi^t)\|_F. \tag{24}$$

In the above equation, $f_\ell(\cdot)$ denotes the 1x1 convolution operator to match the dimensionality of the two feature maps, and $\|\cdot\|_F$ denotes the Frobenius norm .

A.1.4 CAT [38]

$\mathcal{L}_{\text{algo}}$ is defined as follows:

$$\mathcal{L}_{\text{algo}} = \mathcal{L}_{\text{cGAN}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{KA}}\mathcal{L}_{\text{KA}}, \tag{25}$$

where λ_{KA} is a hyperparameter and $\mathcal{L}_{\text{KA}}$ is given by

$$\mathcal{L}_{\text{KA}} = -\sum_{\ell=1}^{L} \mathbb{E}_x \frac{\|\rho(G^s_\ell(x;\phi^s))^{\text{T}}\rho(G^t_\ell(x;\phi^t))\|_{\text{F}}^2}{\|\rho(G^s_\ell(x;\phi^s))^{\text{T}}\rho(G^s_\ell(x;\phi^s))\|_{\text{F}}\|\rho(G^t_\ell(x;\phi^t))^{\text{T}}\rho(G^t_\ell(x;\phi^t))\|_{\text{F}}}. \tag{26}$$

Here, $\rho(\cdot)$ is a reshape operator, which transforms a 4D tensor ($\in \mathbb{R}^{n\times c\times h\times w}$) into a 2D matrix ($\in \mathbb{R}^{n\times chw}$).

A.1.5 CAGC [33]

$\mathcal{L}_{\text{algo}}$ is defined as follows:

$$\mathcal{L}_{\text{algo}} = \mathcal{L}_{\text{cGAN}} + \lambda_{\text{recon}}\mathcal{L}^{\text{mask}}_{\text{recon}} + \lambda_{\text{distill}}\mathcal{L}^{\text{mask}}_{\text{distill}} + \lambda_{\text{LPIPS}}\mathcal{L}^{\text{mask}}_{\text{LPIPS}}, \tag{27}$$

where λ_{recon}, λ_{distill}, and λ_{LPIPS} are hyperparameters while $\mathcal{L}^{\text{mask}}_{\text{recon}}$, $\mathcal{L}^{\text{mask}}_{\text{distill}}$, and $\mathcal{L}^{\text{mask}}_{\text{LPIPS}}$ are denoted as

$$\mathcal{L}^{\text{mask}}_{\text{recon}} = \mathbb{E}_x \|M \odot (G^s(x;\phi^s) - G^t(x;\phi^t))\|_1, \tag{28}$$

$$\mathcal{L}^{\text{mask}}_{\text{distill}} = \sum_{\ell=1}^{L} \mathbb{E}_x \|M \odot (f_\ell(G^s_\ell(x;\phi^s)) - G^t_\ell(x;\phi^t))\|_1, \tag{29}$$

$$\mathcal{L}^{\text{mask}}_{\text{LPIPS}} = \mathbb{E}_x LPIPS(M \odot G^s(x;\phi^s), M \odot G^t(x;\phi^t)). \tag{30}$$

M is a spatial mask whose element is a binary value, and represents whether the corresponding pixel is located at the content of the image. Also, $LPIPS(\cdot,\cdot)$ is a learned perceptual image patch similarity metric [41] to attempt to measure the difference between the generated images of the teacher and student

A.2 More Implementation Details

Figure 4 illustrates the architecture design of the energy-based models which are parameterized with convolutional neural networks. In the figure 4, we set C, the number of channels in intermediate feature maps, to 8 for the Horse $\rightarrow$ Zebra dataset while setting C to 32 for other datasets. “ConvBlock" consists of a convolution using a kernel size of 3 and the LeakyReLU activation function while “ResBlock" is composed of two convolutional layers with a kernel size of 3 and the activation ones together with a residual connection [70].

Following [64], we also add spectral normalization layers [3] to the weights in all convolutional and linear layers in order to alleviate the training instability about sharp changes in gradient space of energy-based models. Furthermore, we additionally minimize the value of the squared output from the energy-based models as a regularization term during training since the output is not bounded, which may cause the training instability. Without the regularization term, we empirically observe that the output becomes numerically unstable values. As a result, these techniques are helpful to improve the training stability.

Concatenate two inputs
3x3 Avg Pooling (stride=2)
ConvBlock, C
ResBlock, $2 \times C$
ResBlock, $4 \times C$
ResBlock, $4 \times C$
ResBlock, $4 \times C$
ResBlock, $4 \times C$
ResBlock, $4 \times C$
ResBlock, $4 \times C$
ReLU Activation
Global Average Pooling
Linear layer, 1

Figure 4: Architecture design of energy-based models.

Table 6: Selected values of λ_{MI} for VEM.

Model	Dataset	Method	λ_{MI}
Pix2Pix	Edges $\rightarrow$ Shoes	OMGD [39] + VEM (Ours)	0.100
	Cityscapes	OMGD [39] + VEM (Ours)	0.200
CycleGAN	Horse $\rightarrow$ Zebra	OMGD [39] + VEM (Ours)	0.100
		CAT [38] + VEM (Ours)	0.005
		GAN-Compression [35] + VEM (Ours)	0.100
	Summer $\rightarrow$ Winter	OMGD [39] + VEM (Ours)	0.100
SAGAN	CelebA	GCC [56] + VEM (Ours)	0.100
StyleGAN2	FFHQ	CAGC [33] + VEM (Ours)	0.050

A.3 Limitation

Although VEM incurs an extra training cost due to additionally running short-run MCMC steps, the proposed method quantitatively and qualitatively improves the performance when combined with existing GAN compression approaches.

A.4 Additional Qualitative Results

We additionally present more qualitative results of VEM and the state-of-the-art methods including the original model in Figure 5, 6, 7, 8, 9, 10, 11, and 12 which demonstrate the effectiveness of VEM.

B Code

We attached the source code to facilitate understanding and reproduction of our algorithm. Please check “README.md" in the supplementary material folder which contains how to run VEM.

Figure 5: Qualitative results of Pix2Pix on the Cityscapes dataset. “Output" denotes the ground-truth and “Original" represents the uncompressed generator.

Figure 6: Qualitative results of Pix2Pix on the Edges → Shoes dataset. “Output" denotes the ground-truth and “Original" represents the uncompressed generator.

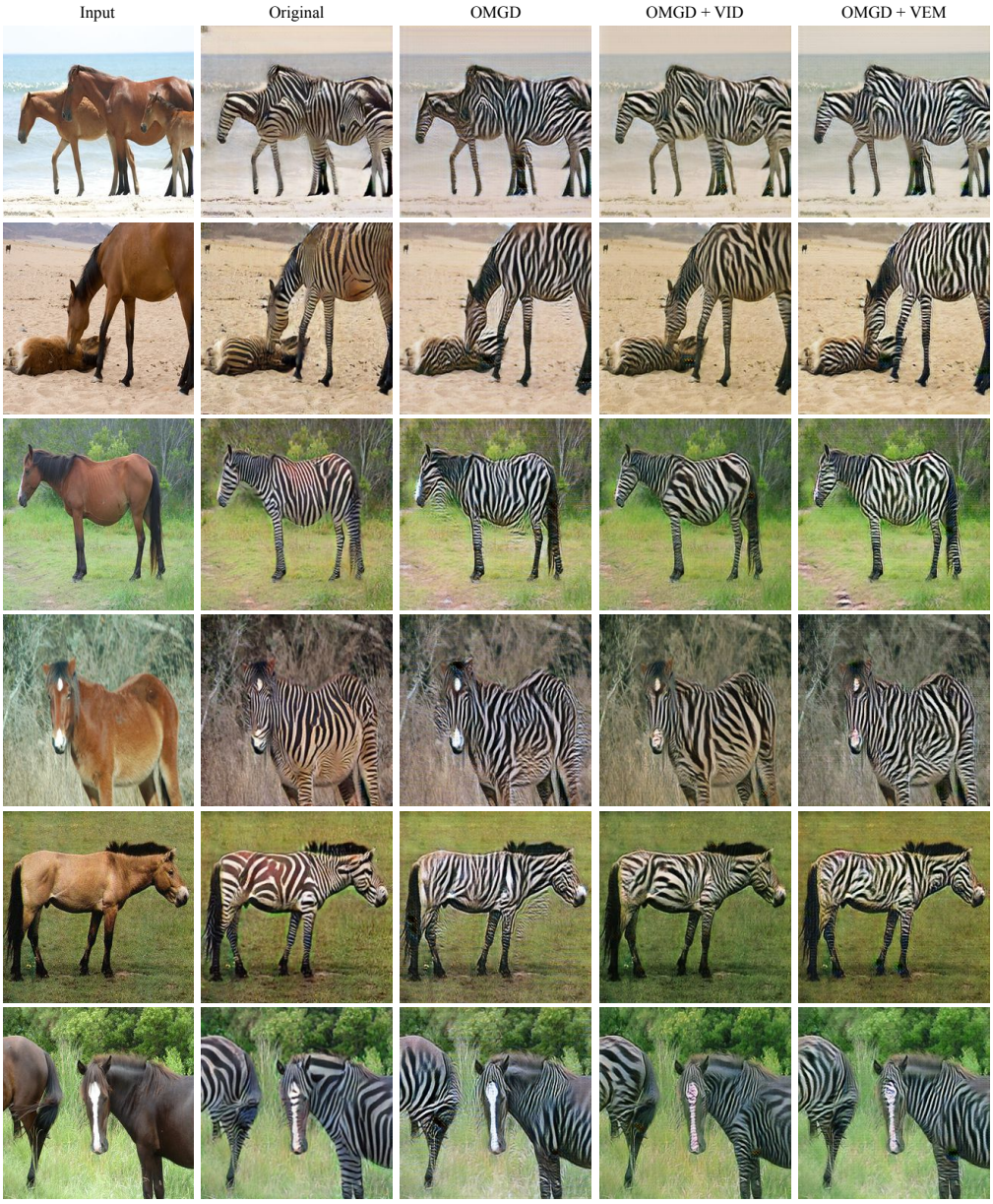

Figure 7: Qualitative results of CycleGAN on the Horse $\rightarrow$ Zebra dataset. “Original" represents the uncompressed generator.

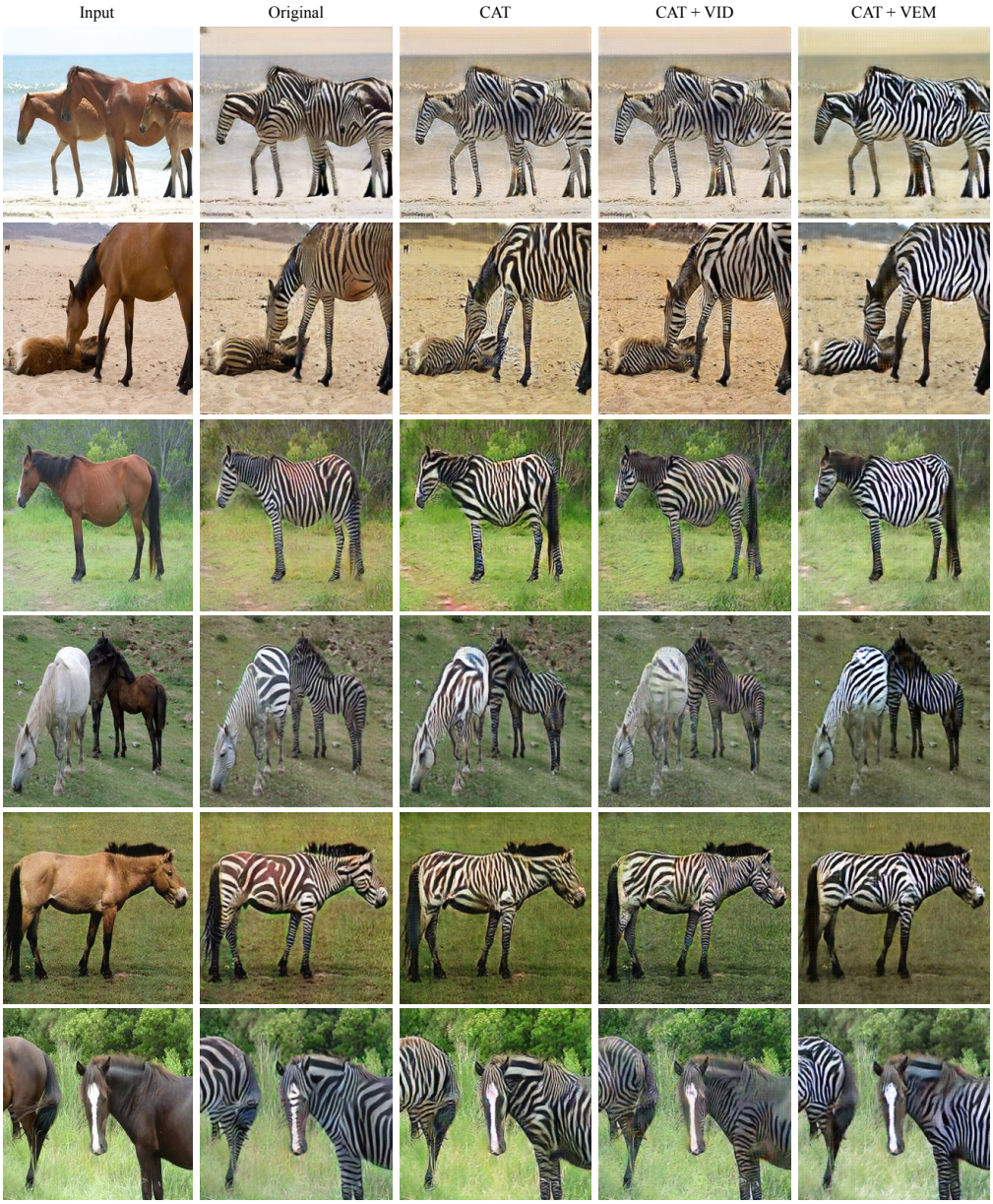

Figure 8: Qualitative results of CycleGAN on the Horse → Zebra dataset. “Original" represents the uncompressed generator.

Figure 9: Qualitative results of CycleGAN on the Horse $\rightarrow$ Zebra dataset. “Original" represents the uncompressed generator.

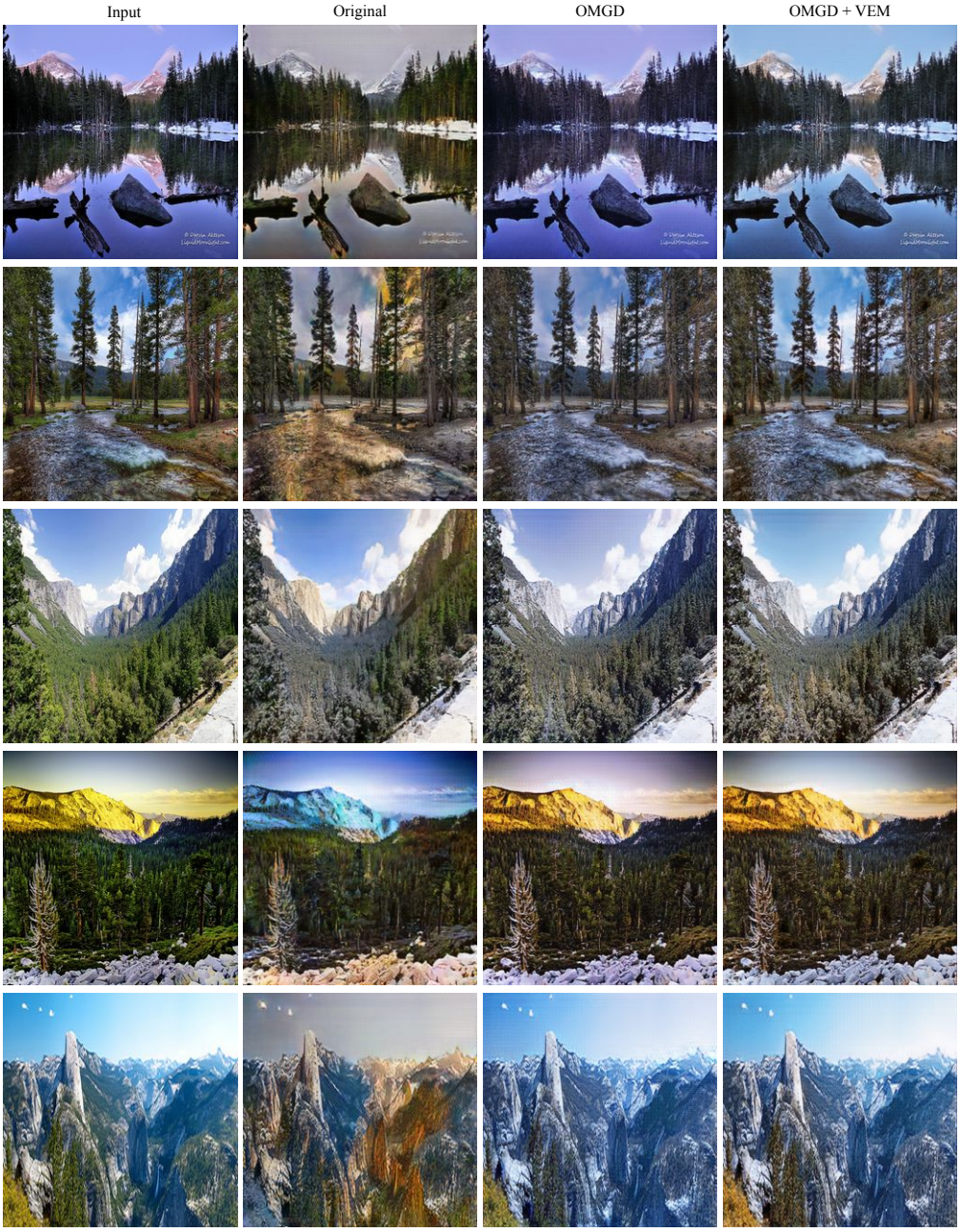

Figure 10: Qualitative results of CycleGAN on the Summer → Winter dataset. “Original" represents the uncompressed generator.

Figure 11: Qualitative results of SAGAN on the CelebA dataset. “Real Images" represents the data samples.

Figure 12: Qualitative results of StyleGAN2 on the FFHQ dataset. “Original" represents the uncompressed generator.