MetaCGAN: A Novel GAN Model for Generating High Quality and Diversity Images with Few Training Data

Ying Ma, Guoqiang Zhong*, Yanan Wang
Department of Computer Science and Technology,
Ocean University of China,
Qingdao 266100, China
mayinger618@gmail.com, gqzhong@ouc.edu.cn, 2964180578@qq.com,

Wen Liu
Qingdao Yisa Data Technology Company, Ltd,
Qingdao 266100, China
l1uw3n@gmail.com

***Abstract*—Given a large amount of data in the base classes and a small number of data in the new classes, meta-learning can learn prior experience from the base classes and transfer knowledge to the new classes by generating network parameters for data generation. In this paper, we propose a novel generative adversarial network called MetaCGAN for generating high quality and diversity images to achieve data augmentation for the new classes with few data. In particular, MetaCGAN consists of two modules, the conditional GAN (CGAN) and MetaNet modules. The CGAN module is our skeleton network that is applied to generate images, while the MetaNet module is our auxiliary network that is applied to provide deconvolutional weights for the generator of CGAN. Experimental results on the MNIST, Fashion MNIST and CelebA data sets demonstrate the superiority of MetaCGAN over baseline models. Both qualitative and quantitative results show that the MetaNet module can learn prior knowledge and transfer it from the base classes to the new classes, which is beneficial for generating high quality and diversity images to the new classes with few images.**

***Index Terms*—MetaCGAN, meta-learning, prior experience, CGAN, MetaNet**

I. INTRODUCTION

In recent years, since generative adversarial network (GAN) can learn data distribution and generate new data according to the distribution, it becomes a research hotspot in the deep learning area. In general, however, to generate high quality images, GAN needs a large amount of training data. When the training samples are scarce, it is a very challenging task to generate high quality images using GAN. Therefore, a critical issue we confront is how to generate high quality and diversity images with few data.

One method to tackle this issue is to use the idea of transfer learning. A model is first trained on the source data set that has large scale of samples and similar distribution with the target data set, and then, the trained model is fine-tuned to adapt to the target data set that has a small number of samples. However, in this case, the fine-tuned model may forget how to generate source samples, which is undesirable. It should be able to generate both source and target samples simultaneously.

Inspired by humans' ability that one can transfer past experience to new tasks, employing past experience to learn new tasks and still remembering past experience, meta-learning is proposed to learn how to learn, which has been widely used in few-shot learning, transferring knowledge from the support sets to the query sets with few samples [1] [2].

In order to address the problem that GAN cannot perform well on scarce training data, we propose a novel model named MetaCGAN that applies a MetaNet module to the generator of the CGAN framework. In particular, the MetaNet module can learn the prior knowledge with a large amount of data in the base classes and transfer knowledge to the new classes with a small number of data, which encourages the generator of MetaCGAN to perform well on generating high quality and diversity images for the new classes.

In summary, our contributions are as follows:

- We propose MetaCGAN, which adds a MetaNet module to the generator of CGAN to learn the prior experience on the base classes and transfer knowledge to the new classes.
- The MetaNet module can generate weights for the generator of MetaCGAN, which enhances MetaCGAN to generate high quality and diversity images with few data in the new classes.
- Qualitative and quantitative experimental results on the MNIST, Fashion MNIST and CelebA data sets show that, compared with baseline models, MetaCGAN performs the best.

II. RELATED WORK

Meta-learning is also called "Learning to learn", and generally deals with the few-shot learning problem or zero-shot learning problem by learning prior knowledge to guide the performance on the new classes. Because it learns transferable prior knowledge between the base classes and the new classes, it can be well adapted to the few data in the new classes. Existing mete-learning algorithms can be roughly divided

* Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

into three categories: that sharing a distance metric between the base and new classes [3] [4] [5] [6], that sharing the initialization of network parameters [7] [8] [9] [10] [11] [12], and that sharing the optimization algorithm [13] [14] [15].

Generative Adversarial Network (GAN) consists of a generator and a discriminator, and is originally proposed by Goodfellow et al. [16]. The generator hopes that the distribution of the generated samples is closer to the distribution of real samples, so that the generator can generate more realistic samples to fool the discriminator, whereas the discriminator strives to distinguish real images and generated images. The ideal state is that the generator and the discriminator achieve Nash equilibrium, namely, the generated samples by the generator are so realistic that the discriminator can judge with 50% correctness. GAN can be applied in many fields, such as image generation [17] [18] [19], image transformation [20] [21] [22], scene synthesis [23] [24], face synthesis [25], text to image generation [26], style transferable [27], image super-resolution [28], image domain transformation [29], and image inpainting [30] [31].

In the literature, some pieces of work have successfully combined meta-learning with GAN. Specifically, most of the existing approaches combining meta-learning with GAN are from the perspective of few-shot learning. In general, applying both the GAN and meta-learning modules improves the performance of few-shot learning. Specifically, MetaGAN combines two meta-learning methods with the discriminator in GAN [32]. However, the quality of the generated images by MetaGAN is not high. Wang et al. combine a meta-learner with a hallucinator, which produces additional training examples to train the generator in GAN, leading to a 6 point boost in classification accuracy on the challenging ImageNet low-shot classification benchmark [33]. Similar to the idea of MetaGAN, it only considers whether GAN is beneficial for few-shot classification but does not consider the authenticity of the generated samples. In order to improve the quality of the generated samples, a GAN meta-trained with Reptile, is proposed to generate novel samples with few training data [34]. Though the quality of the generated samples are better than that generated by the above two methods, the generated samples are noisy and still in low quality.

Unlike existing methods, we add a MetaNet module to the generator in GAN, and apply meta-learning to learn prior experience from the base classes and transfer knowledge to the new classes, which is beneficial to the quality of the generated images and increases the generation ability of the generator. Our proposed method MetaCGAN not only performs well with respect to the quality of the generated samples, but also encourages the generated samples to be diverse.

III. METHODOLOGY

In order to improve the quality and diversity of the generated samples under the condition that the training data are scarce, we build our conditional generative adversarial networks (C-GAN) with a MetaNet module and called it MetaCGAN. We employ the MetaNet module to generate weights that are needed to generate images. With the help of the MetaNet module, MetaCGAN can produce high quality and diversity samples of the new classes.

A. The CGAN framework

We first introduce the CGAN framework on which our model is built. Unlike standard GAN, CGAN adds supplementary information y that may be image or label in both of its generator and discriminator. We regard y as additional input layer in both the generator and discriminator. Same as standard GAN, CGAN achieves Nash equilibrium by maximizing discriminator loss and minimizing generator loss. The objective function of the CGAN framework is:

$$\begin{aligned}\min_G \max_D V(D,G) = E_{x \backsim p_{data}(x)}[logD(x|y)]+ \\ E_{z \backsim p_z(z)}[log(1-(D(G(z|y)))]\end{aligned} \tag{1}$$

In Equation (1), G represents the generator, D represents the discriminator, $p_{data}(x)$ stands for the probability of the real samples, and $p_z(z)$ is the probability of noise.

B. The MetaNet module

Our MetaNet module comprises of three parts, two encoders (a task encoder and an image encoder) and a parameter generator.

1) Encoders: **Task encoder.** The task encoder is fed with information output from the third deconvolutional layer of the generator. Through the task encoder, we can obtain a series of means and variances that are respectively denoted as μ_t and σ_t in a batch.

Image encoder. In order to improve the performance of the MetaNet module on generating the deconvolutional weights, we feed real images into the image encoder. With the information from real images, the generator can generate high quality images. Through the image encoder, we can obtain a series of means and variances that are respectively denoted as μ_i and σ_i in a batch.

Then, we calculate the means of μ_t and μ_i, σ_t and σ_i, respectively denoted as μ_j and σ_j, which contains sufficient statistics for the distribution of the weights.

2) parameter generator: We apply the parameter generator to learn knowledge from the base classes and provide weights for the fourth and fifth deconvolutional layers of the generator to produce high quality images. We use two encoders to calculate means and variances, and employ reparameterization trick that means sampling a random number ϵ from $N(0, I)$ to make $z = \mu_j^N + \sigma_j^N \times \epsilon_j^N$ rather than directly sampling a z from $N(\mu_j, \sigma_j^2)$. In the formula, N means the number of samples in a batch. For the fourth and fifth deconvolutional layers of the generator, we use a single layer perceptron to generate weights that are represented as θ_j^N.

In order to stabilize training and prevent from generating large weights, we employ weight normalization to constrain the weight scale. Specifically, we apply L2 normalization in each filter in the fourth and fifth deconvolutional layers of the generator.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

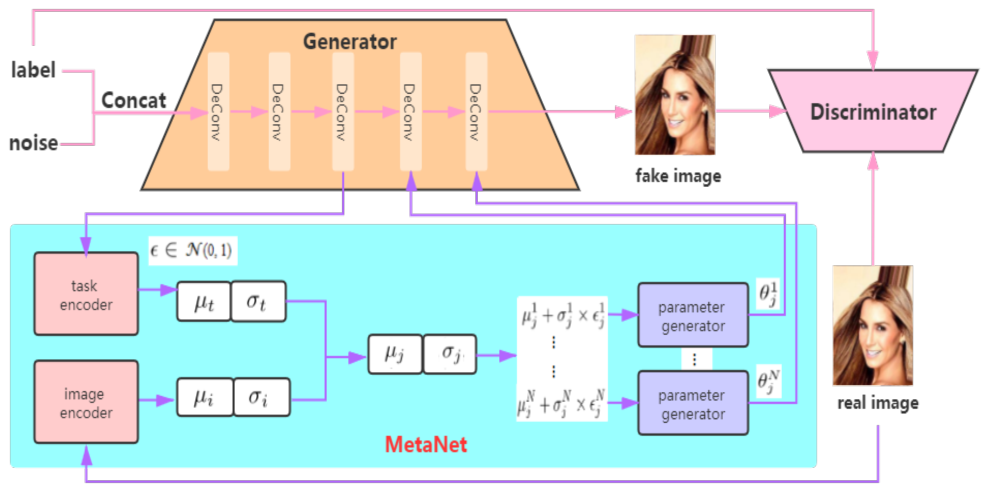


Fig. 1. The architecture of MetaCGAN. The top is the CGAN module, while the below is the MetaNet module.

C. The MetaCGAN model

To enforce the generator of CGAN generating high quality and diversity images with few data in the new classes, we use a MetaNet module to generate deconvolutional weights to improve the generation ability of the generator.

As shown in Fig. 1, our MetaCGAN model comprises of two parts, the MetaNet module and the CGAN framework. We concatenate label and noise, and feed them into the CGAN framework, of which the deconvolutional weights of the fourth and fifth layers are provided by the MetaNet module that can help the CGAN framework to produce high quality and diversity images. For example, as shown in Fig. 1, in the MetaNet module, an face image is fed into the image encoder, while information in the generator through three deconvolutional layers is fed into the task encoder, and then, we employ the reparameterization trick and parameter generator to produce the weights that are needed to generate images for the new classes. Subsequently, we calculate the losses and update the gradient of MetaCGAN in a batch manner. Finally, the CGAN framework with the generated weights can perform well on generating high quality and diversity images with few data in the new classes. Algorithm 1 shows the training scheme of MetaCGAN.

D. Normalization

In MetaCGAN, we employ the MetaNet module to generate useful generator weights that are used for helping to produce high quality samples with few data in the new classes. On the purpose of learning common prior knowledge between the base classes and the new classes, we hope similar classes can share valuable information. In order to enable the base classes and the new classes to interact with each other in a batch, we apply instance normalization (IN) technique [35]. Specifically, in the experiment, we apply IN in both the image encoder and the task encoder modules. Namely, we use IN to the training samples in a batch, and calculate mean, variance and scale

---

Algorithm 1 The training scheme of MetaCGAN.

Require: θ_G: G's parameters; θ_j: feature extractor's parameters; θ_d: discriminative layers' parameters; N: batch size; N_c: the learning steps per iteration.
Require: Adam hyperparameters: $\alpha = 0.5$, $\beta = 0.9$, $N_c = 4$.

```
Initial discriminators parameters θ_d, initial generators parameters θ_G, learning rate= 2 × 10^{-4}.
while not converged do
  for i = 1, 2, ..., epoch do
    Divide the original data set into base classes and new classes.
    if number%N_c = 0 then
      Samples (x_i, z_i, y_i) from the base classes.
    else
      Samples (x_i, z_i, y_i) from the new classes.
    end if
    Employ the task encoder to calculate μ_t and σ_t.
    Employ the image encoder to calculate μ_i and σ_i.
    μ_j, σ_j = (μ_t + μ_i)/2 + (σ_t + σ_i)/2.
    Employ reparameterization trick to obtain z_i = μ_j^i + σ_j^i × ε_j^i.
    Employ the parameter generator to obtain deconvolutional parameter θ_j^i.
    Sample and allocate weights θ_j^i for deconvolutional layer of the generator.
    L_D ← logD(x_i|y_i) + log(1 − D((G(z_i|y_i))))
    L_G ← logD(G(x_i))
    θ_d, θ_j = RMSProp(L_D, θ_d, θ_j)
    θ_G = Adam(L_D, θ_G, α, β)
  end for
  Compute batch losses and update weights.
end while
```

parameters that are shared between the base classes and the new classes.

IV. Experiment

In this section, we report the performance of the proposed MetaCGAN method on the MNIST, Fashion MNIST and CelebA data sets. In the following, we mainly evaluate the effect of the MetaNet module in MetaCGAN, and compare MetaCGAN with previous GANs, on the three data sets.

A. Baseline models

We adopted CGAN, DCGAN and WGAN as our baseline models. When we compared MetaCGAN with CGAN, DCGAN and WGAN, we respectively applied a small number of data in the new classes and a large amount of data in the new classes to train CGAN models. However, DCGAN and WGAN were trained only using a large amount of data in the new classes, and MetaCGAN was only trained on few data in the new classes.

DCGAN increases the generator ability of generating samples by introducing convolutional networks into the generator, which makes the generated samples to be of high quality.

WGAN uses Wassertein distance to measure the distance between the generated data distribution and the real data distribution. In addition, WGAN applies the truncation trick to scale the discriminator's parameters, which to some extent solves the unstable training issue of GAN.

B. Data sets

MNIST. We regarded the numbers $0, 1, 2, 3, 4$ as the base classes that totally included 28038 images and regarded the numbers $5, 6, 7, 8, 9$ as the new classes that each class only included 10 images during training using MetaCGAN. In the test phase, we did not input real images to MetaCGAN when the generator produced fake images, and only fed real images into the discriminator.

Fashion MNIST. We regarded t-shirt class, trouser class, pullover class, dress class and coat class as the base classes that totally included 27494 images and regarded sandal class, shirt class, sneaker class, bag class and ankle boot class as the new classes in which each class only included 10 images during training using MetaCGAN. The test phase was as the same as MNIST.

CelebA. We regarded woman as the base class and used all women images during training using MetaCGAN. In addition, we regarded man as the new class and used 50 men images during training using MetaCGAN. The test phase was as the same as MNIST.

C. Implementation details

We conducted experiment on the NVIDIA GeForce GTX 1080Ti GPU with Python 3.6.9 and Tensorflow 1.12.2. The discriminator with five convolutional layers was trained using RMSProp with learning rate 2×10^{-4}. Generator with five deconvolutional layers was trained using Adam with $\alpha = 0.5$, $\beta = 0.9$ in our MetaCGAN. We set batch size to 16. We sampled three times from the base classes and one time from the new classes, four images each time. We trained the generator with a learning rate 2×10^{-4} for the first 500 epochs and exponentially decayed the learning rate to 0. We fed noise with 100 dimensions. For the baseline models, we downloaded their publicly available codes from GitHub and set parameters according to which reported in the original papers.

D. Experimental results

To demonstrate the advantage of the MetaNet module, we respectively conducted experiments using CGAN and MetaCGAN on the three data sets. In order to demonstrate MetaCGAN with prior experience is effective, we also compared the performance of MataCGAN on a small number of data with that of the baseline models on a large amount of data in the new classes. We did not only qualitatively evaluate the experimental results, but also quantitatively compared the methods.

Qualitative evaluation. Fig. 2 shows the generated images by applying our method MetaCGAN and the baseline models on the MNIST digits. Comparing (a) with (e), we can see that the generated images by MetaCGAN perform better than that by using CGAN, though we used fewer data in the new classes than CGAN, which demonstrates that the key of generating high quality images greatly depends on the MetaNet module. Comparing (b) (c) (d) (e), it is easy to observe that the generated images provide the highest visual quality and diversity using our MetaCGAN, not only on the numbers $0, 1, 2, 3, 4$ but also on the numbers $5, 6, 7, 8, 9$, among the baseline models, even under the condition that the baseline models applied all data whereas our MeteCGAN only used 10 images in the new classes. The reason for this is that our MetaCGAN can learn prior experience on the base classes and transfer knowledge to generate generator's weights, helping for generating images.

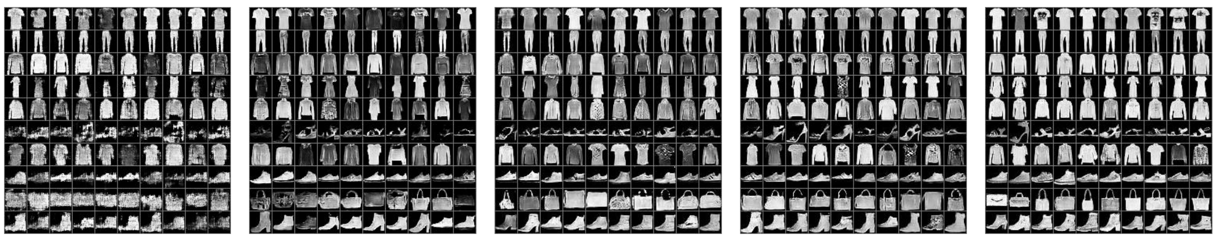
The generated images applied our method MetaCGAN and the baseline models on the Fashion MNIST data set are shown in Fig. 3. As shown in Fig. 3 (a) and (e), it is easy to see that the generated images on the new classes by MetaCGAN are of higher quality and diversity when we applied 10 images in each new class than that by CGAN when we applied 20 images in new class, which demonstrates the MetaNet module is effective for learning the prior experience of the base classes and transferring it to generate the new class images. Fig. 3 (b) (c) (d) are the generated images using all images by CGAN, DCGAN and WGAN. It is easy to observe that MetaCGAN performs the best on both the base classes and the new classes than the baseline models. It demonstrates that the MetaNet module in MetaCGAN is more capable of generating higher quality and diversity images than using a large amount of data.

We also applied our method MetaCGAN to more challenging data set with colorful and high resolution images, the CelebA data set. The generated images applied MetaCGAN and the baseline models are shown in Fig. 4. Comparing (b) (c) (d) with (e), the experimental results show that MetaCGAN is superior to the baseline models when the baseline models applied all data whereas MetaCGAN only used 50 images in

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

(a) The generated images by CGAN using 20 images in each new class. (b) The generated images by CGAN using all images in each new class. (c) The generated images by DCGAN using all images in each new class. (d) The generated images by WGAN using all images in each new class. (e) The generated images by MetaCGAN using 10 images in each new class.

Fig. 2. The generated images using the baseline models and MetaCGAN on the MNIST data set.



(a) The generated images by CGAN using 20 images in each new class. (b) The generated images by CGAN using all images in each new class. (c) The generated images by DCGAN using all images in each new class. (d) The generated images by WGAN using all images in each new class. (e) The generated images by MetaCGAN using 10 images in each new class.

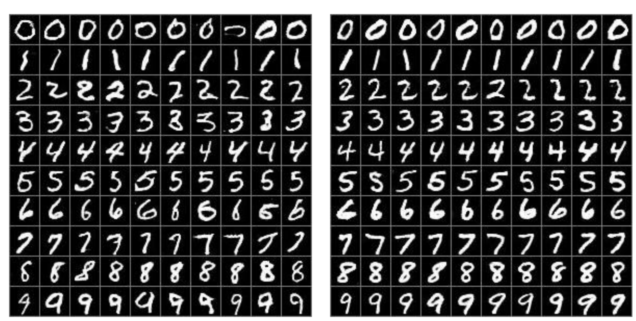
Fig. 3. The generated images using the baseline models and MetaCGAN on the Fashion MNIST data set.

(a) The generated images by CGAN using 1000 images in each new class. (b) The generated images by CGAN using all images in each new class. (c) The generated images by DCGAN using all images in each new class. (d) The generated images by WGAN using all images in each new class. (e) The generated images by MetaCGAN using 50 images in each new class.

Fig. 4. The generated images using the baseline models and MetaCGAN on the CelebA data set.

the new class, not to mention experimental results of CGAN using 1000 images in the new class as shown in Fig. 4 (a). It demonstrates that the generated images by MetaCGAN with prior experience are of higher quality and diversity than the baseline models without prior knowledge.

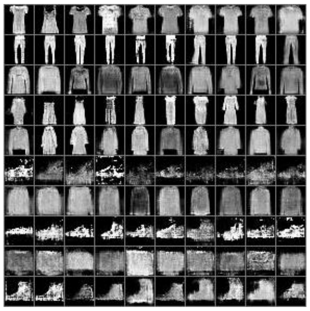
To demonstrate that our MetaNet module can be applied to any GAN and evaluate the effect of the MetaNet module, we also did experiment on the MNIST data set using DCGAN combined with our MetaNet module. The generated images employing DCGAN with all training data are as shown in Fig. 5 (a), while that of applying DCGAN combined with the MetaNet module with 20 images in each new class are as shown in Fig. 5 (b). Comparing their performances, we can see that DCGAN combined with the MetaNet module performs better than DCGAN, on not only the quality of the generated images but also the diversity of the generated images, which demonstrates that the MetaNet module is effective and helpful for the generator in GAN models.



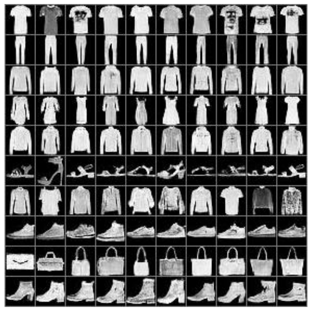
(a) DCGAN (b) DCGAN combined with the MetaNet module.

Fig. 5. The generated images on the Fashion MNIST data set.

In order to evaluate the influence of the IN strategy for the generated images, we conducted a comparison experiment on whether or not employed the IN strategy. The generated images are shown in Fig. 6, Fig. 6 (a) applied batch normalization to replace the IN strategy. We can see that the quality of the generated images by our MetaCGAN with the IN strategy are better than that without the IN strategy. Furthermore, experimental results demonstrate that the IN strategy can help similar classes to share valuable information, which is beneficial to learn common prior knowledge between the base classes and the new classes, generating better deconvolutional weights to improve the ability of the generator for image generation.



(a) Our method MetaCGAN applying batch normalization rather than IN.



(b) Our method MetaCGAN.

Fig. 6. The generated images on the Fashion MNIST data set.

Quantitative evaluation. In order to evaluate the performance of MetaCGAN, inception score (IS) was applied to measure the quality and diversity of the generated images. IS is defined as $IS(G) = \exp(E_x[D_{KL}(p(y|x)||p(y))])$, where x means the generated image, y means a vector obtained by using a well-trained classifier to predict the probability of x belonging to each class, $p(y|x)$ means the conditional class distribution, $p(y)$ means the marginal distribution, and D_{KL} means the KL-divergence. Based on the definition of IS, we need to ensure not only low entropy of $p(y|x)$ but also high entropy of $p(y)$, making IS be large. Concretely, low entropy of $p(y|x)$ represents that the decision boundary of the classes is compact whereas high entropy of $p(y)$ represents high diversity of images. Hence, IS combines the quality and diversity measurements into one criterion.

Generally, the diversity of the generated images can be represented by the discriminability of the decision boundary between classes, namely, if two images are dissimilar with high probability, they should be classified into different classes. In addition, classifier can classify accurately when the quality of the generated images is high. Therefore, we apply IS to measure the quality and diversity of the generated images. Nevertheless, as IS is based on ImageNet and applies inception net-v3, we respectively trained the classifier when we employed IS to evaluate the generated images by MetaCGAN and other baseline models on the MNIST and Fashion MNIST data sets.

Table I shows IS obtained by MetaCGAN and the baseline models on the MNIST and Fashion MNIST data sets. Compared with the baseline models that used all data whereas MetaCGAN only applied few images in the new classes. In this circumstance, IS obtained by our MetaCGAN is the largest,

TABLE I
COMPARISON OF OUR METACGAN WITH CGAN, DCGAN AND WGAN ON THE MNIST AND FASHION MNIST DATA SETS IN TERMS OF IS.

Models	MNIST	Fashion MNIST
CGAN	1.68	2.68
DCGAN	1.88	3.79
WGAN	1.90	3.80
MetaCGAN	**2.48**	**5.46**

which demonstrates that MetaCGAN generates the highest quality and diversity images on the two data sets. The experimental results demonstrate that adding the MetaNet module on CGAN is beneficial for tackling the image generation problem with few data. From the perspective of the generated images, it is easy to see that our MetaCGAN is superior to the baseline models, though the baseline models use a large amount of data. What's more, it also demonstrates that MetaCGAN with prior experience is effective. There is no doubt that the success of our MetaCGAN depends on the MetaNet module that learns prior experience on the base classes and transfers knowledge to generate the new classes.

Except for IS, we also applied Amazon Mechanical Turk (AMT) to evaluate the performance of the generated images on the CelebA data set. Given the generated images by our method and the baseline models, we required the Turkers to choose the best generated image based on perceptual realism and diversity. We provided four randomly generated images by four kinds of methods. In addition, we asked each Turker 5 to 10 simple yet logical questions for validating whether or not she/he took it seriously. As a result of which, we found that 50 Turkers were positive.

TABLE II
THE AMT PERCEPTUAL EVALUATION USING THE BASELINE METHODS AND OUR METACGAN ON THE CELEBA DATA SET.

Model	CGAN	DCGAN	WGAN	MetaCGAN
Score	7.4%	21.7%	23.5%	**47.4%**

Table II shows the score of the AMT perceptual evaluation to our MetaCGAN and the baseline models on the CelebA data set. It is easy to see that the score of using MetaCGAN is highest, which demonstrates that our MetaCGAN is superior to the baseline models and the quality of the generated image is the most authenticity.

V. CONCLUSION

In this work, we propose a novel generative adversarial network called MetaCGAN for generating high quality and diversity images with few data in the new classes. Our training data is based on a large amount of data in the base classes and a small number of data in the new classes. Under this circumstance, inspired by meta-learning that can learn prior experience from the base classes and transfer knowledge to the new classes to generate network parameters for generating images, we build our MetaCGAN that is based on the CGAN framework combined with the MetaNet module. The MetaNet module is applied to provide deconvolutional weights of the generator in CGAN. Furthermore, in order to share valuable information between similar classes, we employ the instance normalization (IN) strategy. The experimental results on the MNIST, Fashion MNIST and CelebA data sets show the superiority of MetaCGAN over the baseline models. Not only qualitative results but also quantitative results demonstrate that our MetaCGAN model can generate the high quality and diversity images on the three data sets. In addition, experimental results demonstrate that the MetaNet module is effective and can transfer prior knowledge from the base classes to the new classes.

Acknowledgments

This work was supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, the National Key R&D Program of China under Grant 2016YFC1401004, the National Natural Science Foundation of China (NSFC) under Grant No. 41706010, the Joint Fund of the Equipments Pre-Research and Ministry of Education of China under Grand No. 6141A020337, and the Fundamental Research Funds for the Central Universities of China.

References

[1] S. Thrun and L. Pratt, Learning to Learn: Introduction and Overview. Learning to Learn 1998: 3-17.
[2] T. Munkhdalai, H. Yu, Meta Networks. ICML 2017: 2554-2563.
[3] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. Torr, and T. Hospedales, Learning to Compare: Relation Network for Few-Shot Learning. CVPR 2018: 1199-1208.
[4] J. Snell, K. Swersky, and R. Zemel, Prototypical Networks for Few-Shot Learning. NIPS 2017: 4077-4087.
[5] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, Matching Networks for One Shot Learning. NIPS 2016: 3630-3638.
[6] G. Koch, R. Zemel, and R. Salakhutdinov, Siamese Neural Networks for One-Shot Image Recognition. ICML 2015.
[7] S. Gidaris and N. Komodakis, Dynamic Few-Shot Visual Learning without Forgetting. CVPR 2018: 4367-4375.
[8] C. Finn, P. Abbeel, and S. Levine, Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML 2017: 1126-1135.
[9] H. Qi, M. Brown, and D. Lowe, Low-Shot Learning with Imprinted Weights. CVPR 2018: 5822-5830.
[10] F. Gomez and J. Schmidhuber, Evolving Modular Fast-Weight Networks for Control. ICANN (2) 2005: 383-389
[11] S. Qiao, C. Liu, W. Shen, and A. Yuille, Few-shot Image Recognition by Predicting Parameters from Activations. CoRR abs/1706.03466 (2017).
[12] D. Ha, A. Dai, and Q.Le, Hypernetworks, CoRR abs/1609.09106 (2016).
[13] M. Andrychowicz, M. Denil, S. Gomez, M. Hoffman, D. Pfau, T. Schaul, and N. Freitas, Learning to Learn by Gradient Descent by Gradient Descent. NIPS 2016: 3981-3989.
[14] S. Ravi and H. Larochelle, Optimization as a Model for Few-Shot Learning. ICLR 2017.
[15] T. Munkhdalai, X. Yuan, S. Mehri, T. Wang, and A. Trischler, Learning Rapid-Temporal Adaptations. CoRR abs/1712.09926 (2017).
[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Farley, S. Ozair, A. Courville, and Y. Bengio, Generative Adversarial Networks. CoRR abs/1406.2661 (2014).
[17] M. Mirza and S. Osindero, Conditional Generative Adversarial Nets. CoRR abs/1411.1784 (2014).
[18] A. Radford, L. Metz and S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ICLR (Poster) 2016.
[19] M. Arjovsky, S. Chintala, and L. Bottou, Wasserstein Generative Adversarial Networks. ICML 2017: 214-223.
[20] P. Isola, J. Zhu, T. Zhou, and A. Efros, Image-to-Image Translation with Conditional Adversarial Networks. CVPR 2017: 5967-5976.
[21] J. Zhu, T. Park, P. Isola, and A. Efros, Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. ICCV 2017: 2242-2251.
[22] T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim, Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. ICML 2017: 1857-1865.
[23] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. CVPR 2018: 8798-8807.
[24] R. Huang, S. Zhang, T. Li, and R. He, Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. ICCV 2017: 2458-2467.
[25] H. Zhang, T. Xu, and H. Li, StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. ICCV 2017: 5908-5916.
[26] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, X. He, AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. CVPR 2018: 1316-1324.
[27] C. Li and M. Wand, Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. ECCV (3) 2016: 702-716.
[28] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. CVPR 2017: 105-114.
[29] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. CVPR 2018: 8789-8797.
[30] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, Generative Image Inpainting with Contextual Attention. CVPR 2018: 5505-5514.
[31] G. Liu, F. Reda, K. Shih, T. Wang, A. Tao, and B. Catanzaro, Image Inpainting for Irregular Holes Using Partial Convolutions. ECCV (11) 2018: 89-105.
[32] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, MetaGAN: An adversarial Approach to Few-Shot Learning. NIPS 2018: 2371-2380.
[33] Y. Wang, R. Girshick, M. Hebert, and B. Hariharan, Low-shot Learning from Imaginary Data. CVPR 2018: 7278-7286.
[34] L. Clouatre and M. Demers, FIGR: Few-Shot Image Generation with Reptile. CoRR abs/1901.02199 (2019).
[35] D. Ulyanov, A. Vedaldi, and V. Lempitsky, Instance Normalization: The Missing Ingredient for Fast Stylization. CoRR abs/1607.08022 (2016)