Model Compression with Generative Adversarial Networks

Anonymous authors
Paper under double-blind review

Abstract

The ever-increasing accuracy of machine learning models often comes at the expense of higher computational costs and memory requirements at test time, making them impractical to deploy on memory-constrained or CPU-constrained devices. *Model compression* (also known as *distillation*) is a technique to compress a complex model into a simpler one while maintaining most of the original accuracy. This can be done by using the same dataset for both the model training and compression tasks or by exploiting additional data. However, in many real-world applications, additional data are not available, and the repeated use of the original training data leads to suboptimal compression. In this work, we propose to use generative adversarial networks (GANs) to approximately sample from the distribution of the original data, thus generating "unlimited" synthetic data that can be used to perform the compression task. Our *GAN-assisted model compression* (GAN-MC) approach shows significant improvement in compressing complex models such as deep neural networks and large random forests on both image and tabular datasets. Furthermore, based on the model compression results, we propose a comprehensive metric—the *Compression Score*—to evaluate the quality of generative models, which captures both the discriminability and the diversity of the synthetic data. We show that the Compression Score performs well in cases when the popular Inception Score fails.

1 Introduction

Modern machine learning models have achieved remarkable levels of accuracy, but their complexity can make them slow in generating predictions, expensive to store and hard to deploy for real-world use. Ideally, we would like to replace such cumbersome models with simpler models that perform equally well. One way to address this problem is to perform *model compression* (also known as *distillation*), which consists in training a student model to mimic the relative class probabilities of a teacher model (Bucilu et al., 2006). For example, expensive deep neural network (DNN) teachers have been used to train inexpensive shallow neural network (Ba & Caruana, 2014; Urban et al., 2016) and decision tree students (Craven & Shavlik, 1996; Frosst & Hinton, 2017). Other model compression methods have also been developed recently; examples include parameter sharing (Chen et al., 2015), network pruning (Han et al., 2015), and predicting network parameters (Denil et al., 2013).

Despite the widespread adoption of model compression techniques, a critical problem remains unsolved: given well-trained complex models, in most cases, no new data are available for the compression task. Rather, the same data is typically used both to train a complex model and to compress it into a cheaper one, leading to suboptimal compression performance.

When not enough real data is available to perform model compression, using synthetic data generated by high-quality generative models like generative adversarial networks (GANs) (Goodfellow et al., 2014) is an appealing and natural choice that has yet to be explored. With adversarial games between generators and discriminators, GANs are able to match the distributions of synthetic and real data. Impressive performance has been achieved for various data types including images (Goodfellow et al., 2014), text (Yu et al., 2017) and electronic health records (Choi et al., 2017). Here, we propose to use GANs to approximately sample from the distribution of the original data, thus generating "unlimited" synthetic data that can be used to perform the compression task.

A critical gap hampering the usage of GANs in the context of model compression is the difficulty in reliably configuring (e.g., identifying architectures and hyperparameters for) GANs to produce realistic data. This is due to the difficulty in evaluating generative models due to a lack of universal similarity metrics and the varied probabilistic criteria (Theis et al., 2015). The popular Inception Score (Salimans et al., 2016) quantifies discriminability based on the conditional label distribution estimated from a trained Inception model. However, the Inception Score favors synthetic data of an adversarial nature, because the discriminability it measures is tailored to a particular neural network classifier, not the underlying distribution of the data. Conditional generative models can easily generate synthetic samples which would produce high confidence predictions in many different neural architectures, including Inception, while diverging from the real data distribution.

On the other hand, the diversity of synthetic data is often measured by multi-Scale Structural Similarity (MS-SSIM) (Wang et al., 2004), which is built on the perceptual similarity formula for images. Human perception, however, may not fully capture the fidelity of synthetic data and is hard to define for non-image data, like tabular data. To address these shortcomings, we develop a *Compression Score* that quantifies the test accuracy of students trained using synthetic data; this offers a principled, robust and effective metric to evaluate GAN-generated datasets.

In summary, we make the following principal contributions in this paper:

1. We propose *GAN-assisted model compression* (GAN-MC), a simple approach to improving teacher-student compression by augmenting the compression training set with GAN data.
2. On CIFAR-10 image classification, we show GAN-MC consistently improves student test accuracy for a variety of deep neural network teacher-student pairings and two popular compression objectives.
3. For random forest teachers, we demonstrate 25 to 500-fold reductions in storage and execution costs with less than 1.2% loss in test performance across a suite of real-world tabular datasets.
4. We introduce a new Compression Score for evaluating the quality of GAN-generated datasets and illustrate its advantages over the popular Inception Score on CIFAR-10.

2 Model Compression with GANs

2.1 Deep Neural Network Compression

In the standard teacher-student approach to compressing a neural network classifier, a relatively inexpensive prediction rule, like a shallow neural network, is trained to predict the unnormalized log probability values—the *logits* z—assigned to each class by a previously trained deep network classifier. The inexpensive model is termed the *student*, and the expensive deep network is termed the *teacher*. Given dataset of n feature vectors paired with teacher logit vectors, $\{(x^{(1)}, z^{(1)}), ..., (x^{(n)}, z^{(n)})\}$, Ba & Caruana (2014) proposed framing the compression task as a multitask regression problem with L^2 loss,

$$L(\theta) = ||g(x;\theta) - z||_2^2. \tag{1}$$

Here, θ represents any student model parameters to be learned (e.g., the weights of the student network), and $g(x;\theta)$ is the vector of logits predicted by the student model for the input feature vector x.

Hinton et al. (2015) introduced an alternative compression objective function, indexed by a temperature parameter $T > 0$. Specifically, the student is trained to mimic the annealed teacher class probabilities,

$$q_j(z/T) = \exp(z_j/T)/\textstyle\sum_k \exp(z_k/T),$$

for each class j by solving a multitask regression problem with cross-entropy loss,

$$L_T(\theta) = -\textstyle\sum_j q_j(z/T)\log(q_j(g(x;\theta)/T)).$$

Hinton et al. (2015) showed that, under a zero-mean logit assumption, cross-entropy regression recovers L^2 logit-matching as $T \to \infty$; however, the two approaches can differ for small T. In Sec. 3, we demonstrate the effectiveness of using GAN data for both compression approaches on the CIFAR-10 dataset.

2.2 Random Forest Compression

Ensemble methods are important techniques in machine learning. Recently, random forests have achieved great success in large-scale tabular dataset (Breiman, 2001). However, at prediction time, the time and space complexity for random forests with a large number of trees and nodes can be prohibitive. Effectively mimicking a large random forest with a small forest or a single decision tree has the potential to reduce prediction runtime and storage costs by several orders of magnitude. Focusing on the common setting of binary classification with labels in $\{0, 1\}$, we propose to train a student regression random forest to mimic the teacher forest's predicted probability p of a datapoint x belonging to class 1.

2.3 GAN-assisted Model Compression (GAN-MC)

In a typical compression setting, as much data as possible has been dedicated to training the highly accurate teacher model, leaving little fresh data for training the student model. A standard solution is to simply reuse the teacher training set as the student training set. However, we will see in Secs. 3 and 4 that this leads to suboptimal student performance. To boost student performance and compression efficiency, we propose a simple solution: augment the compression training set with (unlimited) synthetic feature vectors generated by a high-quality GAN. We call this approach *GAN-assisted model compression* (GAN-MC).

Intuition. In a model compression task, a student is classically trained to mimic the predictions of a teacher on feature vectors drawn from the data distribution. When the original training sample is insufficient, synthetic data with a similar distribution can suffice to train an accurate student. The generator G of a GAN for instance can produce an unlimited number of fake feature vectors by transforming independent noise vectors drawn from a simple distribution. The distributions of the synthetic and real data are encouraged to align via an adversarial game between a generator G and a discriminator D.

To generate GAN feature vectors which capture the important class-related features, we use the auxiliary classifier GAN (AC-GAN) of Odena et al. (2016). The AC-GAN generator G produces a synthetic feature vector $X_{fake} = G(W, C)$ given a random noise vector W and an independent target class label C drawn from the real data class distribution. For any given feature vector x, the AC-GAN discriminator D predicts both the probability of each class label $P(C \mid x)$ and the probability of the source being real or fake, $P(S \mid x)$ for $S \in \{real, fake\}$. Given a training dataset $\mathcal{D}_{real}$ of labeled feature vectors, two components contribute to the AC-GAN training objective:

$L_{source} = \frac{1}{|\mathcal{D}_{real}|}\sum_{(x,c)\in\mathcal{D}_{real}} \log P(S = real \mid x) + \mathbb{E}_{W,C\sim p_c}[\log P(S = fake \mid G(W, C))]$ and

$$L_{class} = \frac{1}{|\mathcal{D}_{real}|}\sum_{(x,c)\in\mathcal{D}_{real}} \log P(C = c \mid x) + \mathbb{E}_{W,C\sim p_c}[\log P(C \mid G(W, C))], \qquad (2a)$$

representing the expected conditional log-likelihood of the correct source and the correct class of a feature vector, respectively. In the adversarial game, the generator G is trained to maximize $L_{class} - L_{source}$ and the discriminator D is trained to maximize $L_{class} + L_{source}$.

3 Deep Neural Network GAN-MC

Experiment setup. We now investigate how GAN-MC performs when used to compress convolutional deep neural networks (CNNs) for the CIFAR-10 dataset of Krizhevsky & Hinton (2009). CIFAR-10 consists of 32×32 RGB images from 10 classes, which are divided into 50,000 training and 10,000 testing samples.

We implement the two compression methods as introduced in Sec. 2.1. For the method which regresses the logits with L^2 loss, the teacher and the student are NIN and LeNet models, with test accuracies of 78.1% and 66.2% for the image recognition task. The student is trained to minimize the L^2 loss in Eq. 1. The Adam optimizer is used for training with learning rate 10^{-4}. For the method which uses the cross-entropy loss $L_T(\theta)$ of the soft probability with temperature T, we examine three additional networks: WideResNet-28-10, ResNet-18 and a 5-layer CNN with 3 convolution layers, with test accuracies of 95.8%, 94.2% and 78.8% for the image recognition task. The student's

Table 1: Test accuracy of the learned student in model compression with different neural network architectures for CIFAR-10. The student is trained to mimic the soft probability distribution from the teacher.

	Teacher	Student	Teacher Only	Student Only	Student Compressed with Training Data	Student Compressed with Training & GAN
1	NIN	LeNet	78.1%	66.2%	71.5%	**75.8%**
2	ResNet-18	5-layer CNN	94.2%	78.8%	84.4%	**86.6%**
3	WideResNet-28-10	ResNet-18	95.8%	94.2%	94.3%	**95.0%**

objective is $L(\theta) = \alpha L_T(\theta) + (1-\alpha)L_0(\theta)$, where $0 < \alpha \leq 1$ and $L_0(\theta)$ is the original cross-entropy classification loss $-\sum_j p_j \log(q_j(g(x;\theta)))$ and $p_j = 1$ for data in class j. For the teacher-student pair 1, 2 and 3 in Table 1, the temperature T is 5, 20 and 6, and α is set to be 0.9, 0.9 and 0.95. The Adam optimizer with learning rate 10^{-3} is used for the first two teacher-student pairs and SGD optimizer with learning rate decayed from 0.1 is used for the last one.

The AC-GAN is implemented in Keras (Chollet et al., 2015). The discriminator D is a CNN with 6 convolution layers and Leaky ReLU nonlinearity. The generator G consists of 3 'deconvolution' layers which transform the class c and noise w into a 32×32 image with 3 color channels. The latent size of noise W is 110. We use a Adam optimizer with learning rate 0.0002 and momentum term β_1 0.5, as suggested by Radford et al. (2015).

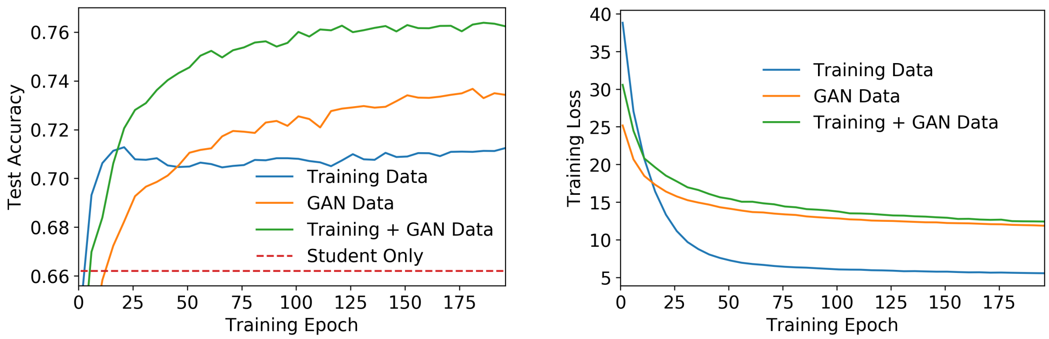
We examined three scenarios — compression on only training data, only GAN data and a mixture of training and GAN data. The GAN data are produced in real time during the training. The mixture of training and GAN data is realized by sampling from the training set with frequency f_{real} and generating GAN data with frequency f_{fake}. During the compression, the probability for the student to be trained on GAN data $p_{fake} = f_{fake}/(f_{fake} + f_{real})$ is fixed. In the experiment, we set p_{fake} to be 0.8 when the teacher is NIN and the student is LeNet. For the last two teacher-student pairs in Table 1, p_{fake} is 0.2. The effect of the choice of p_{fake} is studied below and in Fig. 2.

Results. Significant improvement is achieved when GAN data are used to compress the CNNs. A typical result of training a student by regressing the logits with L^2 loss is given in Fig. 1. The test accuracy of the teacher and the student are 78.1% and 66.2% when trained alone. The change of student's test accuracy during compression is depicted in Fig. 1a. At the end of the training, the test accuracy reaches 71.0%, 73.7% and 76.2% for student trained on only training data, only GAN data and a mixture of training and GAN data. An increase of 5% in accuracy is achieved by simply introducing GAN data. When the student is trained to predict the soft probability distribution of the teacher, the performance is also improved by training with GAN data, as given in Table 1.

Contribution of GAN data. The improvement is due to the mitigation of overfitting from the great enrichment of fresh data. The teacher is pre-trained on the training dataset with the training accuracy of 100%. When the student is made to mimic the teacher on the same training dataset, the overfitting effect could be severe. In contrast, when the student is trained on the GAN data, unlimited fresh data are provided and the overfitting effect is effectively mitigated.

During the compression training, there is a trade-off between the closeness to the real data distribution and the influence of overfitting. As seen in Fig. 1b, the loss incurred when training on real data first quickly decreases and then remains significantly smaller than the loss incurred with GAN data. At the start of the compression training, compression with real data is more effective, because the real data provide a more faithful reflection of the real data distribution, and the overfitting effect is not yet severe. Correspondingly, a quicker increase in test accuracy is observed at the start, as shown in Fig. 1a. After around 10 epochs, the influence of overfitting gradually increases and becomes dominant over the advantage of fidelity to the real data distribution. The loss for real training data becomes smaller than the loss with GAN data, as shown in Fig. 1b, and the test accuracy stops increasing, as confirmed by Fig. 1a.

Effect of the GAN training proportion parameter p_{fake}. Adopting the experimental setup of Fig. 1, we next examine how p_{fake}, the probability of selecting a GAN datapoint over a real dat-



(a) Test accuracy of the learned student.

(b) Training loss for logits regression.

Figure 1: Learning curves of model compression for CIFAR-10. The student is trained by regressing logits with L^2 loss of only training data (blue curve), only GAN data (green curve) and a mixture of the training data and GAN data (orange curve, $p_{fake} = 0.8$). The teacher and student are NIN and LeNet with test accuracies of 78.1% and 66.2% (red dashed curve in (a)). The results are averaged over 3 independent runs.

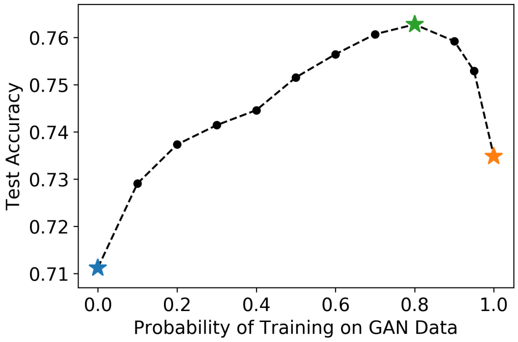


Figure 2: Student test accuracy vs. probability p_{fake} of training on GAN data in the CIFAR-10 compression setup of Fig. 1 (see Sec. 3 for more details). Compression only using training (resp. GAN) data corresponds to the case when p_{fake} equals to 0 (resp. 1). The colored data points are the final value of curves with the same color in Fig. 1a. We report the average of 3 independent runs.

apoint when training the student, affects compression performance. We plot the dependence of trained student test accuracy on p_{fake} in Fig. 2. When p_{fake} equals to 0, only training data are used in compression; when p_{fake} goes to 1, only GAN data are used. We observe a clear non-monotonic dependence with a combination of GAN and real datapoints providing significantly higher accuracy than GAN or real datapoints alone. We suspect this reflects the trade-off between the overfitting caused by training data reuse and the inability of a GAN to perfectly approximate the true data distribution.

4 Random Forest GAN-MC

Experiment setup. We next use three tabular datasets from the UCI machine learning repository and Kaggle to analyze how GAN-MC performs when used to compress large random forests for binary classification. A description of each dataset is given in Table 2. Higgs and MAGIC (MAGIC Gamma Telescope) are physics datasets from UCI (Dheeru & Karra Taniskidou, 2017). The nearly-class-balanced Higgs dataset was developed to learn whether a given observation was produced by a Higgs boson (Baldi et al., 2014). For our experiment, a subset of 200,000 class-balanced datapoints were selected uniformly at random. The target of the MAGIC dataset is the registration of high energy gamma particles in a gamma telescope. The Evergreen (StumbleUpon Evergreen) dataset is from Kaggle (`https://www.kaggle.com/c/stumbleupon`) and its target is whether webpages are evergreen or not. Feature extraction is carried out on Evergreen dataset and 29 continuous features are obtained (Liu et al., 2017). We split the datasets into training and test sets uniformly at random, with training split sizes given in Table 3.

In the experiment, the teacher is a random forest classifier with 500 trees, and the student is a regression random forest with one to 20 trees, both trained using scikit-learn (Pedregosa et al., 2011). The trees in teacher and student have similar depth after training. All the features are considered for the best split. The teacher is pre-trained on the training data. When the student is trained without the knowledge from the teacher, the class labels (0 and 1) are viewed as real value targets. For the AC-GAN implementation in Keras, both the generator and the discriminator are one layer fully-connect neural network with 50 neurons and ReLU activation. The noise vector W has length 100. The Adam optimizer with learning rate 0.0002 and momentum term $\beta_1 = 0.5$ is used.

We study three scenarios: compression on a set of training data, a set of GAN data and a mixture of training and GAN data. The GAN data are generated before the training with number n_{fake}. The mixture set is generated by pooling the training and GAN data together. Considering the running time complexity, here we set $n_{fake}/n_{real} = 9$, where n_{real} is the number of real data.

Results. The results of compressing a random forest with 500 trees into a single decision tree are given in Table 3. We use test accuracy as our performance metric for the balanced Higgs dataset and test AUC for the unbalanced MAGIC and Evergreen datasets. We experiment with a variety of training dataset sizes, ranging from n = 1k to n = 100k to demonstrate the versatility of GAN-MC. For all datasets, compression with GAN data outperforms compression with training data and substantially outperforms the student model trained without compression. Moreover, for the Higgs dataset, the accuracy boost from GAN compression (62.1% to 68.5% on Higgs 100k) is 10 times the accuracy boost achieved using training data compression (62.1% to 62.7%).

The example of the Evergreen dataset is also enlightening. Compression with training data increases student test AUC from 0.731 to 0.856, and compression with only GAN data yields a further improvement of 0.882, nearly matching the 0.889 test AUC of the teacher. Remarkably, this is achieved with a single decision tree which demands 500 times less storage space and computation than the teacher at prediction time.

Fig. 3 displays student test performance as a function of the number of trees in the student forest. For each dataset save Higgs 1k, compressing with GAN data offers the best (or nearly the best) performance for all forest sizes. Indeed, for the Evergreen and MAGIC datasets, near-maximal performance is achieved by a single GAN-MC decision tree, with additional trees yielding relatively minor performance gains. For Higgs 1k, the combination of training and GAN data offers the best performance for all multi-tree forests, with an accuracy boost consistently 2-4 times that of compression with training data alone.

Similar to the deep neural network compression, the significant improvement arises from the mitigation of the overfitting effect. Here the best performance is usually achieved when only GAN data are used. This is because the GAN data simulate the real data distribution well and the overfitting effect is particularly severe when the large random forest is pre-trained on the same training dataset.

5 Evaluating GANs with a Compression Score

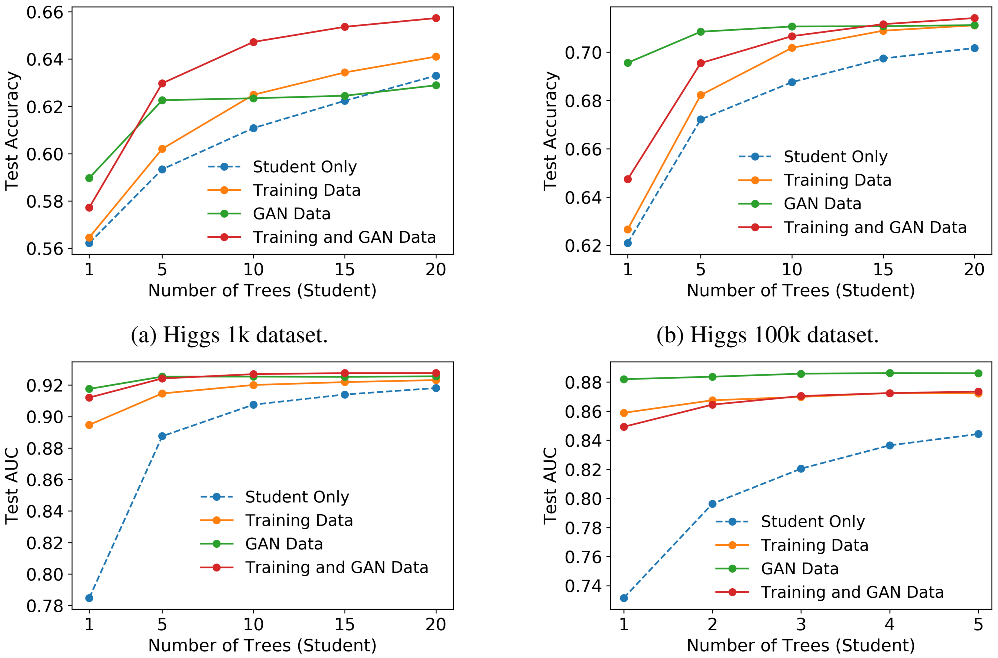
The evaluation of synthetic datasets is an important but challenging task. Two criteria commonly considered essential for a high-quality dataset are datapoint diversity and *discriminability*, the ability of datapoints to be correctly classified with high confidence. Current metrics for discriminability,

Table 2: Description of tabular datasets used for GAN-assisted random forest compression.

Dataset	# Datapoints	# Features	Class Imbalance
Higgs	200k	28	0: 50.0%; 1: 50.0%
MAGIC	19k	11	0: 35.2%; 1: 64.8%
Evergreen	7k	29	0: 48.7%; 1: 51.3%

Table 3: Test accuracy (Higgs) and test AUC (Evergreen and MAGIC) of the learned student in model compression. Here a random forest with 500 trees is compressed into a single decision tree.

Dataset	Training Data Size	Teacher Only	Student Only	Student Compressed with Training Data	GAN Data	Training & GAN
Higgs	1k	66.4%	56.2%	56.5%	**59.0%**	57.7%
	100k	72.6%	62.1%	62.7%	**69.6%**	64.7%
MAGIC	10k	0.935	0.785	0.895	**0.918**	0.912
Evergreen	5k	0.889	0.731	0.856	**0.882**	0.849



(a) Higgs 1k dataset.

(b) Higgs 100k dataset.

(c) MAGIC dataset.

(d) Evergreen dataset.

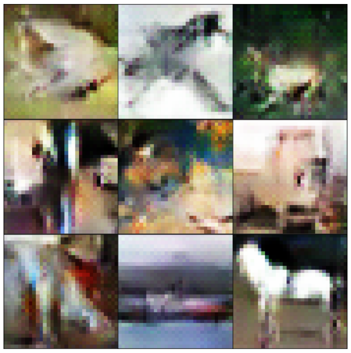
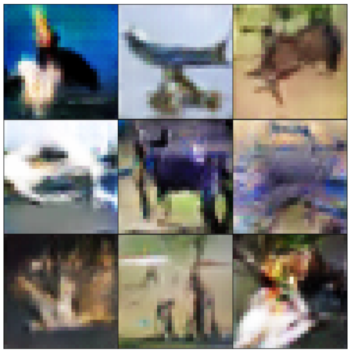
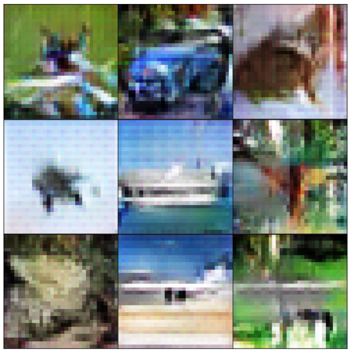
Figure 3: Test accuracy (Higgs) and test AUC (Evergreen and MAGIC) of the learned student in model compression. Here a 500 tree random forest is compressed into a compact random forest. When the student is trained without the knowledge from the teacher, its performance is given by the blue dashed curve. The compression is carried out on only training data (orange curve), only GAN data (green curve) and a mixture of the training data and GAN data (red curve).

like the popular Inception Score (Salimans et al., 2016), rely on the predictions of pre-trained neural networks. As a result, these discriminability measures are subject to the idiosyncrasies of those neural networks and need not reflect the nature of real data. For example, if the classification loss L_{class} is heavily upweighted relative to the source loss L_{source} while training an AC-GAN, the generator will be more likely to produce feature vectors classified with high confidence by neural networks; such feature vectors need not resemble real data but will nevertheless receive high Inception Scores (a higher score is meant to indicate a higher-quality sample). To simultaneously account for both discriminability and diversity in a holistic and principled manner, we propose to use the performance of a student trained on GAN data as a measure of GAN dataset quality.

Compression Score. To evaluate the quality of a generated dataset $\mathcal{D}$ relative to a real dataset $\mathcal{D}_{real}$, we define a *Compression Score* based on the test accuracy $\mathrm{acc}(\mathcal{D})$ of a student trained on $\mathcal{D}$ to mimic a teacher previously trained on $\mathcal{D}_{real}$:

$$\texttt{CompressionScore}(\mathcal{D};\mathcal{D}_{real}) = \frac{\mathrm{acc}(\mathcal{D}) - \mathrm{acc}_{\mathrm{random}}}{\mathrm{acc}(\mathcal{D}_{real}) - \mathrm{acc}_{\mathrm{random}}}.$$

Table 4: Inception and Compression Scores for CIFAR-10 images; larger scores should signify higher quality images. Inferior data generated by training well-trained GAN for 10 additional epochs using only the classification objective L_{class} (see Sec. 5.1). Inception Score increases for inferior images despite evident unrealistic artifacts. Compression Score decreases for inferior images.

Real Data	Well-trained GAN	Inferior GAN
		
Inception: 11.2 ± 0.1 Compression: 1.007 ± 0.006	Inception: 5.80 ± 0.06 Compression: 0.812 ± 0.004	Inception: 5.93 ± 0.06 Compression: 0.694 ± 0.005

Here $\text{acc}_{\text{random}} = 1/n_{class}$ is the accuracy of a random classifier that guesses a class uniformly at random, and n_{class} is the number of classes.

By design, the Compression Score equals 1 for the real dataset $\mathcal{D}_{real}$ and goes to 0 when the fake data distribution differs from the real one and little class-related feature is captured. The Compression Score is between 0 and 1 in most cases, and a higher value indicates better quality. To reduce time complexity and mitigate the influence of overfitting, we train the student for only 5 epochs; our experiments in Sec. 5.1 suggest that this is sufficient to reflect the GAN data quality.

Intuition. Student performance on real test data after training on fake data provides a principled and holistic measure of synthetic data quality, reflecting both discriminability and diversity. The student is trained to mimic the the teacher on the fake data. The condition that the student also mimics the teacher well on the real data is that the fake data correctly captures the class-related features of real samples and closely approximates the real data distribution. In particular, we would not expect a student trained on unrealistic or adversarial fake data to perform well on real test data.

5.1 Evaluating GANs: An Illustration with CIFAR-10

To illustrate the potential benefit of the Compression Score over the commonly-used Inception Score, we reinstate the CIFAR-10 experimental setup of Fig. 1. As discussed in Sec. 3, the initial epochs of compression training are particularly sensitive to the similarity between the true data distribution and the candidate dataset $\mathcal{D}$ and particularly insensitive to overfitting. To exploit these desirable properties and simultaneously reduce evaluation time, we train each student for only 5 epochs. The standard error is obtained from 3 independent runs.

Evaluation. We evaluate the compression score on real data, well-trained GAN data and inferior data which have high confidence classifications under the teacher network but do not resemble real data. The inferior data are generated by training the well-trained AC-GAN for 10 additional epochs only on the classification objective L_{class} given in Eq. (2a). That is, both the generator G and discriminator D are trained to maximize L_{class}, while ignoring the traditional GAN objective component L_{source}.

In Table 4, the quality of the GAN data degrades noticeably after the additional training with only L_{class}. Unrealistic artifacts are evident in the inferior images, but the Inception Scores of those images are even higher than those of the well-trained images. In contrast, the Compression Score decreases in accordance with our expectations as the GAN images become evidently worse.

REFERENCES

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pp. 2654–2662, 2014.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5:4308, 2014.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541. ACM, 2006.

Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pp. 2285–2294, 2015.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*, 2017.

François Chollet et al. Keras. https://keras.io, 2015.

Mark Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*, pp. 24–30, 1996.

Misha Denil, Babak Shakibi, Laurent Dinh, Nando De Freitas, et al. Predicting parameters in deep learning. In *Advances in neural information processing systems*, pp. 2148–2156, 2013.

Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pp. 1135–1143, 2015.

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

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yu Liu, Hantian Zhang, Luyuan Zeng, Wentao Wu, and Ce Zhang. Mlbench: How good are machine learning clouds for binary classification tasks on structured data. *ArXiv e-prints*, 2017.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*, 2016.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pp. 2852–2858, 2017.