

# UGC: Unified GAN Compression for Efficient Image-to-Image Translation

Yuxi Ren<sup>\*</sup> Jie Wu<sup>\*†</sup> Peng Zhang Manlin Zhang  
 Xuefeng Xiao Qian He Rui Wang Min Zheng Xin Pan  
 ByteDance Inc

{renyuxi.20190622, wujie.10, zhangpeng.ucas, zhangmanlin.12, xiaoxuefeng.ailab, 1988heqian, ruiwang.rw, zhengmin.666, panxin.321}@bytedance.com

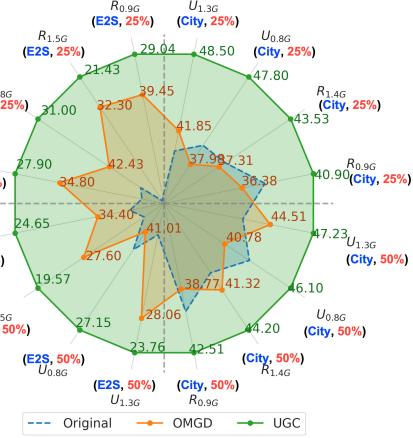
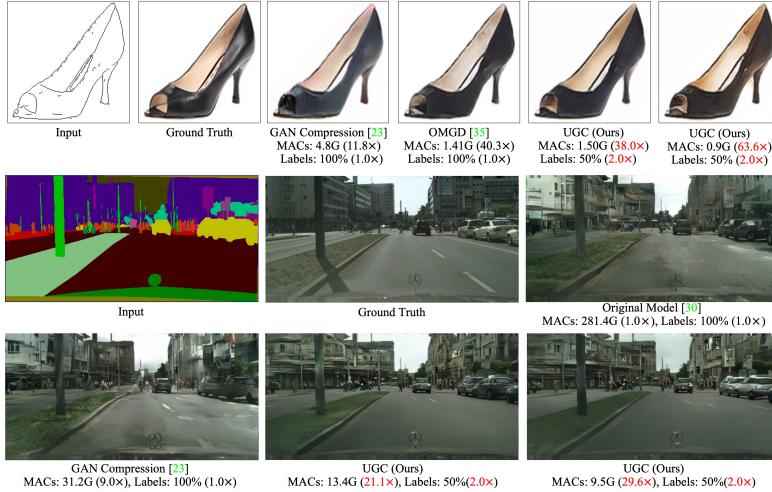


Figure 1. We introduce Unified GAN Compression (UGC) for compressing image-to-image translation-based GANs. UGC reduces **29.6-63.6×** MACs and **50%** of data labels while preserving visual fidelity. UGC establishes new state-of-the-art performance across various model constraints (12-67× MACs), label constraints (**10%, 25%, 50%**), datasets (Cityscapes (**City**)<sup>[8]</sup>, Edges2Shoes (**E2S**)<sup>[48]</sup>) and generator architectures (ResNet<sup>[14, 17]</sup>, UNet<sup>[36, 17]</sup>).  $R_{0.9G}$  denotes the Resnet-style generator with 0.9G MACs. Our code and models are made public at: <https://github.com/bytedance/UGC>

## Abstract

Recent years have witnessed the prevailing progress of Generative Adversarial Networks (GANs) in image-to-image translation. However, the success of these GAN models hinges on ponderous computational costs and labor-expensive training data. Current efficient GAN learning techniques often fall into two orthogonal aspects: i) model slimming via reduced calculation costs; ii) data/label-efficient learning with fewer training data/labels. To combine the best of both worlds, we propose a new learning paradigm, Unified GAN Compression (UGC), with a unified optimization objective to seamlessly prompt the synergy of model-efficient and label-efficient learning. UGC sets up semi-supervised-driven network architecture search and adaptive online semi-supervised distillation stages sequentially, which formulates a heterogeneous mutual learning scheme to obtain an architecture-flexible, label-efficient,

and performance-excellent model. Extensive experiments demonstrate that UGC obtains state-of-the-art lightweight models even with less than **50%** labels. UGC that compresses **40×** MACs can achieve 21.43 FID on edges→shoes with **25%** labels, which even outperforms the original model with 100% labels by 2.75 FID.

## 1. Introduction

Recently, Generative Adversarial Networks (GANs)<sup>[12, 1, 32, 33, 4, 17, 55]</sup> have achieved prominent results in various visual generative tasks, such as image-to-image translation<sup>[17, 44, 30, 55, 7]</sup> and style transfer<sup>[21, 22, 39, 9]</sup>. Albeit with varying degrees of progress, most of its recent successes rely on explosive computational complexities or extensive labeled images. Training or deploying these excellent GAN models with unwieldy resource demands is arduous, especially in the hardware-constrained<sup>[23, 35, 18]</sup> or low-data regime<sup>[52, 5, 27]</sup>. To

<sup>\*</sup>Equal contribution. <sup>†</sup>Corresponding author.

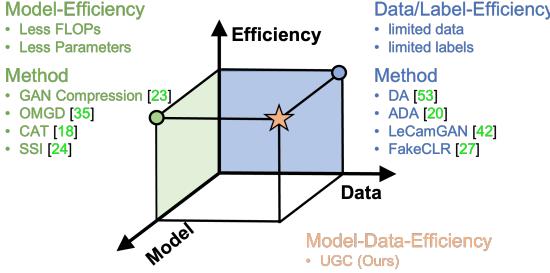


Figure 2. Unified GAN compression in the **model-data-efficiency** space, which consists of **model-efficient** learning to reduce computational costs, and **data-efficient** learning with fewer labels.

alleviate this issue, efficient GAN learning has become a newly-raised and crucial research topic in recent years. We analyze current corresponding algorithms and classify them into a unified model-data-efficiency space in Figure 2, which reveals that related techniques can be divided into two orthogonal aspects: model-efficient or data-efficient. As shown in Figure 2 left part, model-efficient approaches[23, 35, 26, 18, 51, 24] typically make efforts to learn slimming GANs to reduce ponderous computational costs and hulking memory usage. On the other hand, data-efficient learning[5, 52, 20, 27] has recently emerged as a promising approach by leveraging few-shot learning[29, 53] or regularization techniques[5, 52, 20, 42] to optimize GANs with limited training labels, as shown in Figure 2 right part.

Based on the above observation, we can find that although these two learning paradigms both reside in the common model-data-efficiency space, current image-to-image translation based GAN compression works only focus on a unilateral optimization objective. A natural question is: *whether these two orthogonal compression paradigms can be learned uniformly in a collaborative framework?* Compared with individual learning objectives, a unified solution confronts two additional challenges: i) how model and data-efficient learning cooperate in an unified GAN compression framework? ii) how to overcome the rising challenge of GAN training instability with multiple compression tools?

In many real-world scenarios for image-to-image translation tasks, the labels consistently occupy 50% of the image input, making it challenging and time-consuming to obtain large amounts of labeled data. In this work, our main focus is to study label efficiency in data-efficient GANs and provide the first attempt to answer the above questions. We introduce a novel GAN compression objective called Unified GAN Compression (UGC) that seamlessly promotes both model-efficient and label-efficient learning. The proposed UGC optimization framework comprises two stages: a semi-supervised-driven network architecture search and adaptive online semi-supervised distillation. Each stage integrates both model-based and label-centric compression

tools, formulating a heterogeneous mutual learning scheme. In the first stage, semi-supervised learning optimizes sub-networks within the search space via distillation loss on unlabeled images, leveraging the weight-sharing mechanism to facilitate the network architecture search (NAS) procedure. In the second stage, the discriminator-free student network is optimized in the online distillation setting, employing the teacher discriminator to guide the label-efficient GAN learning procedure. The contributions of this paper can be summarized in three-fold:

- **New Insight:** To the best of our knowledge, we offer the first attempt to design a unified optimization paradigm for GAN compression, which pioneers to advance image-to-image translation based GAN compression into jointly modeling both model-efficient and label-efficient GAN learning. We believe that our work can provide inspired insight and suggests a new path forward in efficient GAN learning.
- **Unified and Pioneering:** UGC seamlessly integrates three orthogonal compression techniques, i.e., network architecture search, online distillation and semi-supervised learning, to be jointly optimized in a collaborative framework. On the one hand, semi-supervised learning provides auxiliary supervision signals to facilitate NAS and the distillation procedure. On the other hand, NAS and distillation technology promote obtaining an architecture-flexible and performance-excellent model.
- **High Effectiveness:** Extensive experiments on edges→shoes [48] and cityscapes [8]) demonstrate that UGC successes to compress pix2pix [17] by **39×** MACs , pix2pixHD by **17×** and GauGAN by **31×**, and achieves state-of-the-art compression performance (see Figure 1). Under this extreme compression ratio, UGC can further reduce **50%** of labels without losing the visual fidelity of generated images. Compared with the existing competitive approaches, UGC helps to obtain better image quality with much less computational costs and labels. UGC provides a feasible solution for deploying GAN in the hardware-constrained and label-limited regime.

## 2. Related Works

### 2.1. Image-to-Image Translation GANs

Generative Adversarial Networks (GANs) [12] have achieved a remarkable leap on a series of image generation tasks, such as image-to-image translation[55, 17, 6, 7, 41, 30], image synthesis[32, 34, 28, 50, 4, 3] and image stylization[21]. Pix2Pix[17] proposed to employ paired data for image-to-image translation. Pix2PixHD[44] further accomplished the high-resolution photorealistic images

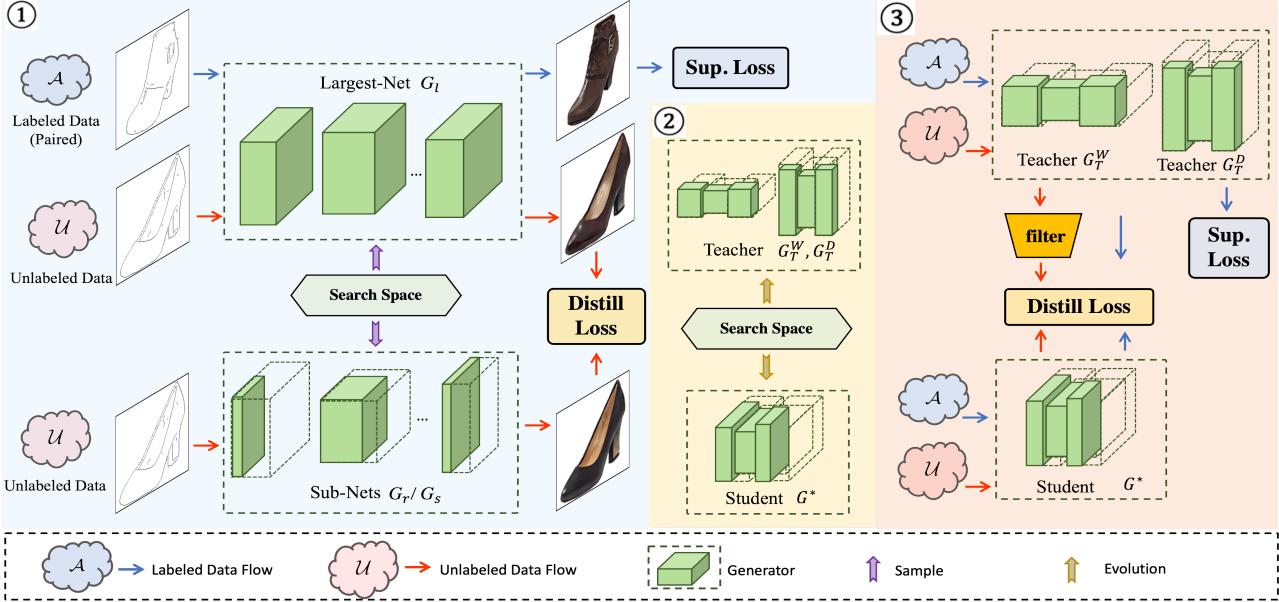


Figure 3. The whole pipeline of Unified GAN Compression (UGC) framework. UGC framework contains a two-stage optimization procedure: semi-supervised network architecture search (Step ①) and adaptive online semi-supervised distillation(Step ② + Step ③). Step ①: A largest-net  $G_l$  and sub-nets  $G_r$ ,  $G_s$  are sampled from the search space and mutually optimized via a weight-sharing mechanism. Step ②: Teacher models  $G_T^W$ ,  $G_T^D$  and student models  $G^*$  are evolutionary searched from the search space with respect to computational constraints. Step ③: Student model  $G^*$  is optimized with distillation loss via multi-teachers guidance. The blue arrow depicts the supervised training process with pair datasets  $\mathcal{A}$ , while the red counterpart denotes the semi-supervised optimization paradigm for unlabeled data  $\mathcal{U}$ .

generation by coarse-to-fine generator and multi-scale discriminator. GauGAN[31] boosted the performance of semantic image synthesis via spatially-adaptive normalization. These GAN models generate high-fidelity images at the cost of bulky computation and expensive data annotation, which restricts their application in the hardware-constrained or low-data regime. Although unpaired image translation [55, 7] tried to address the label constraints by introducing a weakly-supervised setting, they still suffer from extreme generation instability and quality degradation during model slimming or data volume shrinking.

## 2.2. Model-Efficient GAN Learning

GAN-oriented model compression has been a popular research topic due to the extensive application of GANs in real-scene. Li *et al.* [23] optimized a "once-for-all" generator and sampled the best-performed architecture according to the target computational constraints. Ren *et al.* [35] proposed an online multi-granularity distillation to optimize the lightweight models in a discriminator-free way. Li *et al.* [26] designed a generator-discriminator cooperative compression scheme to improve the adversarial training stabilization in GAN compression. Zhang *et al.* [51] selectively distill the high-frequency bands of the generated images to transfer high-quality information effectively. Li *et al.* [24] innovatively proposed a spatially sparse inference technology to achieve acceleration. Although these methods achieved a prominent balance between computational

cost and image quality, they fail to take data efficiency into consideration. When labeled data from the target domain decrease, even the state-of-the-art GAN compression method[35] still suffers from noticeable performance degradation, as shown in Figure 4.

## 2.3. Data-Efficient GAN Learning

Recently, data-efficient GANs has drawn much attention for their advantage of reducing the manual annotation effort, which mitigate GANs degradation on limited data/labels primarily by data augmentation, regularization, and pre-training techniques: 1) Data Augmentation approach [52, 20, 54] focus on relieving the discriminator from overfitting by increasing the training data diversity. For example, Zhao *et al.* [52] adopted the differentiable augmentation to stabilize the training process of the discriminator. 2) Regularization methods [47, 5, 42, 27] introduce various priors to improve the generalization of GANs. Tseng *et al.* [42] designed a regularization loss derived from LeCam-divergence to restrain the convergence of the discriminator. Li *et al.* [27] designed a contrastive learning based method to enhance performance in a limited-data regime. 3) Pre-training methods [13, 29] present transfer-learning solutions for data-efficient GANs. However, these works are not suitable for the realistic hardware-constrained scenario. To our knowledge, this paper presents the first model-label bi-dimensional efficient GAN learning paradigm.

### 3. Unified GAN Compression

Unified GAN Compression attempts to accomplish GAN-oriented tasks in the hardware-constrained and label-limited regime. Conditional GANs[17, 44, 30] aim to learn a mapping function between a source domain  $X = \{x_i\}_{i=1}^N$  and a target domain  $Y$ . In the label-efficient setting, we construct the labeled (paired) training set  $\mathcal{A} = \{x_i^{\mathcal{A}}, y_i^{\mathcal{A}}\}_{i=1}^M$  and an unlabeled datasets  $\mathcal{U} = \{x_j^{\mathcal{U}}\}_{j=1}^{N-M}$ . For the label-efficient setting, we set the labeled proportion (i.e.,  $\frac{M}{N}$ ) to 10%, 25% and 50% to accomplish the GAN-oriented tasks.

This paper offers the first attempt to integrate model-efficient and label-efficient algorithms in the GAN tasks, leveraging network architecture search[4, 43], semi-supervised learning[2, 40, 45] and distillation algorithm[10, 35, 51] to learn efficient GAN models in a cooperative setting. The whole pipeline of the UGC framework is illustrated in Figure 3, which consists of a two-stage training process. Specifically, semi-supervised-driven network architecture search and adaptive online semi-supervised distillation are introduced in section 3.1 and 3.2, respectively.

#### 3.1. Semi-Supervised Network Architecture Search

Network Architecture Search (NAS)[49, 43] is a widely-used model slimming technology for GAN-oriented compression. However, previous methods[11, 16, 23, 18] rely heavily on the labeled dataset, thus constraining the model performance in the low-annotation regime. To alleviate this problem, we introduce Semi-Supervised Learning (SSL) into NAS procedure, which can be regarded as auxiliary supervision to relieve the high dependency on labeled data. As is shown in Step ① of Figure 3, we construct a depth-width dynamic supernet as the search space  $\mathcal{S} = \{G_i\}$ ,  $G_i$  is a sub-network sampled from  $\mathcal{S}$ . We leverage sandwich training rule [49] to sample the largest sub-network  $G_l$ , smallest sub-network  $G_s$  and a random sub-networks  $G_r$  from  $\mathcal{S}$ , then formulates the following optimization function:

$$\mathcal{L}_{stage1} = \mathcal{L}_{sup}^{\mathcal{A}}(G_l) + \mathcal{L}_{dist}^{\mathcal{U}}(G_r, G_s) \quad (1)$$

The largest sub-network  $G_l$  is optimized on the labeled dataset  $\mathcal{A}$  and other attentively sampled sub-networks are supervised by distillation loss on the unlabeled dataset  $\mathcal{U}$ .

**Supervised Largest Sub-Network Training.** We follow the training scheduler in [17, 44, 30] that employs paired samples from dataset  $\mathcal{A}$  to optimize the largest sub-network  $G_l$ . Take Pix2Pix[17] as an example,  $G_l$  is trained to map  $x^{\mathcal{A}}$  to  $y^{\mathcal{A}}$  while its corresponding discriminator  $D$  is optimized to distinguish the fake images generated by  $G_l$  from the real images:

$$\begin{aligned} \mathcal{L}_{GAN}(G_l, D) = & \mathbb{E}_{x^{\mathcal{A}}, y^{\mathcal{A}}} [\log D(x^{\mathcal{A}}, y^{\mathcal{A}})] \\ & + \mathbb{E}_{x^{\mathcal{A}}} [\log (1 - D(x^{\mathcal{A}}, G_l(x^{\mathcal{A}})))] \end{aligned} \quad (2)$$

Meanwhile, a reconstruction loss is introduced to push the output of  $G_l$  to be close to the ground truth  $y^{\mathcal{A}}$ :

$$\mathcal{L}_{Recon}(G_l) = \mathbb{E}_{x^{\mathcal{A}}, y^{\mathcal{A}}} [\|y^{\mathcal{A}} - G_l(x^{\mathcal{A}})\|_1]. \quad (3)$$

To sum up, the supervised loss function  $\mathcal{L}^{\mathcal{A}}$  of  $G_l$  is:

$$\begin{aligned} \mathcal{L}_{sup}^{\mathcal{A}}(G_l) = & \arg \min_{G_l} \max_D \mathcal{L}_{GAN}(G_l, D) \\ & + \lambda_{recon} \mathcal{L}_{Recon}(G_l) \end{aligned} \quad (4)$$

**Semi-supervised Sub-Networks Training.** To reduce training dependency on target domain images, we randomly sample a middle-sized sub-network  $G_r$  and the smallest sub-networks  $G_s$ [49] from the search space  $\mathcal{S}$ . Then we conduct an online distillation[35, 26] on  $\mathcal{U}$  to capture complementary concepts from unlabeled images.

Specifically, we feed source domain images  $x^{\mathcal{U}}$  from 4.3 to  $G_l$  to generate the pseudo paired target domain images  $\hat{y} = G_l(x^{\mathcal{U}})$ . Then semi-supervised learning is conducted via online distillation loss  $\mathcal{L}_{OD}$ , where  $G_l$  guides the optimization direction of sub-networks  $G_r$  and  $G_s$  step by step. This progressive guidance technique ensures sub-networks are no longer deeply bound with the discriminator, which can train more flexibly and obtain further compression:

$$\begin{aligned} \mathcal{L}_{dist}^{\mathcal{U}}(G_r, G_s) = & \mathcal{L}_{OD}(G_l(x^{\mathcal{U}}), G_r(x^{\mathcal{U}})) \\ & + \mathcal{L}_{OD}(G_l(x^{\mathcal{U}}), G_s(x^{\mathcal{U}})) \end{aligned} \quad (5)$$

We utilize the distillation loss in OMGD [35] as our  $\mathcal{L}_{OD}$ , which is composed by Structural Similarity (SSIM) Loss[46], Perceptual Losses[19] and Total Variation Loss[37].

In short,  $G_l$  is trained on  $\mathcal{A}$  and supervises sub-networks, while sub-networks absorb knowledge from  $\mathcal{U}$  and further promote  $G_l$  via a weight-sharing mechanism. This mutual assistant strategy between sub-networks and the largest net promotes the whole super-network in a progressive collaborative way, which *overcomes the first challenge* that facilitates model-data mutual learning in a unified framework.

#### 3.2. Adaptive Online Semi-Supervised Distillation

As summarized in [23, 18], the fine-tuning procedure helps to further boost the target generation capability. So we follow [35] to conduct an online multi-teachers distillation to facilitate lightweight GAN learning. We evolutionary search student and teacher models from  $\mathcal{S}$  in the first stage and conduct an adaptive online multi-teachers distillation fine-tuning scheme to achieve better performance in the second stage. Online distillation formulates the optimization of the student model in the discriminator-free setting, where the teacher model guides the student model progressively and steadily. The adaptive semi-supervised online KD loss consists of a supervised part and a semi-supervised

counterpart:

$$\mathcal{L}_{stage2} = \mathcal{L}_{sup}^A(G_T^D) + \mathcal{L}_{sup}^A(G_T^W) + \mathcal{L}_{dist}^A(G^*) + \mathcal{L}_{dist}^U(G^*) \quad (6)$$

Where  $G^*$  is the student model and  $G_T^W, G_T^D$  are two architectural-complementary teacher models.

**Evolutionary Architecture Search.** As is shown in Step ② of Figure 3, Given a trained super-net with  $\mathcal{S}$  and a target model constraint, we utilize the evolutionary search algorithm to obtain the best-performed sub-network  $G^*$ . Additionally, we evolutionary search the two architectural-complementary teacher models that have about 20 times more computational costs than  $G^*$  from  $\mathcal{S}$ . Abbreviated as a deeper network  $G_T^D$  and a wider network  $G_T^W$ . We inherit weights from the super-net to initialize  $G_T^D, G_T^W$  and  $G^*$ , which avoids optimizing the target model from scratch and accelerates network convergence speed. Furthermore, the broad setting of the super-net allows us to flexibly obtain student and teacher models with diverse constraints, eliminating extra work of manually redesigning and training teacher models.

**Supervised Student Model Training.** Specifically,  $G_T^D$  and  $G_T^W$  are directly optimized as Eq. 4, and the corresponding teacher discriminators are inherited from the first stage in section 3.1. For the student network  $G$ , the supervised objective can be formulated as :

$$\begin{aligned} \mathcal{L}_{dist}^A(G^*) &= \mathcal{L}_{OD}(G_T^D(x^A), G^*(x^A)) \\ &\quad + \mathcal{L}_{OD}(G_T^W(x^A), G^*(x^A)) \end{aligned} \quad (7)$$

**Semi-Supervised Student Model Training.** Obviously, Eq.7 does not involve the target domain label, so  $x^U$  can also participate in this optimization process to further enhance model capacity. However, the generation ability of teacher networks for unlabeled data can fluctuant since they neither can access the paired ground truth nor has powerful generalization ability as  $G_l$  in 3.1. Therefore, we design a discriminators-guided adaptive filtering scheme to improve the pseudo-label quality in the distillation procedure:

$$\Phi(G_T^D) = \begin{cases} RandInt([0, 1]), & D[G_T^D(x^U)] > D_{EMA} \\ 1, & D[G_T^D(x^U)] < D_{EMA} \end{cases} \quad (8)$$

where  $D_{EMA}$  is the exponential moving average (EMA) prediction score of the discriminator for generated fake images and  $RandInt([0, 1])$  means to choose from 0 and 1 randomly. The semi-supervised objective is calculated with the guidance of an adaptive filter:

$$\begin{aligned} \mathcal{L}_{dist}^U(G^*) &= \Phi(G_T^D) \times \mathcal{L}_{OD}(G_T^D(x^U), G^*(x^U)) \\ &\quad + \Phi(G_T^W) \times \mathcal{L}_{OD}(G_T^W(x^U), G^*(x^U)). \end{aligned} \quad (9)$$

The experimental results reveal that this distillation fine-tuning scheme attempts to *address the second challenge* and improves the stability of GAN training under multiple compression technologies. We illustrate the complete procedure of Unified GAN Compression in Algorithm 1.

---

### Algorithm 1 Unified GAN Compression Scheme.

---

```

1: Input: labeled dataset  $\mathcal{A}$ , unlabeled dataset  $\mathcal{U}$ , search space  $\mathcal{S} = \{G_i\}$ , discriminator  $D$ , update interval  $n$ .
2: Output: trained student generator  $G^*$ 
3: Stage 1: Semi-Sup Network Architecture Search
4: Initialize search space  $\mathcal{S}$ 
5: for  $steps = 1, \dots, T$  do
6:   Sample  $G_l, G_w, G_s$  from  $\mathcal{S}$ 
7:   Sample a mini-batch data:  $(x^A, y^A) \in \mathcal{A}, x^U \in \mathcal{U}$ 
8:   Calculate  $\mathcal{L}_{sup}^A(G_l, D)$  by Eq.4.
9:   Calculate  $\mathcal{L}_{dist}^U(G_w, G_s)$  by Eq.5
10:  Update parameters of  $G_l, G_s, G_w, D$ .
11: end for
12: Stage 2: Adaptive Online Semi-Sup Distillation
13: Evolutionary Architecture Search: Search  $G^*, G_T^W, G_T^D$  from  $\mathcal{S}$ 
14: Initialize  $G^*, G_T^W, G_T^D, D$  from Stage 1.
15: for  $steps = 1, \dots, T$  do
16:   Sample a mini-batch data:  $(x^A, y^A) \in \mathcal{A}, x^U \in \mathcal{U}$ 
17:   if  $steps \% n = 0$  then
18:     Calculate  $\mathcal{L}_{sup}^A(G_T^W, D)$  by Eq.4.
19:     Calculate  $\mathcal{L}_{sup}^A(G_T^D, D)$  by Eq.4.
20:     Update parameters of  $G_T^W, G_T^D, D$ .
21:   end if
22:   Calculate  $\mathcal{L}_{dist}^A(G^*)$  by Eq.7
23:   Calculate  $\mathcal{L}_{dist}^U(G^*)$  by Eq.9
24:   Update parameters of  $G^*$ 
25: end for

```

---

## 4. Experiments

### 4.1. Experimental Settings

We follow models, datasets and evaluation metrics used in previous works [10, 23, 25, 38] for a fair comparison.

**Models.** Pix2Pix [17] is the widely-used model in GAN-oriented model compression [23, 35, 26, 18, 51]. We follow [23, 35] that adopts the UNet[36] and the ResNet style[14] generator to compress the Pix2Pix model. Pix2PixHD [44] is an image-to-image translation approach advanced in Pix2Pix for higher-resolution images, where the ResNet-style generator with more parameters is adopted. GauGAN [31] is a state-of-the-art semantic image synthesis algorithm that employs spatially adaptive (DE)normalization.

**Datasets and Evaluation Metrics.** We conduct the experiments on cityscapes [8] and edges→shoes [48] dataset. In the label-efficient training procedure, we randomly sample 10%, 25%, 50% data from the training set as the labeled data and treat others as the unlabeled counterpart. We follow [23, 35] to measure mIoU(mean Intersection over Union) on cityscapes and FID[15](Frechet Inception Distance) on edges→shoes to evaluate the quality of the generated image. Higher mIoU implies the generated images are more realistic. FID compares the distribution of generated images with real images, and a smaller FID means the performance is more convincing. We use MACs and Parameters to measure the computational costs of the model.

**Implementation details.** We set the initial learning rate as 0.0002 and linearly decay to zero in all experi-

Table 1. The performance comparison with state-of-the-art methods on Cityscapes Dataset.

Model	Arch.	Method	<i>Model Constraints</i>		<i>Label Constraints</i> @ mIoU ( $\uparrow$ )			
			MACs	#Params	10%	25%	50%	100%
Pix2Pix [17]	ResNet	Original	56.8G	11.3M	34.26	37.83	40.09	44.32
		GAN Com. [23]	4G~5G (13 $\times$ )	0.7M (16 $\times$ )	36.27	38.69	41.43	40.77
		OMGD [35]	0.87G (65 $\times$ )	0.08M (141 $\times$ )	32.90	36.38	38.77	42.15
		UGC	0.8G~0.9G (67 $\times$ )	0.10M (113 $\times$ )	<b>37.78 (+3.52)</b>	<b>40.90 (+3.07)</b>	<b>42.51 (+2.42)</b>	—
		OMGD [35]	1.41G (40 $\times$ )	0.14M (81 $\times$ )	35.24	37.31	41.32	45.21
	UNet	Original	1.4G~1.5G (39 $\times$ )	0.15M (75 $\times$ )	<b>39.90 (+5.64)</b>	<b>43.53 (+5.70)</b>	<b>44.20 (+4.11)</b>	—
		OMGD [35]	18.6G	54.5M	33.36	40.10	42.78	42.71
		UGC	0.71G (26 $\times$ )	1.9M (29 $\times$ )	34.99	37.98	40.78	45.52
		OMGD [35]	0.8G~0.9G (22 $\times$ )	1.5M (36 $\times$ )	<b>44.83 (+11.47)</b>	<b>47.80 (+7.70)</b>	<b>46.10 (+3.32)</b>	—
		UGC	1.22G (15 $\times$ )	3.4M (16 $\times$ )	35.47	41.85	44.51	48.91
		OMGD [35]	1.2G~1.3G (15 $\times$ )	2.0M (27 $\times$ )	<b>44.46 (+11.10)</b>	<b>48.50 (+8.40)</b>	<b>47.23 (+4.45)</b>	—
Pix2PixHD [44]	ResNet	Original	151.3G	182.6M	45.66	49.66	55.50	58.30
		UGC	8G ~ 10G (17 $\times$ )	5.0M (37 $\times$ )	<b>46.34 (+0.68)</b>	<b>52.40 (+2.74)</b>	<b>56.65 (+1.15)</b>	—
		UGC	12G ~ 14G (12 $\times$ )	7.0M (26 $\times$ )	<b>47.22 (+1.56)</b>	<b>54.42 (+4.76)</b>	<b>57.28 (+1.78)</b>	—
GauGAN [30]	ResNet	Original	281.4G	93.1M	46.30	55.54	58.24	62.30
		GAN Com. [23]	30G ~ 32G (9 $\times$ )	20M (5 $\times$ )	48.93	55.21	58.19	61.22
		UGC	8G ~ 10G (31 $\times$ )	3.0M (31 $\times$ )	<b>54.85 (+8.55)</b>	<b>60.94 (+5.40)</b>	<b>62.47 (+4.23)</b>	—
		UGC	12G ~ 14G (22 $\times$ )	5.0M (19 $\times$ )	<b>56.50 (+10.20)</b>	<b>61.53 (+5.99)</b>	<b>62.73 (+4.49)</b>	—

Table 2. The performance comparison with state-of-the-art methods in Pix2Pix model on Edges→Shoes Dataset.

Arch.	Method	<i>Model Constraints</i>		<i>Label Constraints</i> @ FID ( $\downarrow$ )			
		MACs	#Params	10%	25%	50%	100%
ResNet	Original [17]	56.8G	11.3M	89.78	65.80	38.18	24.18
	GAN Com. [23]	4G~5G (13 $\times$ )	0.7M (16 $\times$ )	59.13	34.82	29.78	26.60
	OMGD [35]	0.87G (65 $\times$ )	0.08M (141 $\times$ )	44.96	39.45	34.40	34.48
	UGC	0.8G~0.9G (67 $\times$ )	0.10M (113 $\times$ )	<b>33.00 (-56.78)</b>	<b>29.04 (-36.76)</b>	<b>24.65 (-13.53)</b>	—
	OMGD [35]	1.41G (40 $\times$ )	0.14M (81 $\times$ )	42.56	32.30	27.60	25.88
	UGC	1.4G~1.5G (39 $\times$ )	0.15M (75 $\times$ )	<b>25.30 (-64.48)</b>	<b>21.43 (-44.37)</b>	<b>19.57 (-18.61)</b>	—
UNet	Original [17]	18.6G	54.5M	86.42	47.46	38.51	34.31
	OMGD [35]	0.71G (26 $\times$ )	1.9M (29 $\times$ )	87.20	42.43	41.01	32.30
	UGC	0.8G~0.9G (22 $\times$ )	1.5M (36 $\times$ )	<b>37.80 (-48.62)</b>	<b>31.00 (-16.46)</b>	<b>27.15 (-11.36)</b>	—
	OMGD [35]	1.22G (15 $\times$ )	3.4M (16 $\times$ )	43.12	34.80	28.06	25.00
	UGC	1.2G~1.3G (15 $\times$ )	2.0M (27 $\times$ )	<b>34.80 (-51.62)</b>	<b>27.90 (-19.56)</b>	<b>23.25 (-15.26)</b>	—

ments.  $\lambda_{SSIM}$ ,  $\lambda_{feature}$ ,  $\lambda_{style}$ ,  $\lambda_{TV}$  in  $\mathcal{L}_{OD}$  are fixed as 1e1, 1e4, 1e1, 1e-5 respectively. To construct the search space  $\mathcal{S}$ , we first insert 3 resnet blocks after every upsample/downsample layer into the original model. The sub-networks sampling procedure dynamically selects layer channel number  $c=[8, 16, \dots, 64]$ , step=8; and corresponding resnet blocks  $d=[0, 1, 2]$ .

## 4.2. Comparison with state-of-the-art methods

In this section, we compare UGC with several state-of-the-art GAN compression approaches and the original models under different label constraints. We reimplemented all competitors under their official codebase and maintained their training settings to conduct the experiments under

10%, 25%, and 50% partition protocols. As shown in Table 1, Table 2 and Figure 4, we report the comparison results on edges→shoes [48] and cityscapes [8] dataset.

### 4.2.1 Label Efficiency

From Table 1 and 2, we have the following observations:

**Overall Performance.** UGC shows consistent significant advantages under 10%, 25% and 50% label constraints, which reveals that UGC help to capture additional data characteristic to facilitate model compression. Compared with a similar MACs model (1.4G~1.5G) from OMGD, UGC decreases the FID by **10.84** on edges→shoes with **25%** labels. And this result even outperforms the original model with **full** dataset by a large margin (24.18 vs. 21.43).

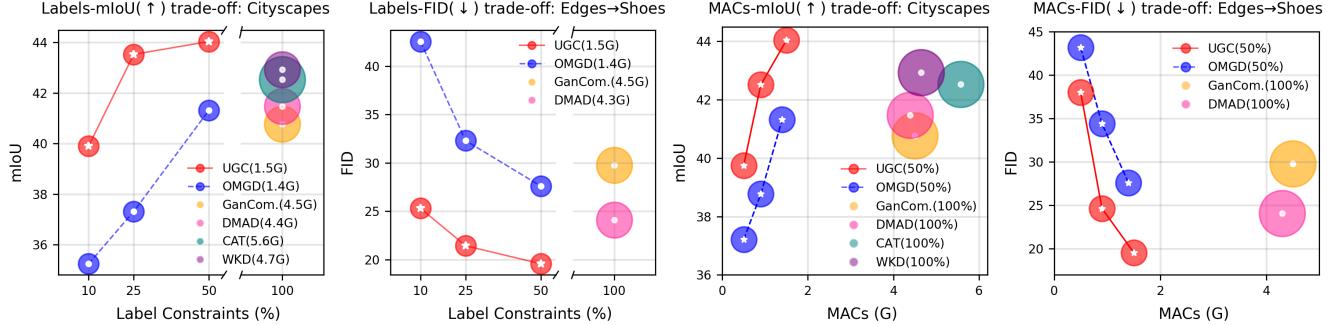


Figure 4. MACs-/Labels-Performance trade-off between UGC and state-of-the-art approaches including GAN compression[23], DMAD[25], CAT[18], OMGD[35] and WKD[51]. The first two figures depict the trade-off in different label constraints. A larger circle denotes that the model has greater MACs. The last two figures illustrate the results under different model constraints, where the circle size represents the usage of the labels. UGC significantly outperforms these methods with much less computational cost and labeled data.

**Model Type.** UGC generalizes well on different conditional paired image-to-image GANs. The UGC-based Pix2PixHD model outperforms the original model measures by mIoU, which increases from 55.50 to 57.28 with 50% labels. GauGAN also shows promising results, UGC obtains a **10.9%** mIoU improvement on cityscapes under 25% labels. UGC can surpass the original model trained on 100% dataset in 50% label protocols.

**Generator Architecture.** UGC has strong robustness for diverse generator architecture. On cityscapes, UGC with ResNet style generator (dubbed as UGC(R)) improves the mIoU by **13.2%, 16.7%, 7%** on 10%, 25% and 50% labeled settings respectively. UGC with UNet style generator(dubbed as UGC(U)) with **50%** labels achieves better performance than OMGD(U) with **100%** labeled protocols, declining FID from 25.00 to 23.25 on edges→shoes.

#### 4.2.2 Model Efficiency

Essentially, UGC manages to obtain lightweight models in a label-efficient way. We compare the performance from the model-constraint perspective in this section. Compared to the original model in Pix2Pix, UGC(R) is compressed about **39×** of MACs and **75×** of parameters, UGC(U) is compressed by **15×** of MACs and **27×** of parameters. Nevertheless, even under this extreme compression ratio, UGC(R) and UGC(U) surpass the original model by a large margin. UGC achieves better performance than OMGD on Pix2Pix models with similar compression ratios. Compared to OMGD, UGC(R) and UGC(U) decrease the FID for **24.4%** and **7%** respectively on edges→shoes using 50% labels. On GauGAN and Pix2PixHD, UGC achieves a similar mIoU with the original model while reducing the computation by **23×** and half of the labels. Conclusively, UGC coordinates model compression well with data compression, achieving state-of-the-art light-weighted models even with less than 50% labels.

Table 3. Ablation studies on Semi-Supervised Learning (SSL).

Dataset	Method	Label Constraints	
		10%	25%
Cityscapes	Ours w/o NAS-SSL	44.15	45.93
	Ours w/o Finetune-SSL	43.52	47.07
	Ours w/o ALL-SSL	43.51	45.77
	Ours	44.46	48.5
Edges→Shoes	Ours w/o NAS-SSL	38.16	34.42
	Ours w/o Finetune-SSL	37.30	30.1
	Ours w/o ALL-SSL	40.57	35.37
	Ours	34.80	27.90

Table 4. Ablation studies on NAS.

Dataset	MACs	Method	Label Constraints	
			10%	25%
Cityscapes	1.2G	Ours w/o NAS	42.04	44.64
		Ours	44.46	48.50
	0.8G	Ours w/o NAS	38.57	42.88
		Ours	44.83	47.80
Edges→Shoes	1.2G	Ours w/o NAS	44.4	34.77
		Ours	34.80	27.90
	0.8G	Ours w/o NAS	66.30	41.10
		Ours	37.80	31.00

#### 4.3 Ablation Study

In this section, we conduct ablation studies to investigate the effectiveness of each essential component of UGC. Experiments are carried out on the UNet style generator of the Pix2Pix model on Cityscapes and edges→shoes.

**Analysis of SSL in NAS.** Firstly, we construct a variant (abbreviated as "Ours w/o NAS-SSL") that trains the super-network in section 3.1 without unlabeled dataset  $\mathcal{U}$ . Experiments in Table 3 indicate that the absence of unlabeled images leads to a noticeable drop in performance. For example, "Ours w/o NAS-SSL" increases the FID from 34.8 to 38.16 compared to our approach on the 10% labeled edges→shoes dataset.

**Analysis of SSL in Finetune.** Secondly, we design a variant "Ours w/o Finetune-SSL" that removes the unlabeled data usage during the online semi-supervised distil-

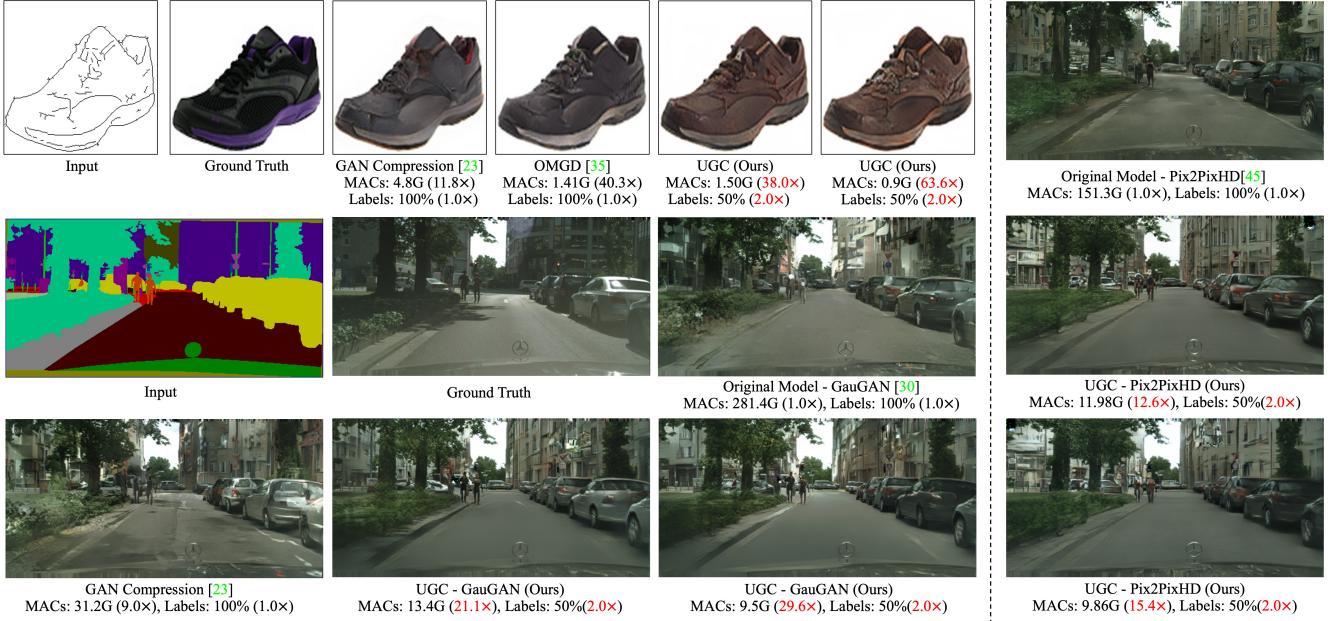


Figure 5. Qualitative Results. UGC reduces MACs by **38 $\times$**  and **29.6 $\times$**  for Pix2Pix and GauGAN with 50% labels, respectively.

Table 5. Latency Speedup on Mobile Phones.

Model	Method	MACs(G)	Mi 10 Latency(ms)	
			CPU	GPU
Pix2Pix	Original[17]	56.8	601	217.8
	GAN Com.[23]	5.42(10.5 $\times$ )	177.6(3.4 $\times$ )	52.3(4.2 $\times$ )
	UGC	1.50(38.0 $\times$ )	92.3(6.5 $\times$ )	34.1(6.4 $\times$ )
GauGAN	Original[31]	281.4	2366.7	1252.5
	GAN Com.[23]	31.2(9.0 $\times$ )	549.1(4.3 $\times$ )	227.7(5.5 $\times$ )
	UGC	13.6(21.1 $\times$ )	332.5(7.1 $\times$ )	155.8(8.0 $\times$ )
		<b>9.5(29.6<math>\times</math>)</b>	<b>275.6(8.6<math>\times</math>)</b>	<b>137.8(9.1<math>\times</math>)</b>

lation procedure in section 3.2. As summarized in Table 3, without SSL in the finetuning process, the generator declines 2.9% and 7.9% under the 25% labeled dataset setting on cityscapes and edges $\rightarrow$ shoes, respectively. The variant “Ours w/o ALL-SSL” without  $\mathcal{U}$  in both NAS and finetune process performs worst on all datasets.

**Analysis of NAS.** To delve deep into the significance of NAS process, we additionally train a fixed-architecture model with similar MACs on  $\mathcal{A}$  and  $\mathcal{U}$ . As shown in Table 4, NAS-based UGC obtains 8.6% and 19.8% performance improvement under 25% label setting in 1.2G MACs. The experiment results demonstrate that our proposed NAS technique effectively boosts the model performance in GANs.

#### 4.4. Hardware Acceleration

We report the inference latency results on the GPU and CPU of Mi 10 using TNN toolkits. As shown in Table 5, UGC obtains significant inference acceleration on mobile phones. For Pix2Pix, the generator compressed by UGC surpasses **40FPS** inference speed, achieving the demand for interactive applications on mobile phones. UGC attempts to reduce latency from 2366.7ms to 275.6ms for GauGAN, with a nearly **90%** latency decline and outperforms [23] by a large margin. It demonstrates that UGC provides a feasible solution for real-time image translation on edge devices.

#### 4.5. Qualitative Results

We depict the visualization results of UGC and the state-of-the-art GAN compression methods in Figure 5. Pix2Pix compressed by UGC shows advantageous texture and glossy reconstruction ability on edges $\rightarrow$ shoes compared to other methods. UGC on GauGAN and Pix2PixHD are excellent at synthesizing the complex details of buildings and vehicles, achieving the best generation of natural street scenes.

### 5. Conclusion

In this paper, we propose a new learning paradigm, Unified GAN Compression (UGC), with a unified optimization objective to seamlessly prompt the synergy of model-efficient and label-efficient learning. UGC sets up semi-supervised-driven network architecture search and

TNN: <https://github.com/Tencent/TNN>

adaptive online semi-supervised distillation stages sequentially, which formulates a heterogeneous mutual learning scheme to obtain an architecture-flexible, label-efficient, and performance-excellent model. Extensive experiments demonstrate that UGC can obtain state-of-the-art lightweight models even with less than **50%** labels. UGC provides a feasible solution for deploying the GAN model in the hardware-constrained and low-annotation regime.

## References

- [1] Martín Arjovsky, Soumith Chintala, and L. Bottou. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017.
- [2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixermatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.
- [3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.
- [4] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019.
- [5] Jie Cao, Luanxuan Hou, Ming-Hsuan Yang, Ran He, and Zhenan Sun. Remix: Towards image-to-image translation with limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15018–15027, 2021.
- [6] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, M. Enzweiler, Rodrigo Benenson, Uwe Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.
- [9] Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11389–11398, 2022.
- [10] Y. Fu, Wuyang Chen, Haotao Wang, Haoran Li, Yingyan Lin, and Zhangyang Wang. Autogan-distiller: Searching to compress generative adversarial networks. In *ICML*, 2020.
- [11] X. Gong, S. Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3223–3233, 2019.
- [12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
- [13] Timofey Grigoryev, Andrey Voynov, and Artem Babenko. When, why, and which pretrained gans are useful? *arXiv preprint arXiv:2202.08937*, 2022.
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.
- [16] L. Hou, Ze-Huan Yuan, Lei Huang, Hua-Wei Shen, Xueqi Cheng, and Changhu Wang. Slimmable generative adversarial networks. *ArXiv*, abs/2012.05660, 2020.
- [17] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.
- [18] Qing Jin, Jian Ren, Oliver J. Woodford, and Jiazhao Wang. Teachers do more than teach: Compressing image-to-image models. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [19] J. Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.
- [20] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.
- [21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.
- [23] Muyang Li, J. Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han. Gan compression: Efficient architectures for interactive conditional gans. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5283–5293, 2020.
- [24] Muyang Li, Ji Lin, Chenlin Meng, Stefano Ermon, Song Han, and Jun-Yan Zhu. Efficient spatially sparse inference for conditional gans and diffusion models. *arXiv preprint arXiv:2211.02048*, 2022.
- [25] Shaojie Li, Mingbao Lin, Yan Wang, M. Xu, Feiyue Huang, Yongjian Wu, Ling Shao, and Rongrong Ji. Learning efficient gans using differentiable masks and co-attention distillation. *ArXiv*, abs/2011.08382, 2020.
- [26] Shaojie Li, Jie Wu, Xuefeng Xiao, Fei Chao, Xudong Mao, and Rongrong Ji. Revisiting discriminator in gan com-

- pression: A generator-discriminator cooperative compression scheme. *Advances in Neural Information Processing Systems*, 34:28560–28572, 2021.
- [27] Ziqiang Li, Chaoyue Wang, Heliang Zheng, Jing Zhang, and Bin Li. Fakelcr: Exploring contrastive learning for solving latent discontinuity in data-efficient gans. In *European Conference on Computer Vision*, pages 598–615. Springer, 2022.
- [28] Takeru Miyato, T. Kataoka, Masanori Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *ArXiv*, abs/1802.05957, 2018.
- [29] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2021.
- [30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [32] A. Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.
- [33] A. Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.
- [34] Scott E. Reed, Zeynep Akata, Xinchen Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.
- [35] Yuxi Ren, Jie Wu, Xuefeng Xiao, and Jianchao Yang. Online multi-granularity distillation for gan compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6793–6803, 2021.
- [36] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.
- [37] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992.
- [38] H. Shu, Yunhe Wang, Xu Jia, Kai Han, H. Chen, Chunjing Xu, Q. Tian, and Chang Xu. Co-evolutionary compression for unpaired image translation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3234–3243, 2019.
- [39] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2, 2021.
- [40] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.
- [41] Hao Tang, Dan Xu, Nicu Sebe, Yanzhi Wang, Jason J. Corso, and Yan Yan. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [42] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7921–7931, 2021.
- [43] Dilin Wang, Meng Li, Chengyue Gong, and Vikas Chandra. Attentivenas: Improving neural architecture search via attentive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6427, 2021.
- [44] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [45] Yaxing Wang, Salman Khan, Abel Gonzalez-Garcia, Joost van de Weijer, and Fahad Shahbaz Khan. Semi-supervised learning for few-shot image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4453–4462, 2020.
- [46] Zhou Wang, A. Bovik, H. R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.
- [47] Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. *Advances in Neural Information Processing Systems*, 34:9378–9390, 2021.
- [48] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014.
- [49] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *European Conference on Computer Vision*, pages 702–717. Springer, 2020.
- [50] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019.
- [51] Linfeng Zhang, Xin Chen, Xiaobing Tu, Pengfei Wan, Ning Xu, and Kaisheng Ma. Wavelet knowledge distillation: Towards efficient image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12464–12474, 2022.
- [52] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33:7559–7570, 2020.
- [53] Yunqing Zhao, Henghui Ding, Houjing Huang, and Ngai-Man Cheung. A closer look at few-shot image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9140–9150, 2022.

- [54] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.
- [55] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.