Are we pruning the correct channels in image-to-image translation models

Yiyong Li[1]
haoja625@163.com
Zhun Sun†[1]
zhunsun@gmail.com
Chao Li[2]
chao.li@riken.jp

[1] Bigo Technology pte. ltd.
Singapore
[2] Tohoku University
Sendai, Miyagi, Japan
[3] Center for Advanced Intelligence
Project (AIP), RIKEN
Tokyo, Japan

Abstract

Although the demands of deploying deep models to the edge devices are proliferating, it is a challenging problem to compress image-to-image translation models based on Generative Adversarial Networks (GANs). In particular, most of the compression approaches do not apply to the GAN models. In this paper, we revisit weight pruning approaches for GANs and theoretically build a novel perturbation model to analyze the effect of pruning certain weights for the instance normalization (`IN`)-based image-to-image translation GAN models. Furthermore, we develop a new training framework by imposing perturbation-bound-induced pruning loss. In the experimental analysis, we observe that the former pruning approaches do wrongly prune the channels with high visual impacts. We then depict the effectiveness of the proposed model by conducting both on-training pruning and zero-shot pruning of current state-of-the-art models. Specifically, we compress the CycleGAN [51] model using the `horse2zebra` dataset. In the on-training pruning task, we achieve $\times$5.31 and $\times$5.44 compression ratio to the original model in terms of FLOPs and Memory consumption, respectively; In the zero-shot pruning tasks, we obtain a decrease of 4.94 in the FID score compared to the best model provided in OMGD [39], both with a negligible decrease in output visual quality.

1 Introduction

Deep learning has transitioned into an edge computing era, countless efforts have been made to actualize the real-time execution of various computer vision tasks on the mobile phone, to name a few [7, 13, 29, 32, 38, 41, 49]. Among them, one exciting application is the Generative Adversarial Network (GAN) [6] based image-to-image translation [51]. By translating source images to another domain, it accomplishes a lot of visually appealing effects, such as image synthesis [2, 48], style transfer [15, 18] or image enhancement [20, 44, 45].

However, GAN-based image-to-image translation is notorious for its computational resource intensity, limiting the deployment to edge devices. One reason is that the image-to-image translation tasks have both their sources and outputs being spatially gigantic (*e.g.*

†Corresponding author

(a) Input (b) un-pruned (c) eb=0.194 (d) eb=1.604 (e) eb=2.430

Figure 1: Examples of `horse2zebra` generation results using a pre-trained CycleGAN with specific channels pruned. (a) the input image; (b) the generated zebra image using the un-pruned model; (c)(d)(e) the generated zebra images when the channels with perturbation error bounds (eb) of $0.194, 1.604, 2.430$ pruned (See Section 5). It is conspicuous that the appearances alter more severely as the perturbation error bound of the channel grows.

images of size $256 \times 256 \times 3$), which results in dramatically increased computational complexity if the number of hidden layer channels increases. There have been various methods proposed to obtain more compact form of the generators in GANs, such as architecture search [27, 28, 52], weight pruning [14, 22, 31, 34] and low-rank factorization [21, 37]. In this work, we focus on the well-studied weight pruning approach and discuss its potential improvement for the specified GAN compression task.

Most weight pruning methods start from defining a measure to calculating the importance of weights. The measure can either be heuristic or optimization-based. For instance, the heuristic measures can be the magnitude of weights [7] or redundancy of channels (approximation error by linear combinations of others) [35]; the optimization-based measures are usually adopted to the weights or activations in a multiplicative style, *e.g.* $\gamma(\mathcal{W}^T\mathcal{X})$, then the measure γ is optimized during the training progress subjected to a regularization, such as ℓ_0 [10] or ℓ_1 (LASSO) [34]. However, in practice, a direct adaptation of these measures often suffers great performance decay compared with the original generator, as reported in a former study [40]. From our initial experimental analysis, we observe that the measures based on models designed for discriminative tasks, *e.g.* image classification, semantic segmentation, are not always suitable for the GAN models. The major obstruction is that the normalization operations, *e.g.* batch normalization [16], do not contribute much when measuring the discriminativity of the weights (or learned feature representations)[1]. On the other hand, the instance normalization [17, 42, 51] is the key feature for image-to-image translation GAN models. To prune weights of these models without a proper measure result in two consequences: first, the visual concepts perceptible to humans may be modified; thus, the outputs would look different than those from unpruned models; second, instability may happen during the training progress since the min-max GAN objective is known to be sensitive to perturbations. We will demonstrate these problems in the experimental analysis.

In this study, we first introduce a measure that is specifically proposed for the image-to-image translation GAN models equipped with the instance normalization (`IN`). Intuitively, it depicts the contribution made from the input channels to the output channels, *w.r.t.* the convolution and instance normalization parameters. We then represent a perturbation error bound of pruning certain weights using this measure. To bring them to practical usage, we propose a novel loss term during the training progress. The loss term distinguishes the impacts of weights, *i.e.* weights that have fewer impacts on the outputs now will have a lower bound, yet weights that have significant impacts on the outputs will be retained. Finally,

[1] Or assumed to be in many studies since the normalization is not thoroughly considered in the measures.

we prune weights whose bound is below a certain threshold. We present the overall framework as Bound-Induced GAN pruning (BIGP), the main contributions of this paper can be summarized as follows:

- We propose a novel theoretical perturbation model, by which we rigorously prove its error upper-bound caused by pruning channels of hidden layers. We verify this model using zero-shot (without fine-tuning) pruning and indicate its effectiveness in the practical scene.
- We build a new loss term based on the error upper-bound analysis, as well as an on-training pruning framework, which can be amalgamated with other knowledge distillation approaches.
- In the experimental analysis, we demonstrate that the proposed on-training and zero-shot pruning methods outperform the state-of-the-art results in terms of compression ratio, FID score, training stability, and similarity to the uncompressed model.

2 Related Work

Taxonomy of Model Compression. The literature focusing on obtaining tiny and efficient models can be generally classified into 5 categories: network architecture search (NAS) [5], knowledge distillation [12], low-rank factorization [37], weight pruning [7] and quantization [7]. Modern model compression systems often combine two or more approaches for best practice. For instance, it is common to combine the knowledge distillation on a pruned small network. Our work also follows this design, while the core contribution focuses on the weight pruning part.

Weight Pruning of General Proposal. We briefly summarize the weight pruning related works using the measure defined in their works. The most commonly employed measures are heuristic (manually crafted), such as the weights with small values [8], influence caused by pruning [19], magnitude of batch normalization [50], inactivity of neurons [14], entropy of activations [33], neuron importance score [47] and weight similarity/redundancy [9]. Another branch is the optimization-based measure, they are used to the weights or activations in a multiplicative style, *e.g.* $\gamma(\mathcal{W}^T\mathcal{X})$. During the optimizing progress, the γ is learned subjected to a type of structured regularization, for instance, ℓ_0 [10], ℓ_1 (LASSO) [34] and ℓ_2 [46]. After optimization, the models are then pruned according to the sparse measure γ.

Compression Methods that are Specific to GANs. There have been several notable works focusing on the compression of GAN models. A search-based channel pruning method is proposed in [40]. The core idea is to search for the best student (pruned) architecture that minimizes the reconstruction error using evolutionary algorithms (EA). Compared to the multiple training progress that happened in EA, our method only needs to train once. Yet we show that the choice of preserved channels is reasonably similar (92%+). Another work [43] combines several well-developed methods to build a joint optimization framework of knowledge distillation, pruning, and quantization. While the pruning progress is borrowed from [34], this implementation holds the best on-training pruning performance currently. Meanwhile, a series of full knowledge distillation approaches [1, 3, 24, 25, 26, 30, 39] emerge recently. These works focus on building stable and semantic-rich distillation systems for a pre-defined small network. Compared to the on-training pruning approaches, their small networks are either pruned after the training process or manually crafted. Our work riches them with a new perturbation analysis and an effective GAN pruning method.

3 Perturbation Analysis

We start with notation and review mathematical expressions of basic building blocks used in our analysis, including instance normalization (`IN`) [42] and rectified linear unit (`relu`) [36] operations for being self-contained.

Notation. In the work we denote scalars by italic letters, *e.g.*, $n, m \in \mathbb{R}$, and denote vectors and matrices by boldface letters, *e.g.*, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ and $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{W \times H}$. For higher-order tensors, we denote them by calligraphic letters, *e.g.*, $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{C \times W \times H}$. For any integer k, we use $[k]$ to denote the set of integers from 1 to k. Supposing a mth-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$, its vectorization is defined as $\mathbf{a} = vec(\mathcal{A}) \in \mathbb{R}^{\prod_{i=1}^{m} n_i}$, where its entries satisfy

$$\mathcal{A}(i_1, i_2, \ldots, i_m) = \mathbf{a}\left((i_m - 1)n_1 \cdots n_{m-1} + \cdots + (i_2 - 1)n_1 + i_1\right), \tag{1}$$

where $i_j \in [n_j]$ for all $j \in [m]$. Given a vector $\mathbf{a} = [a_0, a_1, \ldots, a_{n-1}]^\top \in \mathbb{R}^n$, its involving *circulant matrix* is represented by $\mathbf{A} = circ(\mathbf{a}) \in \mathbb{R}^{n \times n}$, where the (i, j)th entry of $\mathbf{A}$ satisfies $\mathbf{A}(i, j) = a_{(j-i) \bmod n}$ and "mod" denotes the modular operation. The operation $\|\cdot\|_{\ell_1}$ denotes the ℓ_1 norm and $|\cdot|$ denotes the absolute value of a scalar. In convolutional neural networks (CNNs), for a feature map $\mathcal{X} \in \mathbb{R}^{C \times W \times H}$ associated with C channels, width W and height H, we compactly denote the features corresponding to the ith channel by $\mathcal{X}_i \in \mathbb{R}^{W \times H}$, $i \in [C]$, which equals the ith front slice of $\mathcal{X}$.

Below, we provide a perturbation error bound of pruning a certain channel. Here we consider the sequential concatenation of `IN`, `relu` and convolution (`conv`) as the basic building block to construct (the generator of) GANs. Assuming the input feature map $\mathcal{X} \in \mathbb{R}^{C \times W \times H}$ of the channel C, width W and height H before `conv` we have[2]

$$\begin{aligned} \texttt{IN}: \quad \mathcal{Y}_i &= (\mathcal{X}_i - \mu_i)/\sigma_i, \mathcal{S}_i = \gamma_i \mathcal{Y}_i + \beta_i, \\ \texttt{relu}: \quad \mathcal{T}_i &= \texttt{relu}(\mathcal{S}_i), \end{aligned}$$

where μ_i, σ_i are the mean and variance, respectively, γ_i and β_i are re-scalar parameters, and $i \in [C]$. Furthermore, we assume the kernel tensor in `conv` as a fourth-order tensor $\mathcal{W} \in \mathbb{R}^{C \times D \times K \times K}$, of which K corresponds to the filter size and C, D corresponds to the input and output channel size, respectively. Then the output of `conv` can be written as

$$\mathcal{Z} = \mathcal{W} \star \mathcal{T} \in \mathbb{R}^{D \times W \times H}, \tag{2}$$

where $\star$ denotes the convolution operator. Note that the bias term in the normal `conv` is ignored since the following `IN` in the network will eliminate the effect by the bias. Below, we first give the relationship between the output of `conv` and each channel of the input feature map:

Lemma 1 *Let $\mathbf{y}_i = vec(\mathcal{Y}_i) \in \mathbb{R}^{WH}, \mathbf{z} \in \mathbb{R}^{DWH}$ be the vectorlization of $\mathcal{Y}_i$ and $\mathcal{Z}$ respectively, and define the vectors $\mathbf{w}_{i,j} \in \mathbb{R}^{WH}$, $i \in [C]$, $j \in [D]$, $p, g \in [K]$, which satisfy*

$$\mathbf{w}_{i,j}(l) = \begin{cases} \mathcal{W}(i, j, g, p) & \text{if} \quad l = (g-1)H + p \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

[2] Here we omit the stability constant ε for brevity of the theoretical discussion.

where $l \in [WH]$. We then have the following equation hold:

$$\mathbf{z} = \sum_{i\in[C]} \underbrace{\begin{bmatrix} circ(\mathbf{w}_{i,1}) \\ \vdots \\ circ(\mathbf{w}_{i,D}) \end{bmatrix}}_{\mathbf{G}_i:=} \texttt{relu}(\gamma_i \mathbf{y}_i + \beta_i). \tag{4}$$

The proof is given in the supplementary material. Since $\mathbf{y}_i$ is the "normalization" of $\mathcal{X}_i$, the ith channel of the input feature map, Lemma 1 shows that the output of `conv` equals the summation of non-linearizatrion of *each channel* of the input. It implies that pruning each channel would impact the output *independently* even after the `conv` operation. Next, we prove that the perturbation by pruning is ONLY determined by the learnable variables including $\mathcal{W}, \gamma$, and β. To achieve the goal, we first define a novel measure function, which reflects the sensitivity of our (`IN, relu, conv`) system by the pruning operations.

Definition 1 (Sensitivity Measurement) *Given the kernel $\mathcal{W}$ in `conv`, and the variables γ, β in `IN`, we define the sensitivity measurement as a matrix function $F(\mathcal{W}, \gamma, \beta) : \mathbb{R}^{C\times D\times K\times K} \times \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}^{C\times D}$, of which the (i, j)th entry $F_{i,j}$ obeys*

$$F_{i,j}(\mathcal{W}, \gamma, \beta) = \sqrt{WH}|\gamma_i| \sqrt{\sum_{g,p\in[K]} \mathcal{W}(i, j, g, p)^2} + |\beta_i| \left| \sum_{g,p\in[K]} \mathcal{W}(i, j, g, p) \right|, \tag{5}$$

where γ_i, β_i, $i \in [C]$ denotes the ith entry of γ and β, respectively.

Remark. As shown in Eq. (5), the function $F_{i,j}$ is non-negative, and consists of the "strength" of the (i, j)th filter in $\mathcal{W}$ weighed by γ and β. It implies that *the (i, j)th entry of F reflects the contribution from ith input channel to the jth output channel.*

To validate this claim, we below prove that the perturbation of the output by pruning is bounded by Eq. (5). Specifically,

Proposition 1 (Perturbation Error Bound) *Assume $\mathcal{Z} \in \mathbb{R}^{D\times W\times H}$ to be the output of `conv`, and $\mathcal{Z}^{\bar{i}}$ to denote the result by pruning the ith channel of the input. then the norm of perturbation $\Delta_i = \mathcal{Z} - \mathcal{Z}^{\bar{i}}$ is bounded by the following conditions: if $\gamma_i = 0$, then $\|\Delta_i\|_{\ell_1} = 0$; otherwise, we have*

$$\|\Delta_i\|_{\ell_1} \leq \begin{cases} WH \sum_{j\in[D]} F_{i,j}(\mathcal{W}, \gamma, \mathbf{0}), & \beta_i \geq \tau_i \\ WH \sum_{j\in[D]} F_{i,j}(\mathcal{W}, \gamma, \beta), & |\beta_i| < \tau_i \\ 0, & otherwise \end{cases}, \tag{6}$$

where $\mathbf{0} \in \mathbb{R}^C$ denotes the full-zero vector and $\tau_i = \sqrt{WH}|\gamma_i|$ for all $i \in [C]$.

It is worth noting that the conditions $\gamma_i = 0$ or $\beta_i \leq -\tau_i$ uncommonly happen in practice, and pruning the corresponding channels does not affect the output. The proposition shows that the perturbation error bound is fully determined by $\mathcal{W}, \gamma, \beta$.

Proposition 1 implies the worst-case scenario when pruning a certain channel. Therefore we could apply the perturbation error bound as a part of the loss function, by which we enforce a part of the channels have smaller bounds than the rest. Then we could prune these channels without having over-changed outcomes, resulting in better performance and more stable training progress.

4 Pruning Approach

We next demonstrate how to practically apply Eq. (6) to the pruning algorithm. We denote G, D as the generator and discriminator in GANs, respectively. Since only the generator G will be deployed on edge devices while the discriminator will be discarded after training. Therefore, in this work, we only consider representing G with a more compact form.

Step 1, Use Perturbation Error Bound as a Loss. Assume $\mathcal{W}_{[l]} \in \mathbb{R}^{C_l \times D_l \times K_l \times K_l}, \gamma_{[l]} \in \mathbb{R}^{C_l}, \beta_{[l]} \in \mathbb{R}^{C_l}$ to be all the learnable variables in the lth sub-block of G for $l \in [L]$, then the new loss function, namely Bound-Induced GAN pruning (BIG), is given by

$$L_{BIG}(\{\mathcal{W}_{[l]}\}, \{\gamma_{[l]}\}, \{\beta_{[beta]}\}) = \sum_{l\in[L]} \sum_{i\in[C_l]} P_{l,i}\left(\mathcal{W}_{[l]}, \gamma_{[l]}, \beta_{[l]}\right). \tag{7}$$

In the equation, $P_{l,i} = 0$ if $\gamma_{[l],i} = 0$, while if $\gamma_{[l],i} \neq 0$, then

$$P_{l,i} = \begin{cases} \sum_{j\in[D]} F_{i,j}\left(\mathcal{W}_{[l]}, \gamma_{[l]}, \mathbf{0}\right), & \beta_i \geq \tau_{[l],i} \\ \sum_{j\in[D]} F_{i,j}\left(\mathcal{W}_{[l]}, \gamma_{[l]}, \beta_{[l]}\right), & |\beta_i| < \tau_{[l],i} \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

where $\tau_{[l],i} = \sqrt{W_l H_l}|\gamma_{[l],i}|$ and W_l, H_l denotes the width and height of the input feature map in the lth block. We can see from Eqs. (7) and (8) that the new loss function is related to the sum of perturbation bound given in Eq. (6). **Therefore, minimizing the loss (7) will push the model to learn more distinguish channels, which are selected to be pruned or not.** Here, by "distinguish", we mean the contributions of channels to the output visual effect are learned to be smaller or larger, hence we can prune the ones with small contributions. This is achieved by the separated constrains define in (8). In practice (deployed in our experiment), the overall loss function is suggested as

$$L_{all} = L_{GAN} + \lambda_1 L_{dist} + \lambda_2 L_{BIG}, \tag{9}$$

where $\lambda_1, \lambda_2 > 0$ denotes the tuning parameters, L_{GAN} and L_{dist} denote the ordinary GAN loss and distillation loss [43], respectively. The details about these 2 losses are provided in the supplementary material.

Step 2, Update Re-scalar Parameters as Pruning. During the training, we can assign the re-scalar parameters `IN` to 0 to prune their corresponded channels once they cause only small perturbations to outputs. Formally, at step t, we conduct the update $\gamma^{(t)}_{[l],i} = \beta^{(t)}_{[l],i} = 0$, if one of the following four conditions is satisfied:

$$\begin{aligned} &(\mathtt{i}) \quad \beta^{(t-1)}_{[l],i} \leq -\tau^{(t-1)}_{[l],i}, \\ &(\mathtt{ii}) \quad \gamma^{(t-1)}_{[l],i} = 0, \\ &(\mathtt{iii}) \quad \frac{\sum_{j\in[D_l]} F_{i,j}\left(\mathcal{W}^{(t-1)}_{[l]}, \gamma^{(t-1)}_{[l]}, \mathbf{0}\right)}{\sum_{i\in[C_l]} P_{l,i}\left(\mathcal{W}^{(t-1)}_{[l]}, \gamma^{(t-1)}_{[l]}, \beta^{(t-1)}_{[l]}\right)} < \rho_1, \\ &(\mathtt{iv}) \quad \frac{\sum_{j\in[D_l]} F_{i,j}\left(\mathcal{W}^{(t-1)}_{[l]}, \gamma^{(t-1)}_{[l]}, \beta^{(t-1)}_{[l]}\right)}{\sum_{i\in[C_l]} P_{l,i}\left(\mathcal{W}^{(t-1)}_{[l]}, \gamma^{(t-1)}_{[l]}, \beta^{(t-1)}_{[l]}\right)} < \rho_2. \end{aligned} \tag{10}$$

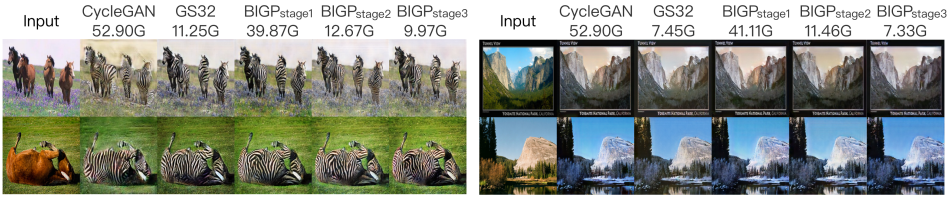

Figure 2: Examples of CycleGAN pruning results. The methods and their FLOPs (in G) are annotated above the images. Each row shows a different sample.

Table 1: Quantitative evaluation of GAN pruning results. CycleGAN donates the original unpruned model. $\uparrow$ / $\downarrow$ means the larger/smaller value, the better compression performance.

Dataset	Method	FLOPs	Memory	FID($\downarrow$)	$r_{ops}(\uparrow)$	$r_{mem}(\uparrow)$
horse2zebra	CycleGAN [51]	52.9G	43.51MB	74.04	1.00	1.00
	GS32[43]	11.68G	9.1MB	91.46	4.53	4.78
	BIGP$_{\text{Stage1}}$	39.87G	32MB	70.53	1.32	1.36
	BIGP$_{\text{Stage2}}$	12.7G	10MB	77.89	4.17	4.35
	BIGP$_{\text{Stage3}}$	9.97G	8MB	82.99	5.31	5.44
summer2winter	CycleGAN [51]	52.9G	43.51MB	77.76	1.00	1.00
	GS32[43]	7.45G	6.62MB	80.60	7.10	6.57
	BIGP$_{\text{Stage1}}$	41.11G	31.24MB	77.48	1.29	1.39
	BIGP$_{\text{Stage2}}$	11.46 G	9.89MB	76.60	4.62	4.40
	BIGP$_{\text{Stage3}}$	7.33G	6.51 MB	78.87	7.22	6.68

where $\rho_1, \rho_2 > 0$ denotes the hyper-parameters controlling the tolerance. Otherwise, they are updated by the general gradient descent methods. That is

$$\gamma_{[l],i}^{(t)} = \gamma_{[l],i}^{(t-1)} - \eta^{(t-1)} \nabla L_{all,\gamma}^{(t-1)}, \quad \beta_{[l],i}^{(t)} = \beta_{[l],i}^{(t-1)} - \eta^{(t-1)} \nabla L_{all,\beta}^{(t-1)}, \tag{11}$$

where $\nabla L_{all,\gamma}^{(t-1)}$ and $\nabla L_{all,\beta}^{(t-1)}$ respectively denote the gradient of L_{all} *w.r.t.* $\gamma_{[l],i}^{(t-1)}$ and $\beta_{[l],i}^{(t-1)}$, and $\eta^{(t-1)}$ denotes the learning rate at step $t-1$.

Insight of the (iii),(iv)th conditions in Eq. (10): Intuitively, the (iii),(iv)th conditions in Eq. (10) reflect the ratio of the perturbation by pruning the ith channel. If the ith channel satisfies the $3,4$th conditions with small $\rho_i, i = 1$ or 2, it implies that the output of the lth block would be not significantly changed even though discarding the ith channel. The difference between the two conditions is that the fourth condition focuses on the first term in Eq. (5) by setting $\beta_i = 0$, while the third condition takes the whole values of Eq. (5) into account. We suggest $\rho_1 \ll \rho_2$ in practice (they roughly follow $\rho_1 = 1e-4$ and $\rho_2 = 1e-3$ in the experiments).

In this work, we follow the training and pruning strategy in [43] yet γ, and β is updated by the new form as mentioned beforehand. More details of the training and pruning methods used in the experiment are introduced in the supplementary material.

5 Experimental analysis

5.1 Dataset, Metrics and Configurations

Dataset. We employ the following 2 datasets in our experiments, and the details can be found in [51]: `Horse2zebra` dataset is a subset of ImageNet-1K [4]. We employ 1,067 horse images for training, 120 horse images, and 140 zebra images for testing; `summer2winter` dataset is collected from Flickr. We employ 1,231 summer scenes for training and 309 summer and 240 winter scenes for testing. We employ the generated images for training for distillation, which is a common choice for GAN compression. Generated images from the teacher models help train the small models to output image styles closer to the teacher models. We also employ the target domain datasets for the L_{GAN} loss. Also, the target domain image for testing is only employed for the calculation of the FID score.

Metrics. We employ 3 metrics to evaluate the performance, with G_0 denoting the original model and G denoting the model to be evaluated, respectively: Flops and its compression ratio r_{ops} = Flops of G_0/Flops of G; Memory size and its compression ratio r_{mem} = Model size of G_0/Model size of G; FID score [11] between the generated samples and the test samples, which is used for evaluating the generation quality.

Configurations. We follow the major configuration used in Gan Slimming [43]. Here we briefly introduce the multi-stage training strategy employed in the experiments.

Similar to the learning rate decay strategy adopted in the optimization progress, we also adjust the weight of L_{BIG}, *i.e.*, λ_2; and the thresholds for pruning channels, *i.e.*, ρ_1, ρ_2. We use 3 sets of these hyper-parameters denoted as `Stage1`, `Stage2`, `Stage3`. The motivation is that we first stabilize the training progress in `Stage1`, then perform pruning aggressively in `Stage2`. Finally, we fine-tune the pruning results in `Stage3`. Although the training progress is staged, **only the fine-tuned model after `Stage3` is the desired outcome**. Detailed hyper-parameters and the thresholds will be provided in the supplemental materials. We also conduct a training stability analysis in the supplementary materials to analyze the choice of the three stages strategy.

5.2 Results

Comparison with Gan Slimming. Gan Slimming [43] is a commonly used baseline model that integrates pruning, distillation, and quantization. Its pruning is achieved by the sparse regularization term on the channels, which is currently the most applied approach. Therefore, we conduct our experiments and pruning analysis based on this approach.

The qualitative and quantitative results are shown in Figure 2 and Table 1, respectively. Our method is denoted as BIGP, with the stage subscripts. CycleGAN denotes the original unpruned model, and GS32 denotes the best model reported in the Gan Slimming paper. It can be seen that our `Stage3` model has less computational complexity than GS32 in terms of both Flops and Memory, and the visual quality of generated samples is also better than the GS32 model in both datasets.

Zero-shot Model Pruning Analysis. We next conduct zero-shot model pruning experiments to examine the correctness of our error model defined in Eq. (5). Zero-shot pruning means removing (set to $\mathbf{0}$ matrix in practice) the channels without re-training or fine-tuning. The target unpruned model is the super-network ($\mathsf{G}_{\text{super}}$) provided in [23]. And for reference, we also employ their pruned sub-network (G_{sub}) in [23] using the evolutionary algorithm. We then prune the super-network to match the structure of the sub-network.

We evaluated the effectiveness of the measures using two standards: the FID score and the correlativity between pruned channels. The latter counts the number of pruned channels in both the sub-network and the pruned network using a specific measure. Concretely, let $\mathbb{C} = \{c_i, i \in [C]\}$ be the set of channels in $\mathsf{G}_{\text{super}}$, where C is the total number of channels. Let $\mathbb{P}_1 \subset \mathbb{C}$ be the set of channels retained after pruning $\mathsf{G}_{\text{super}}$, and let $\mathbb{P}_2 \subset \mathbb{C}$ be the channels set of G_{sub}. Then we use $\texttt{eff} = \texttt{count}(\mathbb{P}_1 \cap \mathbb{P}_2)/\texttt{count}(\mathbb{P}_2)$ to evaluate the correlativity. Intuitively, the G_{sub} is a sub-optimized solution to the weights pruning problem. Therefore, if the pruned channels obtained from a measure have high correlativity with that in G_{sub}, we consider the measure has better effectiveness.

Table 2: Quantitative analysis of zero-shot pruning on the `horse2zebra` dataset.

(a) The comparison of computational cost. The super-/sub-net are un-pruned and reference small models from [23].

Method	FLOPs	Memory	FID(↓)	`eff`(↑)
super-net	4.59G	2.1MB	60.10	—
sub-net	2.35G	1.5MB	68.77	1.0
GS32	2.35G	1.5MB	188.56	0.5333
BIGP	2.35G	1.5MB	88.47	0.9291

(b) The comparison of channel importance measures. (B,C) stands for the C-th channel from the B-th block.

(B,C)	$\lvert\gamma\rvert$ [34]	Error Bound	Satisfy Eq. (10)	Influence
(6,101)	0.0240	101.12	✗	Significant
(6,22)	0.2283	1.18	✓	Trivial

The quantitative results are shown in Table 2a, and we provide qualitative results in the supplementary materials. We compare with a baseline model that prunes channels randomly (which can be seen as the ℓ_1 norm [34] method without optimization, so we denote it as GS32 for consistency). It can be seen that our proposed measure achieved a correlativity of 0.9291 with the sub-optimal solution.

Zero-shot Pruning on Distillation Models. Next, we graft our method to the distillation-based methods that recently achieved state-of-the-art compression ratio and FID score. Concretely, we conduct the following experiments: 1. Zero-shot pruning on the OMGD officially released models; 2. Add the BIG loss term to OMGD [39] distillation framework, then conduct zero-shot pruning on the trained models. By this, we can examine the effectiveness of the pruning error model. The details of the configurations and the channel analysis are included in the supplementary materials. The results are reported in Table 3, the "+ZSP" stands for zero-shot pruning the above models, and "+L_{BIG}" stands for pre-training with the BIG loss. We can see that, even in the currently reported best performance distillation models, there still exist channels that can be pruned without influencing the generation results. Also, with our proposed BIG loss term, we achieve new state-of-the-art performance under both datasets.

6 Conclusion

In this paper, we develop a novel perturbation model by which we prove the upper bound of perturbations caused by pruning channels of hidden layers. Then we verify the proposed model by conducting zero-shot (without fine-tuning) pruning and indicate its effectiveness in the practical scene. Meanwhile, we suggest a new loss term and a training framework conduct the pruning progress. Finally, in the experimental analysis, we demonstrate that the proposed method outperforms the state-of-the-art method in terms of compression ratio, training stability, and similarity to the uncompressed model.

Table 3: Performance of knowledge distillation models combining the BIG loss pre-training and/or zero-shot pruning. † stands for the officially released models, * stands for our pre-trained models.

horse2zebra			
Model	FLOPs	FID	# Chan. Pruned
GAN-Comp. [25]	2.67G	64.95	–
DMAD [24]	2.41G	62.96	–
CAT [30]	2.55G	60.18	–
GCC [25]	2.40G	59.19	–
OMGD [33]	1.408G†	51.97	–
+ZSP	1.406G	51.70	2
+L_{BIG}	1.408G*	46.72	–
+L_{BIG}+ZSP	1.397G	47.03	10

summer2winter			
Model	FLOPs	FID	# Chan. Pruned
Auto-GAN [25]	4.34G	78.33	–
GAN-KD [1]	3.20G	80.10	–
SP-KD [26]	3.20G	76.59	–
DMAD [24]	3.18G	78.24	–
OMGD [33]	1.408G†	73.79	–
+ZSP	1.404G	73.70	6
+L_{BIG}	1.408G*	73.12	–
+L_{BIG}+ZSP	1.398G	73.13	9

Acknowledgement

This work was partially supported by JSPS KAKENHI (Grant No. 20H04249, 20H04208) and the National Natural Science Foundation of China (Grant No. 62006045).

References

[1] Angeline Aguinaldo, Ping-Yeh Chiang, Alex Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing gans using knowledge distillation. *arXiv preprint arXiv:1902.00159*, 2019.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[3] Hanting Chen, Yunhe Wang, Han Shu, Changyuan Wen, Chunjing Xu, Boxin Shi, Chao Xu, and Chang Xu. Distilling portable generative adversarial networks for image translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3585–3592, 2020.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia. Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[5] Thomas Elsken, Jan Hendrik Metzen, Frank Hutter, et al. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21, 2019.

[6] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[7] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[8] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.

[9] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019.

[10] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017.

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[14] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[15] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[18] 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*, pages 4401–4410, 2019.

[19] Yann LeCun, John S Denker, Sara A Solla, Richard E Howard, and Lawrence D Jackel. Optimal brain damage. In *NIPs*, volume 2, pages 598–605. Citeseer, 1989.

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[21] Chao Li and Zhun Sun. Evolutionary topology search for tensor network decomposition. In *International Conference on Machine Learning*, pages 5947–5957. PMLR, 2020.

[22] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[23] Muyang Li, Ji Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han. Gan compression: Efficient architectures for interactive conditional gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5284–5294, 2020.

[24] Shaojie Li, Mingbao Lin, Yan Wang, Fei Chao, Xudong Mao, Mingliang Xu, Yongjian Wu, Feiyue Huang, Ling Shao, and Rongrong Ji. Learning efficient gans for image translation via differentiable masks and co-attention distillation. *arXiv e-prints*, pages arXiv–2011, 2020.

[25] Shaojie Li, Jie Wu, Xuefeng Xiao, Fei Chao, Xudong Mao, and Rongrong Ji. Revisiting discriminator in gan compression: A generator-discriminator cooperative compression scheme. *arXiv e-prints*, 2021.

[26] Zeqi Li, Ruowei Jiang, and Parham Aarabi. Semantic relation preserving knowledge distillation for image-to-image translation. In *European conference on computer vision*, pages 648–663. Springer, 2020.

[27] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[28] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[30] Yuchen Liu, Zhixin Shu, Yijun Li, Zhe Lin, Federico Perazzi, and Sun-Yuan Kung. Content-aware gan compression. *arXiv e-prints*, 2021.

[31] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744, 2017.

[32] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

[33] Jian-Hao Luo and Jianxin Wu. An entropy-based pruning method for cnn compression. *arXiv preprint arXiv:1706.05791*, 2017.

[34] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017.

[35] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

[36] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[37] Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry Vetrov. Tensorizing neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 442–450, 2015.

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[39] Yuxi Ren, Jie Wu, Xuefeng Xiao, and Jianchao Yang. Online multi-granularity distillation for gan compression. *arXiv e-prints*, 2021.

[40] Han Shu, Yunhe Wang, Xu Jia, Kai Han, Hanting Chen, Chunjing Xu, Qi Tian, and Chang Xu. Co-evolutionary compression for unpaired image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3235–3244, 2019.

[41] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[42] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[43] Haotao Wang, Shupeng Gui, Haichuan Yang, Ji Liu, and Zhangyang Wang. Gan slimming: All-in-one gan compression by a unified optimization framework. In *European Conference on Computer Vision*, pages 54–73. Springer, 2020.

[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*, pages 8798–8807, 2018.

[45] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[46] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *arXiv preprint arXiv:1608.03665*, 2016.

[47] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. Nisp: Pruning networks using neuron importance score propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9194–9203, 2018.

[48] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[49] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[50] Chenglong Zhao, Bingbing Ni, Jian Zhang, Qiwei Zhao, Wenjun Zhang, and Qi Tian. Variational convolutional neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2780–2789, 2019.

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[52] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.