

## LETTER

# Iterative Adversarial Inference with Re-Inference Chain for Deep Graphical Models

Zhihao LIU<sup>†</sup>, Nonmember, Hui YIN<sup>†a)</sup>, Member, and Hua HUANG<sup>†</sup>, Nonmember

**SUMMARY** Deep Graphical Model (DGM) based on Generative Adversarial Nets (GANs) has shown promise in image generation and latent variable inference. One of the typical models is the Iterative Adversarial Inference model (GibbsNet), which learns the joint distribution between the data and its latent variable. We present RGNet (Re-inference GibbsNet) which introduces a re-inference chain in GibbsNet to improve the quality of generated samples and inferred latent variables. RGNet consists of the generative, inference, and discriminative networks. An adversarial game is cast between the generative and inference networks and the discriminative network. The discriminative network is trained to distinguish between (i) the joint inference-latent/data-space pairs and re-inference-latent/data-space pairs and (ii) the joint sampled-latent/generated-data-space pairs. We show empirically that RGNet surpasses GibbsNet in the quality of inferred latent variables and achieves comparable performance on image generation and inpainting tasks.

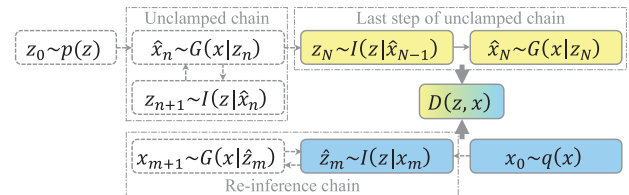
**key words:** deep graphical model, generative adversarial nets, latent variable, inference, generation

## 1. Introduction

Generative adversarial networks (GANs) [1] has been a powerful framework of the deep directed generative model for learning an underlying representation of high-dimensional data. Using the flexibility and adversarial mechanism of GANs, the deep generative model can generate high-quality samples [2] and representative hidden variables [3].

Existing DGMs which generate samples and infer latent variables usually rely on Variational Autoencoder (VAE, [4]) and GANs. VAE is based on Maximum Likelihood Estimate (MLE) and can learn an approximate inference mechanism that expands the scope of application of the model. However, the VAE-based model tends to distribute probability mass diffusely over the data space [5]. Adversarially Learned Inference (ALI, [6]) and GibbsNet [7] cast the learning of both an inference network and a generative network in a GAN-like adversarial framework. Both models are trained to match the joint sampled-latent/generated-data-space distribution  $\pi_G$  and the joint inferred-latent/data-space distribution  $\pi_I$ , but it would be more reasonable if more distributions with data information are used to train the model.

Suppose we have the optimal GibbsNet and a sample



**Fig. 1** Diagram illustrating the architecture of RGNet. The re-inference chain introduced in GibbsNet is in the right. It starts with  $x_0$  from the data distribution and runs for  $M$  steps. Each step of the re-inference chain are then compared with the last step of the unclamped chain using the discriminative network  $D$ .

$x_0$  from the data distribution  $q(x)$ , we can use the inference network  $I$  to get its latent variable  $\hat{z}_0 \sim I(z|x_0)$ , and then use the generative network  $G$  to get the generated sample  $x_1 \sim G(x|\hat{z}_1)$ , and use  $x_1$  to get the re-inference latent variable  $\hat{z}_1 \sim I(z|x_1)$ . Ideally,  $\hat{z}_0$  should be the same as  $\hat{z}_1$ , and the re-inference pair  $(x_0, \hat{z}_1)$  from the re-inference distribution  $\pi_{RI}$  should converge to  $\pi_I$  and  $\pi_G$ . By making generation and inference iteratively for  $M$  steps, we can get  $M$  re-inference pairs  $(x, \hat{z}_m), m \in 1 \dots M$  which are very useful for unsupervised learning.

Therefore, a novel approach named RGNet (Re-inference GibbsNet) is proposed. RGNet follows GibbsNet and introduces a re-inference chain into the training procedure, as shown in Fig. 1. RGNet is trained to match  $\pi_G, \pi_I$ , and the re-inference distribution  $\pi_{RI}$ . We prove theoretically that RGNet has the same optimal discriminator as GibbsNet but has more convergence conditions. In our experiments, RGNet surpasses GibbsNet in the quality of inferred latent variables and achieves comparable performance on image generation and inpainting tasks.

## 2. Re-Inference GibbsNet

The re-inference chain is introduced in RGNet to iteratively make generation and inference to get the re-inference pairs in each iteration of the training process. And RGNet is trained to match  $\pi_G, \pi_I$  and the re-inference distribution  $\pi_{RI}$ .

### 2.1 Value Function

Same as GibbsNet, we use adversarial game to match the three kind of distributions. In each iteration, the generative pair is drawn from the last step of the unclamped chain: start the chain with a latent variable  $z_0$  from a normal distribution  $N(0, I)$  and follow this by  $N$  steps of alternating between

Manuscript received December 13, 2018.

Manuscript revised March 29, 2019.

Manuscript publicized May 7, 2019.

<sup>†</sup>The authors are with School of Computer and Information Technology and Beijing key lab of traffic data analysis and mining, Beijing Jiaotong University, Beijing, 100044, China.

a) E-mail: hyin@bjtu.edu.cn

DOI: 10.1587/transinf.2018EDL8256

sampling from  $G(x|z)$  and  $I(z|x)$ . The inference pair  $(x_0, \hat{z}_0)$  is drawn from  $I(z|x)$ , given  $x_0$  is drawn from the data distribution  $q(x)$ . The re-inference pair  $(x_0, \hat{z}_m), m \in 1 \dots M$  is drawn from the re-inference chain: start the re-inference chain and runs for  $m$  steps of alternating between sampling from  $I(z|x)$  and  $G(x|z)$ . The discriminative network  $D$  learns to discriminate  $G(\hat{x}_N, z_N), I(x_0, \hat{z}_m), m \in 0 \dots M$ , while  $G$  and  $I$  are trained to fool  $D$ . The value function describing the game is given by:

$$\min_{G,I} \max_D V(D, G, I) = \sum_{m=0}^M (E_{q(x)}[\log(D(x_0, I(x_m)))] + E_{p(z)}[\log(1 - D(G(z_N), z_N))]) \quad (1)$$

where  $x_m \sim G(x|\hat{z}_{m-1})$ ,  $z_n \sim I(z|\hat{x}_{n-1})$ ,  $\hat{x}_0 \sim G(x|z_0)$ ,  $m \in 1 \dots M$ , and  $n \in 1 \dots N$ .

## 2.2 Training Procedure of RGNet

Each iteration of the training procedure consists of two phases: the first one is training the model follows the way of GibbsNet (lines 3-11); the second one is starting the re-inference chain to get the re-inference pairs and train the model  $M$  times (line 12-24). The description of the training procedure is shown in Algorithm 1.

---

### Algorithm 1 Training Procedure of RGNet

---

**Input:** sampling step  $N$ , re-inference step  $M$ , iteration number  $Iter$ .  
**Output:** inference network  $I$ , generative network  $G$ , discriminative network  $D$ .

```

1: initialize  $D, G$ , and  $I$ ;
2: for  $i \in 1 \dots Iter$  do
3:   Sample a batch of data  $x_0$  from the data distribution  $q(x)$ ;
4:   Sample a batch of  $z_0$  from a normal distribution  $\mathcal{N}(0, I)$ ;
5:   for  $n \in 1 \dots N$  do
6:      $\hat{x}_{n-1} \leftarrow G(x|z_{n-1})$ ;
7:      $z_n \leftarrow I(z|\hat{x}_{n-1})$ ;
8:   end for
9:    $\hat{x}_N \leftarrow G(x|z_N)$ ;
10:   $\hat{z}_0 \leftarrow I(z|x_0)$ ;
11:  Train  $D, G, I$  using  $(\hat{x}_N, z_N)$  and  $(x_0, \hat{z}_0)$ ;
12:  for  $m \in 1 \dots M$  do
13:    for  $n \in 1 \dots N$  do
14:       $\hat{x}_{n-1} \leftarrow G(x|z_{n-1})$ ;
15:       $z_n \leftarrow I(z|\hat{x}_{n-1})$ ;
16:    end for
17:     $\hat{x}_N \leftarrow G(x|z_N)$ ;
18:     $\hat{z}_0 \leftarrow I(z|x_0)$ ;
19:    for  $j \in 1 \dots m$  do
20:       $x_j \leftarrow G(x|\hat{z}_{j-1})$ ;
21:       $\hat{z}_j \leftarrow I(z|x_j)$ ;
22:    end for
23:    Train  $D, G, I$  using  $(\hat{x}_N, z_N)$  and  $(x_0, \hat{z}_m)$ ;
24:  end for
25: end for

```

---

## 3. Theoretical Analysis

Although the re-inference chain is introduced in RGNet,

the optimal discriminator is a straight forward extension of GANs and has the similar relationship with the Jensen-Shannon divergence. And the introduction of re-inference pairs successfully increases the convergence conditions which ensures better convergence in practice.

### 3.1 Discriminator Optimality

**Proposition 1.** Given a fixed generator ( $G$  and  $I$ ), the optimal discriminator ( $D$ ) is given by

$$D^*(x, z) = \sum_{m=0}^M \frac{I(x_0, \hat{z}_m)}{I(x_0, \hat{z}_m) + G(\hat{x}_N, z_N)}. \quad (2)$$

*Proof.* For a fixed generator, the complete data value function is:

$$V(D, G, I) = \sum_{m=0}^M (E_{(x_0, \hat{z}_m) \sim I(x_0, \hat{z}_m)}[\log(D(x_0, \hat{z}_m))] + E_{(\hat{x}_N, z_N) \sim G(\hat{x}_N, z_N)}[\log(1 - D(\hat{x}_N, z_N))]) \quad (3)$$

For any  $(\sum_{m=0}^M a_m, b) \in \mathbb{R}^2 \setminus (0, 0)$ , the function  $y \rightarrow \sum_{m=0}^M (a_m \log(y) + b \log(1 - y))$  achieves its maximum in  $[0, 1]$  at  $\frac{\sum_{m=0}^M a_m}{\sum_{m=0}^M (a_m + b)}$ . The discriminator does not need to be defined outside of  $Supp(G(\hat{x}_N, z_N)) \cup Supp(I(x_0, \hat{z}_m)), m \in 1 \dots M$ , concluding the proof.  $\square$

### 3.2 Relationship with the Jensen-Shannon Divergence

**Proposition 2.** Under an optimal discriminator  $D^*$ , the generator minimizes the JS divergence which attains its minimum if and only if  $G(\hat{x}_N, z_N) = I(x_0, \hat{z}_m), m \in 0 \dots M$ .

*Proof.* Let  $C(G, I) = \max_D V(D, G, I)$ , then

$$C(G, I) = \sum_{m=0}^M (E_{(x_0, \hat{z}_m) \sim I(x_0, \hat{z}_m)}[\log(\frac{a_m}{a_m + b})] + E_{(\hat{x}_N, z_N) \sim G(\hat{x}_N, z_N)}[\log(\frac{b}{a_m + b})]) \quad (4)$$

For  $G(\hat{x}_N, z_N) = I(x_0, \hat{z}_m), m \in 0 \dots M$ ,  $D^*(x, z) = \frac{1}{2}$  (consider Eq. (2)). Hence, by inspecting Eq. (4) at  $D^*(x, z) = \frac{1}{2}$ , we find  $C(G, I) = -(M + 1) \log(4)$ , and we obtain:

$$C(G, I) = -(M + 1) \log(4) + \sum_{m=0}^M (KL(I(x_0, \hat{z}_m) || \frac{I(x_0, \hat{z}_m) + G(\hat{x}_N, z_N)}{2}) + KL(G(\hat{x}_N, z_N) || \frac{I(x_0, \hat{z}_m) + G(\hat{x}_N, z_N)}{2})) \quad (5)$$

where KL is the Kullback-Leibler divergence. This equation can be transformed to the Jensen-Shannon divergence between the inference/re-inference distributions and the generative distribution:  $C(G, I) = -(M + 1) \log(4) +$

$2 \sum_{m=0}^M JS D(I(x_0, \hat{z}_m) \| G(\hat{x}_N, z_N))$ . And the global minimum of  $C(G, I)$  is  $C^* = -(M+1) \log(4)$ . The only solution is  $G(\hat{x}_N, z_N) = I(x_0, \hat{z}_m), m \in 0 \dots M$ .  $\square$

### 3.3 Convergence

**Proposition 3.** Assuming optimal discriminator and Generator. Then the pairs  $(\hat{x}_n, z_n), (x_n, \hat{z}_{n+m}), n \in 1 \dots N, m \in 1 \dots M$  have the same joint distribution.

*Proof.* In the case of the optimal discriminator and generator, since  $\hat{z} \sim I(z|x)$ , when  $x_n$  is applied to  $I$ , we get  $\hat{z}_{n+1}$  which must form a joint  $(x_n, \hat{z}_{n+1})$  which has the same joint distribution as  $(x_0, \hat{z}_0)$ . Since  $\hat{z}_m \sim I(z|G(x|I(z|x_m)))$  when  $x_n$  is applied to  $I$  and  $G$  by  $m$  steps, we get  $\hat{z}_{n+m}$  which must form a joint  $(x_n, \hat{z}_{n+m})$  which has the same joint distribution as  $(x_0, \hat{z}_0)$ . Since  $(\hat{x}_n, z_n), n \in 1 \dots N$  have the same joint distribution as  $(x_0, \hat{z}_0)$  (see details in [7]),  $(\hat{x}_n, z_n)$  and  $(x_n, \hat{z}_{n+m})$  have the same joint distribution.  $\square$

## 4. Experiments

Since GibbsNet is identical to RGNet when the number of re-inference steps is  $M = 0$  and ALI is identical to GibbsNet when the number of iterative inference steps is  $N = 0$ , we use the same architecture to train these models following the design of ALI to make it fair<sup>†</sup>. Although some numerical results are different from those reported by the authors, our results are meaningful and convincing for comparative evaluation. For good measure, we compare the generation performance of RGNet with other unsupervised generative models with similar architectures.

RGNet is evaluated on three datasets: SVHN [8], CIFAR10 [9], and MNIST [10]. The number of training epochs on SVHN, CIFAR10, and MNIST are set to 100, 100, and 200 respectively. Owing to the instability of GANs training [11], it is difficult for RGNet to match more distributions. Hence, we set  $M \leq 2$  in the following experiments.

We first show the expressiveness of RGNet's learned latent variables by means of semi-supervised classification experiments. And then, to evaluate the generation performance of RGNet's generative network, the Inception Scores [12] and Fréchet Inception Distance [13] based on the pre-trained Inception model [14] are used. In the end, the image generation and inpainting results of RGNet are shown and discussed.

### 4.1 Expressiveness of RGNet's Learned Latent Variables

In this experiment, a 2-layer MLP is constructed on top of the latent variables which are inferred directly from the trained model without passing gradient from the classifier through to  $I$ . Then training MLP on the latent variables

<sup>†</sup>The code can be found at <https://github.com/hhqwesd/RGNet>.

**Table 1** Classification accuracy from different models.

	MNIST	SVHN	CIFAR10
ALI [6]	87.2%	76.2%	35.5%
GibbsNet [7]	80.2%	75.7%	32.8%
RGNet-1	90.0%	81.0%	39.9%
RGNet-2	91.3%	81.2%	43.1%

**Table 2** Inception score and Fréchet inception distance on CIFAR10.

Model	IS	FID
MIX+WGAN [15]	4.04	-
Improved-GAN [12]	4.36	-
AGE [16]	4.96	64.61
PixelIQN [17]	5.29	49.46
Dist-GAN [18]	-	45.60
ALI [6]	4.56	70.58
GibbsNet [7]	4.63	73.42
RGNet-1	5.21	61.88
RGNet-2	5.51	56.85

**Table 3** Inception score and Fréchet inception distance on SVHN.

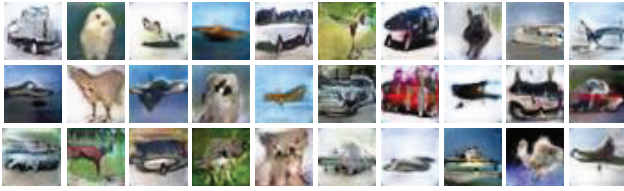
Model	IS	FID
ALI [6]	2.71	134.17
GibbsNet [7]	2.86	105.52
RGNet-1	2.70	109.66
RGNet-2	2.82	96.04

with the same number of epochs. We denote RGNet- $j$  as RGNet trained with  $M = j$ , the same below. The results are shown in Table 1. Under the same configuration on semi-supervised learning task in above datasets, GibbsNet performs worse than ALI, but RGNet still achieves the best results, which demonstrates that the latent variables inferred by RGNet have stronger and more robust expressiveness. What's more, RGNet-2 achieves the best result which further proves the effectiveness of the re-inference chain.

### 4.2 Inception Score and Fréchet Inception Distance

We use the generative network of RGNet to generate 50,000 samples for calculating the Inception Scores (IS) and Fréchet Inception Distance (FID). The IS feeds the inception model with the generated samples and measures the Kullback-Leibler divergence between the predicted conditional label distribution and the actual class probability distribution, which is correlated with human's judgement [12]. The FID uses the inception model to embed the generated samples into a special feature space. And then the Fréchet distance is used to evaluate the generated samples and data samples in that space. The FID can capture the similarity between generated samples and real ones. We report both the IS and FID results on CIFAR10 and SVHN in Table 2 and Table 3, respectively.

On CIFAR10 dataset, AGE is run with the same number of epochs as RGNet and the code is provided by the author. The other models except ALI and GibbsNet are trained with 2-3 times epochs more than RGNet. Compared to the adversarial inference models (ALI and GibbsNet), RGNet achieves the best IS and FID, which proves the advantage of the re-inference chain. Compared to



**Fig. 2** CIFAR10 generation after 20 steps (RGNet-2).



**Fig. 3** SVHN inpainting after 20 steps (RGNet-2). The left side of the image is given and the right side is in-painted.

other models, RGNet still achieves higher IS and comparable FID, which proves the robustness of RGNet. On SVHN dataset, GibbsNet achieves the best IS and RGNet achieves the best FID, which indicates that GibbsNet and RGNet are of about the same generation performance.

Notice that RGNet-2 performs better than RGNet-1 on both CIFAR10 and SVHN, which demonstrates that more re-inference pairs lead to better generation results and there is room for improvement. Taken together, the results of RGNet are competitive among state-of-the-art models and RGNet-2 surpasses RGNet-1 completely, which confirm that the re-inference pairs play an important role in the training procedure.

### 4.3 Generation and Inpainting

**Generation** Here, we show the generation result of RGNet on CIFAR10. Figure 2 shows generated samples from  $G$  of RGNet-2 with sampling steps  $N = 20$ . Samples are meaningful and can reflect the characteristics of the dataset.

**Inpainting** The inpainting is done with the transition operator used in GibbsNet. The inpainting images are the same as expected in Fig. 3. The re-inference chain introduced in RGNet will not affect the flexibility of the model.

## 5. Conclusion

In this paper, a novel approach RGNet is proposed. RGNet improves the quality of generated images and inferred latent variables by matching the generative, inference, and re-inference joint distributions, which introduces new convergence conditions and makes the model converge better. The experimental results show that RGNet achieves state-of-the-art results as a whole in the quality of inferred latent variables and comparable performance on image generation

and inpainting tasks.

## Acknowledgements

This work was supported by National Nature Science Foundation of China (61472029; 51827813; 61473031), National Key R&D Program of China (2017YFB1201104; 2016YFB1200100), and the Fundamental Research Funds for the Central Universities (2016JBZ005).

## References

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," NIPS, 2014.
- [2] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," CVPR, pp.2242–2251, 2017.
- [3] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning," CVPR, pp.8896–8905, 2018.
- [4] D.P. Kingma and M. Welling, "Auto-encoding variational Bayes," ICLR, 2014.
- [5] L. Theis, A.V.D. Oord, and M. Bethge, "A note on the evaluation of generative models," ICLR, 2016.
- [6] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially learned inference," ICLR, 2016.
- [7] A.M. Lamb, D. Hjelm, Y. Ganin, J.P. Cohen, A.C. Courville, and Y. Bengio, "Gibbsnet: Iterative adversarial inference for deep graphical models," NIPS, 2017.
- [8] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A.Y. Ng, "Reading digits in natural images with unsupervised feature learning," NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.
- [9] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," vol.1, no.4, Technical Report, University of Toronto, 2009.
- [10] Y. LeCun, "The mnist database of handwritten digits," <http://yann.lecun.com/exdb/mnist/>, 1998.
- [11] T. Chavdarova and F. Fleuret, "Sgan: An alternative training of generative adversarial networks," CVPR, pp.9407–9415, 2018.
- [12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," NIPS, 2016.
- [13] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," NIPS, 2017.
- [14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Re-thinking the inception architecture for computer vision," CVPR, pp.2818–2826, 2016.
- [15] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," ICML, 2017.
- [16] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "It takes (only) two: Adversarial generator-encoder networks," AAAI, 2018.
- [17] G. Ostrovski, W. Dabney, and R. Munos, "Autoregressive quantile networks for generative modeling," ICML, 2018.
- [18] N.-T. Tran, T.-A. Bui, and N.-M. Cheung, "Dist-gan: An improved gan using distance constraints," ECCV, vol.11218, pp.387–401, 2018.