Synthetic Learning: Learn From Distributed Asynchronized Discriminator GAN Without Sharing Medical Image Data

Qi Chang[1]*, Hui Qu[1]*, Yikai Zhang[1]*, Mert Sabuncu[2],
Chao Chen[3], Tong Zhang[4] and Dimitris Metaxas[2]
[1]Rutgers University [2]Cornell University
[3]Stony Brook University [4]Hong Kong University of Science and Technology
{qc58,hq43,yz422,dnm}@cs.rutgers.edu, msabuncu@cornell.edu,
chao.chen.cchen@gmail.com, tongzhang@tongzhang-ml.org

Abstract

In this paper, we propose a data privacy-preserving and communication efficient distributed GAN learning framework named Distributed Asynchronized Discriminator GAN (AsynDGAN). Our proposed framework aims to train a central generator learns from distributed discriminator, and use the generated synthetic image solely to train the segmentation model. We validate the proposed framework on the application of health entities learning problem *which is known to be privacy sensitive. Our experiments show that our approach: 1) could learn the real image's distribution from multiple datasets without sharing the patient's raw data. 2) is more efficient and requires lower bandwidth than other distributed deep learning methods. 3) achieves higher performance compared to the model trained by one real dataset, and almost the same performance compared to the model trained by all real datasets. 4) has provable guarantees that the generator could learn the distributed distribution in an* all important *fashion thus is unbiased.We release our AsynDGAN source code at: https://github.com/tommy-qichang/AsynDGAN*

1. Introduction

1.1. The privacy policies and challenges in medical intelligence

The privacy issue, while important in every domain, is enforced vigorously for medical data. Multiple level of regulations such as HIPAA [2, 11, 36, 13] and the approval process for the Institutional Review Board (IRB) [6] protect the patients' sensitive data from malicious copy or even tamper evidence of medical conditions [38]. Like a double-edge sword, these regulations objectively cause insufficient collaborations in health records. For instance, America, European Union and many other countries do not allow patient data leave their country [25, 47]. As a result, many hospitals and research institutions are wary of cloud platforms and prefer to use their own server. Even if in the same country the medical data collaborate still face a big hurdle.

1.2. The restriction of the medical data accessibility

It's widely known that sufficient data volume is necessary for training a successful machine learning algorithm [10] for medical image analysis. However, due to the policies and challenges mentioned above, it is hard to acquire enough medical scans for training a machine learning model. In 2016, there were approximately 38 million MRI scans and 79 million CT scans performed in the United States [41]. Even so, the available datasets for machine learning research are still very limited: the largest set of medical image data available to public is 32 thousand [51] CT images, only 0.02% of the annual acquired images in the United States. In contrast, the ImageNet [9] project, which is the large visual dataset designed for use in visual object recognition research, has more than 14 million images that have been annotated in more than 20,000 categories.

1.3. Learning from synthetic images: a solution

In this work, we design a framework using centralized generator and distributed discriminators to learn the generative distribution of target dataset. In the health entities learning context, our proposed framework can aggregate datasets from multiple hospitals to obtain a faithful estimation of the overall distribution. The specific task (e.g., segmentation and classification) can be accomplished locally by acquiring data from the generator. Learning from synthetic images has several advantages:

Privacy mechanism: The central generator has limited information for the raw images in each hospital. When the generator communicates with discriminators in hospitals,

*equal contribution

only information about the synthetic image is transmitted. Such a mechanism prohibits the central generator's direct access to raw data thus secures privacy.

Synthetic data sharing: The natural of synthetic data allows the generator to share the synthetic images without restriction. Such aggregation and redistribution system can build a public accessible and faithful medical database. The inexhaustible database can benefit researchers, practitioners and boost the development of medical intelligence.

Adaptivity to architecture updates: The machine learning architecture evolves rapidly to achieve a better performance by novel loss functions [48, 17], network modules [18, 45, 37, 42] or optimizers [46, 54, 32, 56, 57]. We could reasonably infer that the recently well-trained model may be outdated or underperformed in the future as new architectures invented. Since the private-sensitive data may be not always accessible, even if we trained a model based on these datasets, we couldn't embrace new architectures to achieve higher performance. Instead of training a task-specific model, our proposed method trains a generator that learns from distributed discriminators. Specifically, we learn the distribution of private datasets by a generator to produce synthetic images for future use, without worrying about the lost of the proprietary datasets.

To the best of our knowledge, we are the first to use GAN to address the medical privacy problem. Briefly, our contributions lie in three folds: (1) A distributed asynchronized discriminator GAN (AsynDGAN) is proposed to learn the real images' distribution without sharing patients' raw data from different datasets. (2) AsynDGAN achieves higher performance than models that learn from real images of only one dataset. (3) AsynDGAN achieves almost the same performance as the model that learns from real images of all datasets.

2. Related Work

2.1. Generative Adversarial Networks (GANs)

The Generative Adversarial Nets [12] have achieved great success in various applications, such as natural image synthesis [43, 55, 8], image style translation [22, 58], image super resolution [29] in computer vision, and medical image segmentation [52, 50], cross-modality image synthesis [40], image reconstruction [53] in medical image analysis. The GAN estimates generative distribution via an adversarial supervisor. Specifically, the generator G attempts to imitate the data from target distribution to make the 'fake' data indistinguishable to the adversarial supervisor D. In AsynDGAN framework, we mainly focus on the conditional distribution estimation due to the nature of health entities learning problems. However, the AsynDGAN framework can be easily adopted into general GAN learning tasks.

	Privacy Mechanism	Data transmission	Adaptivity
FL	Randon Noise	Parameters / Gradients	No
SL	Data Block	Cut Layer Gradients	No
AsynDGAN	Data Block	Fake Data, Auxiliary Variable & Discriminator Loss	Yes

Table 1. Comparison between different learning strategies.

2.2. Learning with data privacy

Federated Learning: The federated learning (FL) seeks to collaborate local nodes in the network to learn a globally powerful model without storing data in the cloud. Recently, FL attracts more attention as data privacy becomes a concern for users [14, 27, 7, 19]. Instead of directly exposing users' data, FL only communicates model information (parameters, gradients) with privacy mechanism so protects users' personal information. In [1, 23, 34], the SGD is shared in a privacy protection fashion. However, communicating gradients is dimension dependent. Considering a ResNet101 [15] with $d = 40$ million parameters, it requires at least 170 mb to pass gradients for each client per-iteration. Even with compression technique similar to [1], the communication cost is still non-affordable for large-size networks.

Split Learning: The split learning (SL) [49] separates shallow and deep layers in deep learning models. The central processor only maintains layers that are several blocks away from the local input, and only inter-layer information is transmitted from local to central. In this way, the privacy is guaranteed because the central processor has no direct access to data. It reduces the communication cost from model-dependent level to cut-layer-dependent layer while protecting data privacy. However, such method does not apply to neural networks with skip connections, e.g., ResNets [15].

In AsynDGAN framework, the communication cost in each iteration is free of the dimension d. Only auxiliary data (label and masks), 'fake' data and discriminator loss are passed between the central processor and local nodes in the network. For a 128×128 size gray-scale image, communication cost per-iteration for each node is 8 mb with batch size 128. Since the central processor has only access to discriminator and auxiliary data, the privacy of client is secured via separating block mechanism.

In addition, adaptivity is an exclusive advantage of AsynDGAN framework. With rapid evolution of machine learning methods, practitioners need to keep updated with state-of-the-art methods. However, there will be a high transaction cost to train a new model in a classical distributed learning subroutine. With the AsynDGAN system, one can maintain the generative distribution. Therefore, updating machine learning models can be done locally with the freedom of generating training data. The comparison between FL, SL and our AsynDGAN is shown in Table 1.

3. Method

3.1. Overview

Our proposed AsynDGAN is comprised of one central generator and multiple distributed discriminators located in different medical entities. In the following, we present the network architecture, object function and then analysis the procedure of the distributed asynchronized optimization.

3.2. Network architecture

An overview of the proposed architecture is shown in Figure 1. The central generator, denoted as G, takes task-specific inputs (segmentation masks in our experiments) and generates synthetic images to fool the discriminators. The local discriminators, denote as D^1 to D^n, learn to differentiate between the local real images and the synthetic images from G. Due to the sensitivity of patients' images, the real images in each medical center may not be accessed from outside. Our architecture is naturally capable of avoiding such limitation because only the specific discriminator in the same medical entity needs to access the real images. In this way, the real images in local medical entities will be kept privately. Only synthetic images, masks, and gradients are needed to be transferred between the central generator and the medical entities.

The generator will learn the joint distribution from different datasets that belong to different medical entities. Then it can be used as an image provider to train a specific task, because we expect the synthetic images to share the same or similar distribution as the real images. In the experiments, we apply the AsynDGAN framework to segmentation tasks to illustrate its effectiveness. The U-Net [45] is used as the segmentation model, and details about G and Ds designed for segmentation tasks are described below.

3.2.1 Central generator

For segmentation tasks, the central generator is an encoder-decoder network that consists of two stride-2 convolutions (for downsampling), nine residual blocks [15], and two transposed convolutions. All non-residual convolutional layers are followed by batch normalization [20] and the ReLU activation. All convolutional layers use 3×3 kernels except the first and last layers that use 7×7 kernels.

3.2.2 Distributed discriminators

In the AsynDGAN framework, the discriminators are distributed over N nodes (hospitals, mobile devices). Each discriminator D_j only has access to data stored in the j-th node thus discriminators are trained in an asynchronized fashion. For segmentation, each discriminator has the same structure as that in PatchGAN [21]. The discriminator individually quantifies the fake or real value of different small patches in the image. Such architecture assumes patch-wise independence of pixels in a Markov random field fashion [30, 22], and can capture the difference in geometrical structures such as background and tumors.

3.3. Objective of AsynDGAN

The AsynDGAN is based on the conditional GAN [39]. The objective of a classical conditional GAN is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim s(x)}\mathbb{E}_{y\sim p_{data}(y|x)}[\log D(y|x)] + \mathbb{E}_{\hat{y}\sim p_{\hat{y}}(\hat{y}|x)}[\log(1-D(\hat{y}|x))] \tag{1}$$

where D represents the discriminator and G is the generator. G aims to approximate the conditional distribution $p_{data}(y|x)$ so that D can not tell if the data is 'fake' or not. The hidden variable x is an auxiliary variable to control the mode of generated data [39]. In reality, x is usually a class label or a mask that can provide information about the data to be generated. Following previous works ([33, 21]), instead of providing Gaussian noise z as an input to the generator, we provide the noise only in the form of dropout, which applied to several layers of the generator of AsynDGAN at both training and test time.

In the AsynDGAN framework, the generator is supervised by N different discriminators. Each discriminator is associated with a subset of datasets. It is natural to quantify such a setting using a mixture distribution on auxiliary variable x. In another word, instead of given a naive $s(x)$, the distributions of x becomes $s(x) = \sum_{j\in[N]} \pi_j s_j(x)$. For each sub-distribution, there is a corresponding discriminator D_j which only receives data generated from prior $s_j(x)$. Therefore, the loss function of our AsynDGAN becomes:

$$\begin{aligned}&\min_G \max_{D_1:D_N} V(D_{1:N}, G)\\ &= \sum_{j\in[N]} \pi_j \{\mathbb{E}_{x\sim s_j(x)}\mathbb{E}_{y\sim p_{data}(y|x)}[\log D_j(y|x)]\\ &+ \mathbb{E}_{\hat{y}\sim p_{\hat{y}}(\hat{y}|x)}[\log(1-D_j(\hat{y}|x))]\}\end{aligned} \tag{2}$$

3.4. Optimization process

The optimization process of the AsynDGAN is shown in Figure 2. In each iteration, a randomly sampled tuple (x, y) is provided to the system. Here, x denotes the input label which observed by the generator, and y is the real image only accessible by medical entities. Then the network blocks are updated iteratively in the following order:

1) D-update: Calculating the adversarial loss for j-th discriminator D_j and update D_j, where $j = 1, 2, \cdots, N$.

2) G-update: After updating all discriminators, G will be updated using the adversarial loss $\sum_{j=1}^{N} loss(D_j)$.

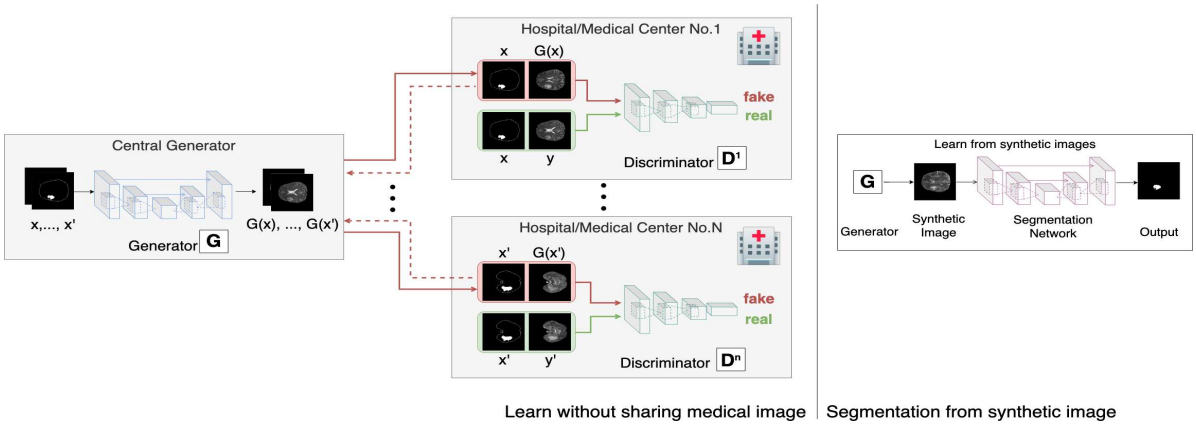


Figure 1. The overall structure of AsynDGAN. It contains two parts, a central generator G and multiple distributed discriminators $D^1, D^2, \cdots, D^n$ in each medical entity. G takes a task-specific input (segmentation masks in our experiments) and output synthetic images. Each discriminator learns to differentiate between the real images of current medical entity and synthetic images from G. The well-trained G is then used as an image provider to train a task-specific model (segmentation in our experiments).

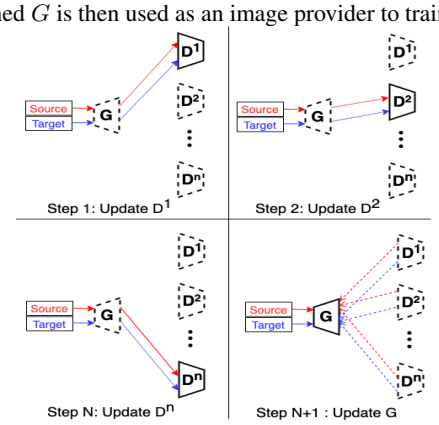


Figure 2. The optimization process of AsynDGAN. The solid arrows show the forward pass, and the dotted arrows show gradient flow during the backward pass of our iterative update procedure. The solid block indicate that it is being updated while the dotted blocks mean that they are frozen during that update step. Red and blue rectangles are source mask and target real image, respectively.

This process is formulated as Algorithm 1. We apply the cross entropy loss and in the algorithm and further analyze the AsynDGAN framework in this setting. We stress that the framework is general and can be collaborated with variants of GAN loss including Wasserstein distance and classical regression loss [3, 31].

3.5. Analysis: AsynDGAN learns the correct distribution

In this section, we present a theoretical analysis of AsynDGAN and discuss the implications of the results. We first begin with a technical lemma describing the optimal strategy of the discriminator.

Lemma 1. *When generator G is fixed, the optimal discriminator $D_j(y|x)$ is :*

$$D_j(y|x) = \frac{p(y|x)}{p(y|x) + q(y|x)} \tag{3}$$

Suppose in each training step the discriminator achieves its maxima criterion in Lemma 1, the loss function for the generator becomes:

$$\begin{aligned}\min_G V(G) &= \mathbb{E}_y \mathbb{E}_{x\sim p_{data}(y|x)}[\log D(y|x)] \\ &+ \mathbb{E}_{\hat{y}\sim p_{\hat{y}}(\hat{y}|x)}[\log(1 - D(\hat{y}|x))] \\ &= \sum_{j\in[N]} \pi_j \int_y s_j(x) \int_x p(y|x) \log \frac{p(y|x)}{p(y|x) + q(y|x)} \\ &+ q(y|x) \log \frac{q(y|x)}{p(y|x) + q(y|x)} dxdy\end{aligned}$$

Assuming in each step, the discriminator always performs optimally, we show indeed the generative distribution G seeks to minimize the loss by approximating the underlying distribution of data.

Theorem 1. *Suppose the discriminators $D_{1\sim N}$ always behave optimally (denoted as $D^*_{1\sim N}$), the loss function of generator is global optimal iff $q(y, x) = p(y, x)$ where the optimal value of $V(G, D^*_{1\sim N})$ is $-\log 4$.*

Remark 1. While analysis of AsynDGAN loss shares similar spirit with [12], it has different implications. In the distributed learning setting, data from different nodes are often dissimilar. Consider the case where $\Omega(s_j(x)) \cap \Omega(s_k(y)) = \emptyset$, for $k \neq j$, the information for $p(y|x), y \in \Omega(s_j(x))$ will be missing if we lose the j-th node. The behavior of trained generative model is unpredictable when receiving auxiliary variables from unobserved distribution $s_j(x)$. The AsynDGAN framework provides a solution for unifying different datasets by collaborating multiple discriminators.

Algorithm 1 Training algorithm of AsynDGAN.

for number of total training iterations **do**
  for number of interations to train discriminator **do**
    for each node $j \in [N]$ **do**
      – Sample minibatch of of m auxiliary variables $\{x_1^j, ..., x_m^j\}$ from $s_j(x)$ and send to generator G.

      – Generate m fake data from generator G, $\{\hat{y}_1^j, ..., \hat{y}_m^j\} \sim q(\hat{y}|x)$ and send to node j.
      – Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_{D_j}} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_j(y_i^j) + \log(1 - D_j(G(\hat{y}_i^j))) \right].$$

    end for
  end for
  for each node $j \in [N]$ **do**
    – Sample minibatch of m auxiliary variables $\{x_1^j, ..., x_m^j\}$ from $s_j(x)$ and send to generator G.

    – Generate corresponding m fake data from generator G, $\{\hat{y}_1^j, ..., \hat{y}_m^j\} \sim q(\hat{y}|x)$ and send to node j.

    – Discriminator D_j passes error to generator G.
  end for
  – Update G by descending its stochastic gradient:

$$\nabla_{\theta_G} \frac{1}{Nm} \sum_{j=1}^{N} \sum_{i=1}^{m} \log(1 - D_j(G(\hat{y}_i^j))).$$

end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

4. Experiments

In this section, we first perform experiments on a synthetic dataset to illustrate how AsynDGAN learns a mixed Gaussian distribution from different subsets, and then apply AsynDGAN to the brain tumor segmentation task on BraTS2018 dataset [5] and nuclei segmentation task on Multi-Organ dataset [28].

4.1. Datasets and evaluation metrics

4.1.1 Datasets

Synthetic dataset The synthetic dataset is generated by mixing 3 one-dimensional Gaussian. In another word, we generate $x \in \{1, 2, 3\}$ with equal probabilities. Given x, the random variable y is generated from $y = y_1 \mathbf{1}_{x=1} + y_2 \mathbf{1}_{x=2} + y_3 \mathbf{1}_{x=3}$ where $\mathbf{1}_{event}$ is the indicator function and $y_1 \sim \mathcal{N}(-3, 2), y_2 \sim \mathcal{N}(1, 1), y_3 \sim \mathcal{N}(3, 0.5)$. Suppose the generator learns the conditional distribution of y: $p(y|x)$ perfectly, the histogram should behave similarly to the shape of the histogram of mixture gaussian.

BraTS2018 This dataset comes from the Multimodal Brain Tumor Segmentation Challenge 2018 [4, 5, 35] and contains multi-parametric magnetic resonance imaging (mpMRI) scans of low-grade glioma (LGG) and high-grade glioma (HGG) patients. There are 210 HGG and 75 LGG cases in the training data, and each case has four types of MRI scans and three types of tumor subregion labels. In our experiments, we perform 2D segmentation on T2 images of the HGG cases to extract the whole tumor regions. The 2D slices with tumor areas smaller than 10 pixels are excluded for both GAN training and segmentation phases. In the GAN synthesis phase, all three labels are utilized to generate fake images. For segmentation, we focus on the whole tumor (regions with any of the three labels).

Multi-Organ This dataset is proposed by Kumar et al. [28] for nuclei segmentation. There are 30 histopathology images of size 1000×1000 from 7 different organs. The train set contains 16 images of breast, liver, kidney and prostate (4 images per organ). The same organ test set contains 8 images of the above four organs (2 images per organ) while the different organ test set has 6 images from bladder, colon and stomach. In our experiments, we focus on the four organs that exist both in the train and test sets, and perform color normalization [44] for all images. Two training images of each organ is treated as a subset that belongs to a medical entity.

4.1.2 Evaluation metrics

We adopt the same metrics in the BraTS2018 Challenge [5] to evaluate the segmentation performance of brain tumor: Dice score (Dice), sensitivity (Sens), specificity (Spec), and 95% quantile of Hausdorff distance (HD95). The Dice score, sensitivity (true positive rate) and specificity (true negative rate) measure the overlap between ground-truth mask G and segmented result S. They are defined as

$$Dice(G, S) = \frac{2|G \cap S|}{|G| + |S|} \tag{4}$$

$$Sens(G, S) = \frac{|G \cap S|}{|G|} \tag{5}$$

$$Spec(G, S) = \frac{|(1 - G) \cap (1 - S)|}{|1 - G|} \tag{6}$$

The Hausdorff distance evaluates the distance between boundaries of ground-truth and segmented masks:

$$HD(G, S) = \max\{ \sup_{x \in \partial G} \inf_{y \in \partial S} d(x, y), \sup_{y \in \partial S} \inf_{x \in \partial G} d(x, y) \} \tag{7}$$

where ∂ means the boundary operation, and d is Euclidean distance. Because the Hausdorff distance is sensitive to small outlying subregions, we use the 95% quantile of the distances instead of the maximum as in [5]. To simplify the problem while fairly compare each experiment, we choose 2D rather than 3D segmentation task for the BraTS2018 Challenge and compute these metrics on each 2D slices and take an average on all 2D slices in the test set.

For nuclei segmentation, we utilize the Dice score and the Aggregated Jaccard Index (AJI) [28]:

$$AJI = \frac{\sum_{i=1}^{n_G} |G_i \cap S(G_i)|}{\sum_{i=1}^{n_G} |G_i \cup S(G_i)| + \sum_{k \in K} |S_k|} \tag{8}$$

where $S(G_i)$ is the segmented object that has maximum overlap with G_i with regard to Jaccard index, K is the set containing segmentation objects that have not been assigned to any ground-truth object.

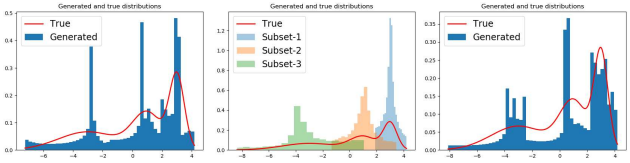
4.2. Implementation details

In the synthetic learning phase, we use 9-blocks ResNet [16] architecture for the generator, and multiple discriminators which have the same structure as that in PatchGAN [21] with patch size 70×70. We resize the input image as 286×286 and then randomly crop the image to 256×256. In addition to the GAN loss and the L1 loss, we also used perceptual loss as described in [24]. We use mini-batch SGD and apply the Adam solver [26], with a learning rate of 0.0002, and momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. The batch size we used in AsynDGAN depends on the number of discriminators. We use batch size 3 and 1 for BraTS2018 dataset and Multi-Organ dataset, respectively.

In the segmentation phase, we randomly crop images of 224×224 with a batch size of 16 as input. The model is trained with Adam optimizer using a learning rate of 0.001 for 50 epochs in brain tumor segmentation and 100 epochs in nuclei segmentation. To improve performance, we use data augmentation in all experiments, including random horizontal flip and rotation in tumor segmentation and additional random scale and affine transformation in nuclei segmentation.

4.3. Experiment on synthetic dataset

In this subsection, we show that the proposed synthetic learning framework can learn a mixture of Gaussian distribution from different subsets. We compare the quality of learning distribution in 3 settings: (1) **Syn-All.** Training a regular GAN using all samples in the dataset. (2) **Syn-Subset-n.** Training a regular GAN using only samples in local subset n, where $n \in \{1, 2, 3\}$. (3) **AsynDGAN.** Training our AsynDGAN using samples in all subsets in a distributed fashion.



(a) Syn-All (b) Syn-Subset-n (c) AsynDGAN

Figure 3. Generated distributions of different methods.

The learned distributions are shown in Figure 3. In particular, any local learning (indicated in Figure 3(b)) can only fit one mode Gaussian due to the restriction of local information while AsynDGAN is able to capture global information thus has a comparable performance with the regular GAN using the union of separated datasets (**Syn-All**).

4.4. Brain tumor segmentation

In this subsection, we show that our AsynDGAN can work well when there are patients' data of the same disease in different medical entities.

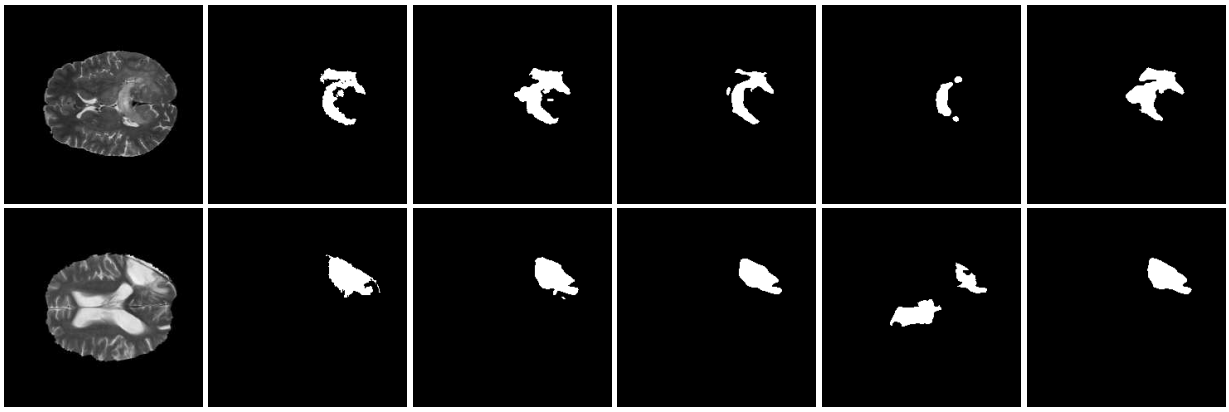
4.4.1 Settings

There are 210 HGG cases in the training data. Because we have no access to the test data of the BraTS2018 Challenge, we split the 210 cases into train (170 cases) and test (40 cases) sets. The train set is then sorted according to the tumor size and divided into 10 subsets equally, which are treated as data in 10 distributed medical entities. There are 11,057 images in the train set and 2,616 images in the test set. We conduct the following segmentation experiments: (1) **Real-All.** Training using real images from the whole train set (170 cases). (2) **Real-Subset-n.** Training using real images from the n-th subset (medical entity), where $n = 1, 2, \cdots, 10$. There are 10 different experiments in this category. (3) **Syn-All.** Training using synthetic images generated from a regular GAN. The GAN is trained directly using all real images from the 170 cases. (4) **AsynDGAN.** Training using synthetic images from our proposed AsynDGAN. The AsynDGAN is trained using images from the 10 subsets (medical entities) in a distributed fashion.

In all experiments, the test set remains the same for fair comparison. It should be noted that in the **Syn-All** and **AsynDGAN** experiments, the number of synthetic images are the same as that of real images in **Real-All**. The regular GAN has the same generator and discriminator structures as AsynDGAN, as well as the hyper-parameters. The only difference is that AsynDGAN has 10 different discriminators, and each of them is located in a medical entity and only has access to the real images in one subset.

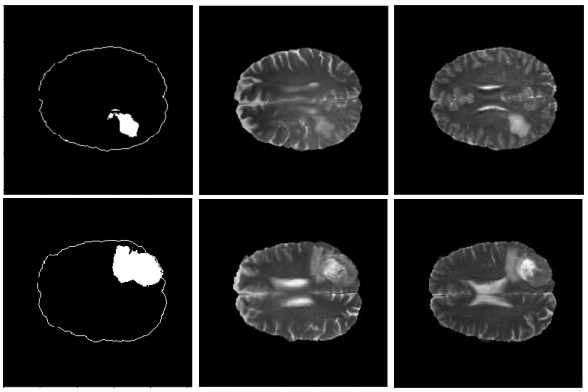
4.4.2 Results

The quantitative brain tumor segmentation results are shown in Table 2. The model trained using all real images



(a) Image (b) Label (c) Real-All (d) Syn-All (e) Real-Subset-6 (f) AsynDGAN

Figure 4. Typical brain tumor segmentation results. (a) Test images. (b) Ground-truth labels of tumor region. (c)-(f) are results of models trained on all real images, synthetic images of regular GAN, real images from subset-6, synthetic images of AsynDGAN, respectively.



(a) Input (b) AsynDGAN (c) Real

Figure 5. The examples of synthetic brain tumor images from the AsynDGAN. (a) The input of the AsynDGAN network. (b) Synthetic images of AsynDGAN based on the input. (c) Real images.

Method	Dice ↑	Sens ↑	Spec ↑	HD95 ↓
Real-All	0.7485	0.7983	0.9955	12.85
Real-Subset-1	0.5647	0.5766	0.9945	26.90
Real-Subset-2	0.6158	0.6333	0.9941	21.87
Real-Subset-3	0.6660	0.7008	0.9950	21.90
Real-Subset-4	0.6539	0.6600	0.9962	21.07
Real-Subset-5	0.6352	0.6437	0.9956	19.27
Real-Subset-6	0.6844	0.7249	0.9935	21.10
Real-Subset-7	0.6463	0.6252	0.9972	15.60
Real-Subset-8	0.6661	0.6876	0.9957	18.16
Real-Subset-9	0.6844	0.7088	0.9953	18.56
Real-Subset-10	0.6507	0.6596	0.9957	17.33
Syn-All	0.7114	0.7099	0.9969	16.22
AsynDGAN	0.7043	0.7295	0.9957	14.94

Table 2. Brain tumor segmentation results.

(**Real-All**) is the ideal case that we can access all data. It is our baseline and achieves the best performance. Compared with the ideal baseline, the performance of models trained using data in each medical entity (**Real-Subset-1∼10**) degrades a lot, because the information in each subset is limited and the number of training images is much smaller.

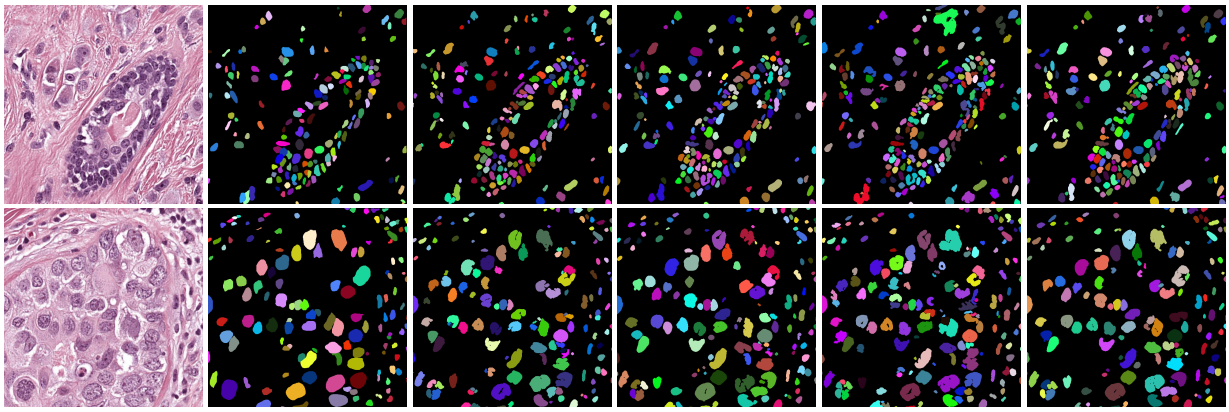
Our AsynDGAN can learn from the information of all data during training, although the generator doesn't "see" the real images. And we can generate as many synthetic images as we want to train the segmentation model. Therefore, the model (**AsynDGAN**) outperforms all models using single subset. For reference, we also report the results using synthetic images from regular GAN (**Syn-All**), which is trained directly using all real images. The AsynDGAN has the same performance as the regular GAN, but has no privacy issue because it doesn't collect real image data from medical entities. The examples of synthetic images from AysnDGAN are shown in Figure 5. Several qualitative segmentation results of each method are shown in Figure 4.

4.5. Nuclei segmentation

In this subsection, we apply the AsynDGAN to multiple organ nuclei segmentation and show that our method is effective to learn the nuclear features of different organs.

4.5.1 Settings

We assume that the training images belong to four different medical entities and each entity has four images of one organ. Similar to Section 4.4, we conduct the following experiments: (1) **Real-All.** Training using the 16 real images of the train set. (2) **Real-Subset-n.** Training using 4 real images from each subset (medical entity), where $n \in \{\text{breast, liver, kidney, prostate}\}$. (3) **Syn-All.** Training using synthetic images from regular GAN, which is trained using all 16 real images. (4) **AsynDGAN.** Training using synthetic images from the AsynDGAN, which is trained using images from the 4 subsets distributively. In all above experiments, we use the same organ test set for evaluation.

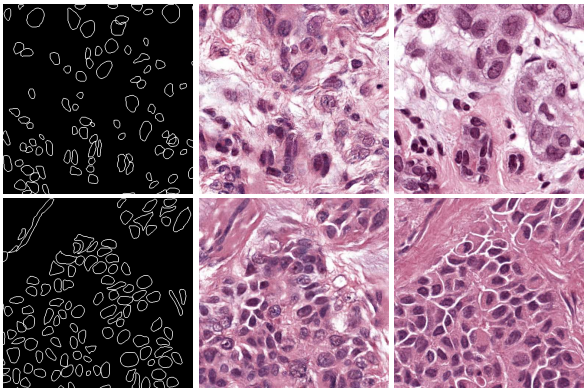


(a) Image (b) Label (c) Real-All (d) Syn-All (e) subset-prostate (f) AsynDGAN

Figure 6. Typical nuclei segmentation results. (a) Test images. (b) Ground-truth labels of nuclei. (c)-(f) are results of models trained on all real images, synthetic images of regular GAN, real images from prostate, synthetic images of AsynDGAN, respectively. Distinct colors indicate different nuclei.

Method	Dice ↑	AJI ↑
Real-All	0.7833	0.5608
Real-Subset-breast	0.7340	0.4942
Real-Subset-liver	0.7639	0.5191
Real-Subset-kidney	0.7416	0.4848
Real-Subset-prostate	0.7704	0.5370
Syn-All	0.7856	0.5561
AsynDGAN	0.7930	0.5608

Table 3. Nuclei segmentation results.



(a) Input (b) AsynDGAN (c) Real

Figure 7. The examples of synthetic nuclei images from the AsynDGAN. (a) The input of the AsynDGAN network. (b) Synthetic images of AsynDGAN based on the input. (c) Real images.

4.5.2 Results

The quantitative nuclei segmentation results are presented in Table 3. Compared with models using single organ data, our method achieves the best performance. The reason is that local models cannot learn the nuclear features of other organs. Compared with the model using all real images, the AsynDGAN has the same performance, which proves the effectiveness of our method in this type of tasks. The result using regular GAN (**Syn-All**) is slightly worse than ours, probably because one discriminator is not good enough to capture different distributions of nuclear features in multiple organs. In AsynDGAN, each discriminator is responsible for one type of nuclei, which may be better for the generator to learn the overall distribution. We present several examples of synthetic images from AsynDGAN in Figure 7, and typical qualitative segmentation results in Figure 6.

5. Conclusion

In this work, we proposed a distributed GAN learning framework as a solution to the privacy restriction problem in multiple health entities. Our proposed framework applies GAN to aggregate and learns the overall distribution of datasets in different health entities without direct access to patients' data. The well-trained generator can be used as an image provider for training task-specific models, without accessing or storing private patients' data. Our evaluation on different datasets shows that our training framework could learn the real image's distribution from distributed datasets without sharing the patient's raw data. In addition, the task-specific model trained solely by synthetic data has a competitive performance with the model trained by all real data, and outperforms models trained by local data in each medical entity.

Acknowledgements

We thank anonymous reviewers for helpful comments. This work was partially supported by ARO-MURI-68985NSMUR, NSF-1909038, NSF-1855759, NSF-1855760, NSF-1733843, NSF-1763523, NSF-1747778 and NSF-1703883.

References

[1] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

[2] George J Annas et al. Hipaa regulations-a new era of medical-record privacy? *New England Journal of Medicine*, 348(15):1486–1490, 2003.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[4] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4:170117, 2017.

[5] Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin Rozycki, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.

[6] Elizabeth A Bankert and Robert J Amdur. *Institutional review board: Management and function*. Jones & Bartlett Learning, 2006.

[7] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

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

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Pedro M Domingos. A few useful things to know about machine learning. *Commun. acm*, 55(10):78–87, 2012.

[11] Centers for Disease Control, Prevention, et al. Hipaa privacy rule and public health. guidance from cdc and the us department of health and human services. *MMWR: Morbidity and mortality weekly report*, 52(Suppl. 1):1–17, 2003.

[12] 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*, pages 2672–2680, 2014.

[13] Lawrence O Gostin, Laura A Levit, Sharyl J Nass, et al. *Beyond the HIPAA privacy rule: enhancing privacy, improving health through research*. National Academies Press, 2009.

[14] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Julian Hochberg. Depth perception loss with local monocular suppression: A problem in the explanation of stereopsis. *Science*, 145(3638):1334–1336, 1964.

[18] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. 2016.

[19] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. Loadaboost: Loss-based adaboost federated machine learning on medical data. *arXiv preprint arXiv:1811.12629*, 2018.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation with Conditional Adversarial Networks. 2016.

[22] 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.

[23] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Advances in Neural Information Processing Systems*, pages 6343–6354, 2018.

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[25] Melita Sogomonjan1 Tanel Kerikmäe. Challenges in the e-health regulatory policy. *HOW DEEP IS YOUR LAW? BREXIT. TECHNOLOGIES. MODERN CONFLICTS*, 367, 2017.

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[28] Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE transactions on medical imaging*, 36(7):1550–1560, 2017.

[29] 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.

[30] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[31] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[32] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean. Boosting algorithms as gradient descent. In *Advances in neural information processing systems*, pages 512–518, 2000.

[33] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[34] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[35] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[36] Rebecca T Mercuri. The hipaa-potamus in health care data security. *Communications of the ACM*, 47(7):25–28, 2004.

[37] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.

[38] Yisroel Mirsky, Tom Mahler, Ilan Shelef, and Yuval Elovici. Ct-gan: Malicious tampering of 3d medical imagery using deep learning. *arXiv preprint arXiv:1901.03597*, 2019.

[39] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. 2014.

[40] Dong Nie, Roger Trullo, Jun Lian, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical image synthesis with context-aware generative adversarial networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 417–425. Springer, 2017.

[41] Irene Papanicolas, Liana R Woskie, and Ashish K Jha. Health care spending in the united states and other high-income countries. *Jama*, 319(10):1024–1039, 2018.

[42] Hui Qu, Zhennan Yan, Gregory M Riedlinger, Subhajyoti De, and Dimitris N Metaxas. Improving nuclei/gland instance segmentation in histopathology images by full resolution neural network and spatial constrained loss. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 378–386. Springer, 2019.

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

[44] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[46] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[47] Jonathan JM Seddon and Wendy L Currie. Cloud computing and trans-border health data: Unpacking us and eu healthcare regulation and compliance. *Health policy and technology*, 2(4):229–241, 2013.

[48] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.

[49] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. 2018.

[50] Yuan Xue, Tao Xu, Han Zhang, L Rodney Long, and Xiaolei Huang. Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. *Neuroinformatics*, 16(3-4):383–392, 2018.

[51] Ke Yan, Xiaosong Wang, Le Lu, and Ronald M Summers. Deeplesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. *Journal of Medical Imaging*, 5(3):036501, 2018.

[52] Dong Yang, Daguang Xu, S Kevin Zhou, Bogdan Georgescu, Mingqing Chen, Sasa Grbic, Dimitris Metaxas, and Dorin Comaniciu. Automatic liver segmentation using an adversarial image-to-image network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 507–515. Springer, 2017.

[53] Guang Yang, Simiao Yu, Hao Dong, Greg Slabaugh, Pier Luigi Dragotti, Xujiong Ye, Fangde Liu, Simon Arridge, Jennifer Keegan, Yike Guo, et al. Dagan: deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction. *IEEE transactions on medical imaging*, 37(6):1310–1321, 2017.

[54] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[55] 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.

[56] Yikai Zhang, Hui Qu, Chao Chen, and Dimitris Metaxas. Taming the noisy gradient: train deep neural networks with small batch sizes. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4348–4354. AAAI Press, 2019.

[57] Yikai Zhang, Hui Qu, and Dimitris Metaxas1 Chao Chen. Local regularizer improves generalization. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. AAAI, 2020.

[58] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.