Label noise learning with the combination to CausalNL and CGAN models

Zixing Gou^{1,*,†}**, Yifan Sun**^{2,6,†}**, Zhebin Jin**^{3,7,†}**, Hanqiu Hu**$^{4,8,\#}$**, Weiyi Xia**$^{5,9,\#}$

[1]School of Mathematics and Statistics, Shandong University, Weihai, 264209, China
[2]Department of math, University of Toronto, Toronto, M5R 0A3, Canada
[3]Sussex Institute of Artificial Intelligence, Zhejiang Gongshang University, Hongzhou, 310018, China
[4]Beijing National Day School, Beijing, 100039, China
[5]Chongqing No.1 Secondary School, Chongqing, 400123, China

*corresponding author's e-mail: kyanegou@gmail.com
[6]robert.sun@mail.utoronto.ca
[7]zj80@sussex.ac.uk
[8]hanqiuhu@outlook.com
[9]xuetuxler@gmail.com
[†]These authors are co-first authors
[#]These authors are co-second authors

Abstract. Since Deep Neural Networks easily overfit label errors, which will degenerate the performance of Deep Learning algorithms, recent research gives a lot of methodology for this problem. A recent model, causalNL, uses a structural causalNL model for instance-dependent label-noise learning and obtained excellent experimental results. The implementation of the algorithm is based on the VAE model, which encodes latent variables Y and Z with the observable variables X and Y. This in turn generates the transfer matrix. But it relies on some unreasonable assumptions. In this paper, we introduce CGAN to the causalNL model, which avoids setting $\mathrm{P}(Y)$ and $\mathrm{P}(Z)$ for a specific distribution. GAN's ability of processing data do not need to set a specific distribution. ICC was validated on several authoritative datasets and compared to a variety of proven algorithms including causalNL. The paper presents notable findings on the ICC model (Introduce CGAN to causalNL) shows excellent training ability on most datasets. Surprisingly, ICC shows totally higher accuracy than causalNL in CIFAR10.

Keywords: Label noise, CGAN, causalNL, Variational Autoencoders (VAE)

1. Introduction

Learning with noisy labels has a historical foundation traced back to [1] and has garnered recent attention [2-5]. In real world, datasets often contain label noise due to the utilization of inexpensive but imperfect data which come from crowd-sourcing and web crawling. The application of these data can result in poor generalization of deep neural networks due to the incorporation of erroneous labels [6,7].

To enhance the generalization capacity of learning models in the presence of noisy labels, a category of existing methods aims to characterize the label noise. These approaches concentrate on revealing the transition relationship between clean and noisy labels for instances, represented as $\mathrm{P}\left(\tilde{Y}\middle|Y,X\right)$, where$\tilde{Y}$, Y, and X denote noisy label, latent clean label, and instance variables respectively. Modeling label noise has a benefit: given solely noisy data, classifiers can theoretically converge towards the optimal ones defined by clean data, provided the transition relationship is identifiable. However, the general identifiability of the transition relationship is not guaranteed. To address this, Yao et.al.[8] assumes that causalNL[8], leveraging causalNL[8] information to improve the identifiability of the transition matrix $\mathrm{P}\left(\tilde{Y}\middle|Y,X\right)$ beyond direct assumptions about the transition relationship. This method encapsulates the causalNL[8] structure using both observable and latent variables: instance X, noisy label $\tilde{Y}$, latent feature Z, and latent clean label Y. CausalNL is a generative model which is designed to capture the intricate relationships between these variables as indicated by the causalNL[8] graph. Additionally, causalNL[8] builds upon the variational autoencoder (VAE) framework.

When an inference network is established. it can deduce latent variables Z and Y while at the same time maximizing the marginal likelihood $\mathrm{p}\left(X,\tilde{Y}\right)$ using the provided noisy data. an illustration of the causalNL[8] graph is provided in Figure 1. In a real-world, X could represent the image containing a digit; The meaning of each symbol in the causal structure graph is shown above.

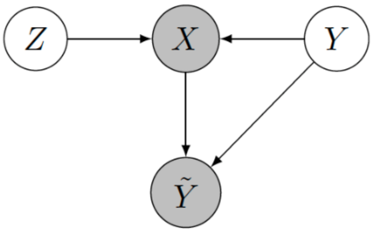

Figure 1. The process of data generation through causalNL[8] model, where the shaded variables are observable and the unshaded variables are latent.

Inspired by causalNL[8] model , we reinforce the decoder of causalNL[8] model with CGAN, using characterization of CGAN for latent clean variable y discriminated and sent to the discriminator and generator. In our model, latent feature Z and latent clean label Y do not need to follow specific distribution; since CGAN could making prior assumptions, directly sample and optimize the image with noise by gaming the results iteratively. The paper outlines the causalNL[8] generative model and the incorporation of variational autoencoders and Conditional Generative Adversarial Nets. Experimental validation highlights the superiority of the proposed approach over baselines in test data accuracy, particularly in cases of substantial label noise rates.

In this work, Section 2 will briefly present related works and causalNL[8] which is relevant to our work. Section 4 will describe our model principle, composition and algorithm implementation. Section 5 will introduce our experiments based on ICV.

2. Related work

Our work is based on many recent studies' ideas and results, summarized below.

Label noise refers to inaccurate annotations caused by human error, ambiguities in data, or inherent difficulty, which leads to detrimental impact on the performance of machine learning models, including reduced generalization ability, and lower performance on unseen data. This has always been a problem, and there are mainly three previous existing ways to solve it.

Robust Loss Functions[9]are pointed out as a way to fix lost data. They apply the forward loss procedure and backward loss procedure with the use of a T matrix to correct the data. Another method Label Smoothing is proposed[10], which involves a mixture of the true label and a smoothed distribution when training. The third way is using a model that contains co-teaching. Models can be trained on different subsets, and then data will be gathered together for analyzation. The paper, Instance-Dependent

Label-Noise Learning under Structural CausalNL[8] Models, provides a way of training with the use of the VAE model based on casual models showing improvements in the testing data while there is a serious mistake made by the contradictory assumptions. Our paper will explain it later and discuss a better model.

To improve the model and have better generative results, we will consider using both the ideas of VAE and GAN to generate a better one.

Variational Autoencoders (VAE) is a type of generative model that combines elements of autoencoders and probabilistic modeling. It is often used for unsupervised learning tasks, such as dimensionality reduction, data compression, and generative modeling. When using a VAE model, we first input data into a neural network encoder with reparameterization trick, and then into the latent space, a lower-dimensional continuous space where each point represents a compressed input data. Finally, the decoder, which is another neural network, maps the data back to the original space and also reconstruct the loss. Another model we consider using is GAN.[11]. Generative Adversarial Network (GAN) with a generator and discriminator is trained in a competitive manner. GAN model has been evolving since it first appears and now there are several specialized architectures for different specific tasks. The first type is Conditional GAN, also known as CGAN for short. Different from the original GAN, CGAN has a generator which gives out a label along the data and a discriminator which attempts to classify if it corresponds to the label. The second one is Deep Convolutional GAN (DCGAN), which can better capture spatial features and improve the imagine quality. Then is circle GAN allowing the conversion of images from one domain to another while it preserve key characteristics. Next, Wasserstein GAN (WGAN) solves some of the formation stability problems in standard GANs by introducing the Wasserstein distance, further improves the results. Finally, there is StyleGAN which focuses on generating images with controllable styles and attributes.

3. Introducing CGAN into causalNL[8]

In this section, we introduce CGAN into the decoder of causalNL[8].We want to utilize the discriminator in CGAN that can provide additional conditional information. Benefited to the CGAN's capability to control the data generation process of data, our method does not require to setting $\mathrm{P}(Z)$ to be a standard normal distribution and $\mathrm{P}(Y)$ to be a uniform distribution.

3.1. VAE-CGAN under the Structural CausalNL Model

We have retained most of the causalNL[8] structure. The causalNL[8] consists of two decoder networks that jointly model the distribution $p_\theta\left(X,\tilde{Y}\middle|Y,Z\right)$ and two encoder (inference) networks which jointly model the posterior distribution $q_\emptyset(Z,Y|X)$

Our method follows causalNL[8]'s causal factorization. Although our method utilizes CGAN to reshape the decoder, the method of modeling the joint distribution does not change. The distribution $p_\theta\left(X,\tilde{Y}\middle|Y,Z\right)$ is decomposed as follows:

$$p_\theta\left(X,\tilde{Y}\middle|Y,Z\right) = p_{\theta 1}(X,|Y,Z)p_{\theta 2}\left(\tilde{Y}\middle|Y,X\right)$$

The posterior distribution can be divided into the following sections to infer latent variable Z and Y from observable variables X and $\tilde{Y}$:

$$q_\emptyset\left(Z,Y\middle|\tilde{Y},X\right) = q_{\emptyset 1}\left(Y\middle|\tilde{Y},X\right)q_{\emptyset 2}(Z|Y,X)$$

There is a unreasonable assumption in causalNL[8]. It approximates $q_{\emptyset 1}\left(Y\middle|\tilde{Y},X\right)$ based on an assumption that the given instance X and the clean label Y are conditionally independent from the noisy label $\tilde{Y}$, i.e , $q_{\emptyset 1}\left(Y\middle|\tilde{Y},X\right) = q_{\emptyset 1}(Y|X)$. This assumption actually violates our structural causalNL[8] model, but we also do not have a good method to avoid this unreasonable assumption. Our method retains this assumption which does not have large approximation error.

$$q_\emptyset(Z,Y|X) = q_{\emptyset 2}(Z|Y,X)\ q_{\emptyset 1}(Y|X)$$

Then, we introduce CGAN based on the mathematical principle and assumptions of causalNL[8]. Our model replaced the first part of the decoder which use latent variables Z and Y to generate X. We design a Conditional GAN which input Y and Z as conditions and input noisy X into generator. Then the X generated and real X are discriminated in discriminator which judge the G(X) is real or fake. Because of the idea of generative adversarial, there is no need to set a special distribution of p(Y) and p(Z).

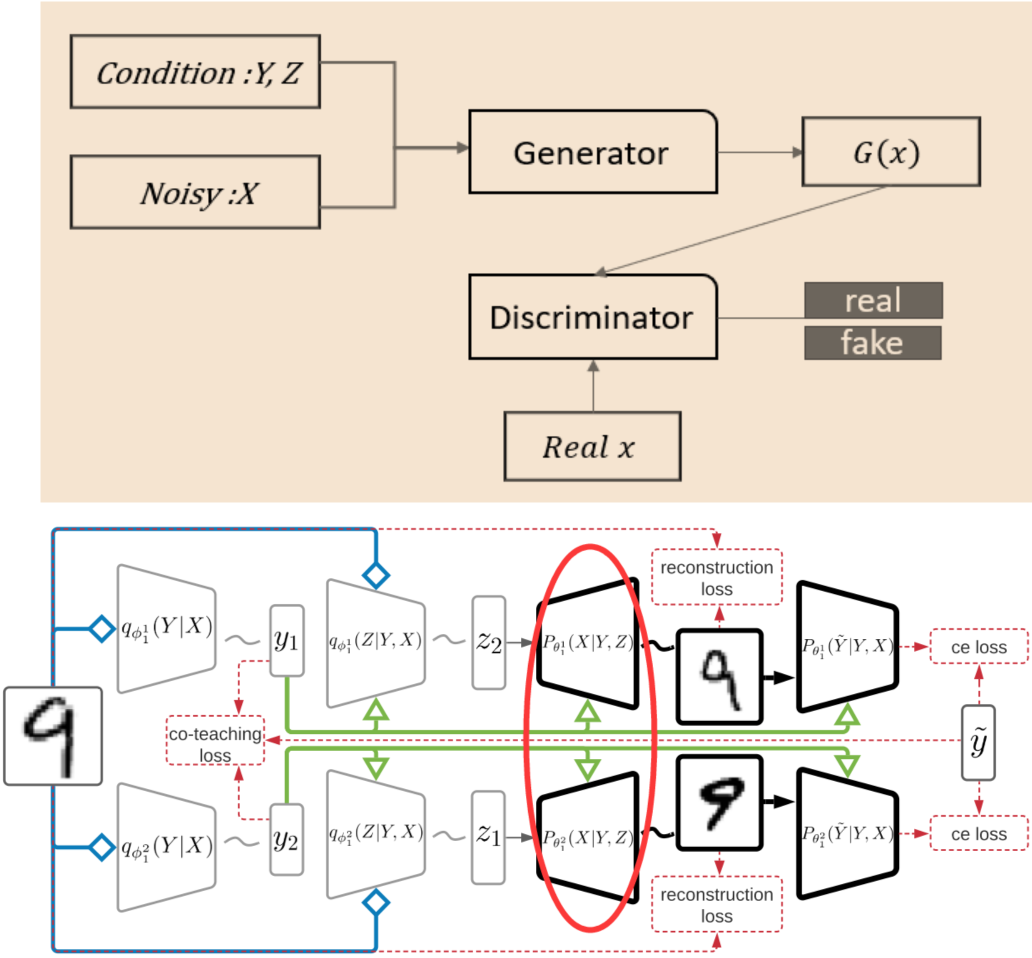

Figure 2. our model: VAE-CGAN under structural causalNL[8] model.

3.2. *Optimization of Parameters*

We follow the TongLiang's work for reference et.al.[8]. The causalNL model expect to minimize $-\mathrm{ELBO}(\mathrm{x},\tilde{y})$ of the marginal probability of each datapoint. It is derived as:

$$-\mathrm{ELBO}(\mathrm{x},\tilde{y}) = -E_{(z,y)\sim q_{\emptyset}(Z,Y|x)}[logp_{\theta 1}(x,|y,z)] - E_{y\sim q_{\emptyset 1}(Y|x)}[logp_{\theta 2}(\tilde{y}|y,x)] + kl(q_{\emptyset 1}(Y|x) \parallel p(Y) + E_{y\sim q_{\emptyset 1}(Y|x)}[kl(q_{\emptyset}(Z|y,x) \parallel p(Z)]$$

kl(·) describe the difference between the distribution. It corresponds to the decoder loss and we retain it. Because we utilize CGAN replace the first part of the decoder which use latent variables Z and Y to generate X, we do not need the reconstruction loss. In our model we use the loss function of CGAN:

$$\min_G max_D V(D,G) = E_{x\sim P_{data}(x)}[log\mathrm{D}(x|y,z)] + E_{\mathrm{X}\sim p_{\mathrm{X}}(X)}[\log(1 - D(G(X|y,z)))]$$

For our model the $-\mathrm{ELBO}(x,\tilde{y})$ will be reconstructed as follows:

$-\ \text{ELBO}\ (x,\tilde{y}) = -E_{y\sim q_{\phi_1}(Y|x)}\left[logp_{\theta_2}(\tilde{y}|y,x)\right] + E_{x\sim P_{data}(x)}[logq_{\emptyset 1}(Y|x)] + E_{y\sim q_{\emptyset 1}(Y|x)}[\log q_{\emptyset 2}(Z|y,x)]$

So far, our model successfully introduces CGAN into the causalNL[8]. Compared with causalNL[8], our model not only greatly reduced bias of the reliable example, but also avoid the unreasonable assumption of the distribution.

3.3. Practical Implementation

```
Algorithm 1 CausalNL[8] – CGAN
  Input: A noisy sample S, Average noise rate $\boldsymbol{\rho}$, Total epoch $\boldsymbol{T_{max}}$ Batch size N.
  1: For T = 1, . . . , $\boldsymbol{T_{max}}$:
  2:   For mini-batch $\overline{\boldsymbol{S}} = \{\boldsymbol{x_i}\}_{i=0}^{N}$, $\tilde{\boldsymbol{L}} = \{\tilde{\boldsymbol{y}}_i\}_{i=0}^{N}$ in S:
  3:     Feed $\overline{\boldsymbol{S}}$ to encoders $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_1^1}$ and $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_1^2}$ to get clean label sets $\boldsymbol{L_1}$ and $\boldsymbol{L_2}$, respectively;
  4:     Feed $(\overline{\boldsymbol{S}},\ \boldsymbol{L_1})$ to encoder $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_2^1}$ to get a representation set $\boldsymbol{H_1}$, feed $(\overline{\boldsymbol{S}}, \boldsymbol{L_2})$ to $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_2^2}$ to. get $\boldsymbol{H_2}$;
  5:     Update $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_2^1}$ and $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_2^2}$ with co-teaching loss;
  6:     Feed $(\boldsymbol{L_1},\ \boldsymbol{H_1})$ to the generator of the CGAN $\hat{\boldsymbol{c}}_{\boldsymbol{\theta}_1^1}$ to get dataset $\overline{\boldsymbol{S}}_{\boldsymbol{1}}$, feed $(\boldsymbol{L_2},\ \boldsymbol{H_2})$ to $\hat{\boldsymbol{c}}_{\boldsymbol{\theta}_1^2}$ to get $\overline{\boldsymbol{S}}_{\boldsymbol{2}}$;
  7:     Feed $(\overline{\boldsymbol{S}}_{\boldsymbol{1}},\ \boldsymbol{L_1})$ to decoder $\widehat{\boldsymbol{p}}_{\boldsymbol{\theta}_2^1}$ to get predicted noisy labels $\tilde{\boldsymbol{L}}_{\boldsymbol{1}}$, feed $(\overline{\boldsymbol{S}}_{\boldsymbol{2}},\ \boldsymbol{L_2})$ to $\widehat{\boldsymbol{p}}_{\boldsymbol{\theta}_2^2}$ to get $\tilde{\boldsymbol{L}}_{\boldsymbol{2}}$;
  8:     Update networks $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_1^1}$, $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_2^1}$, $\hat{\boldsymbol{c}}_{\boldsymbol{\theta}_1^1}$ and $\widehat{\boldsymbol{p}}_{\boldsymbol{\theta}_2^1}$ by calculating loss on $\left(\overline{\boldsymbol{S}},\ \overline{\boldsymbol{S}}_{\boldsymbol{1}},\ \tilde{\boldsymbol{L}},\ \tilde{\boldsymbol{L}}_{\boldsymbol{1}}\right)$, update networks $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_1^2}$, $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_2^2}$, $\hat{\boldsymbol{c}}_{\boldsymbol{\theta}_1^2}$ $\widehat{\boldsymbol{p}}_{\boldsymbol{\theta}_2^2}$ by calculating loss on $\left(\overline{\boldsymbol{S}},\ \overline{\boldsymbol{S}}_{\boldsymbol{2}},\ \tilde{\boldsymbol{L}},\ \tilde{\boldsymbol{L}}_{\boldsymbol{2}}\right)$;
  Output: The inference network $\widehat{\boldsymbol{q}}_{\boldsymbol{\phi}_1^1}$.
```

In this section, we present the structure and loss functions of our model. Our approach involves incorporating duplicate decoders, encoders, and CGANs. Due to our implementation of the co-teaching technique during model training, the branches are constructed identically to each other.

We need two encoder networks, one decoder network and one CGAN in our model:

$$Y_1 = \hat{q}_{\phi_1^1}(X), Z_1 = \hat{q}_{\phi_2^1}(X, Y_1)$$

$$X_1 = \hat{c}_{\theta_1^1}(Y_1, Z_1), \tilde{Y}_1 = \hat{p}_{\theta_2^1}(X_1, Y_1)$$

The initial encoder, $\hat{q}_{\phi_1^1}(X)$, inputs an instance X, and outputs a forecasted clean label Y_1. The second encoder, $\hat{q}_{\phi_2^1}(X, Y_1)$, inputs both the instance X and the generated label Y_1 and outputs a latent feature Z_1. The first decoder trains a CGAN $\hat{c}_{\theta_1^1}(Y_1, Z_1)$ on inputs Y_1 and Z_1 to generate image X_1 using the generator. Furthermore, the X_1 and Y_1 are then used as input for the second decoder, $\hat{p}_{\theta_2^1}(X_1, Y_1)$, which returns predicted noisy labels $\tilde{Y}_1$. We employed the reparameterization technique to facilitate sampling, enabling $\hat{q}_{\phi_2^1}(X, Y_1)$ and $\hat{p}_{\theta_2^1}(X_1, Y_1)$ to undergo backpropagation. The decoder and encoder networks in the second branch have architecture similar to the first branch.

$$Y_2 = \hat{q}_{\phi_1^2}(X), Z_2 = \hat{q}_{\phi_2^2}(X, Y_2)$$

$$X_2 = \hat{c}_{\theta_1^2}(Y_2, Z_2), \tilde{Y}_2 = \hat{p}_{\theta_2^2}(X_2, Y_2)$$

When training the model, the two encoders $\hat{q}_{\phi_1^1}(X)$ and $\hat{q}_{\phi_1^2}(X)$ from different branches train each other given the mini-batch.

Loss functions: There will be three parts for the loss functions. The first part is a loss function of CGAN, the second part is ELBO, and the third part is a co-teaching loss.

For the CGAN loss function, G and D are concurrently trained. We modify G's parameters to minimize $D(G(X|y,z))$, which is equivalent to maximizing $\log(1-D(G(X|y,z)))$. Similarly, we adjust parameters for D to maximize $D(x|y,z)$, as if both players are engaging in a two-player min-max game with a value function $V(G,D)$.

$$\underset{G}{min}\,\underset{D}{max}\,V(D,G)=E_{x\sim p_{data}(x)}[\log D(\mathrm{x}|y,z)]+\ E_{\mathrm{X}\sim p_{\mathrm{X}}(X)}\left[\log\left(1-D\big(G(X|y,z)\big)\right)\right]$$

The first term is $E_{x\sim p_{data}(x)}[\log D(x|y,z)]$, which is the expected value of the probability that the discriminator is given real training data from x conditioned on predicted clean label y and latent representation z. The second term is $E_{\mathrm{X}\sim p_{\mathrm{X}}(X)}[\log\big(1-D(G(\mathrm{X}|\mathrm{y},\mathrm{z}))\big)$, which is the expected value of the probability that the discriminator is given fake generated data or G(X) conditioned on y and z.

For the negative ELBO, we minimize the $-\mathrm{ELBO}(x,\tilde{y})$ of the marginal probability of each datapoint $(x,\tilde{y})$. The first term is $-E_{y\sim q_{\phi_1}(Y|x)}[logp_{\theta_2}(\tilde{y}|y,x)]$, which seek to learn $\tilde{y}$ given y and x, it is the same as the logistic loss on both decoders $\hat{p}_{\theta_2^1}(X_1,Y_1)$, $\hat{p}_{\theta_2^2}(X_2,Y_2)$ .The second term $E_{x\sim P_{data}(x)}[logq_{\phi_1}(Y|x)]$ is set to learn predicted clean label given inference x, this can be substituted by using logistic loss of both encoders $\hat{q}_{\phi_1^1}(X)$ and $\hat{q}_{\phi_1^2}(X)$. The last term $E_{y\sim q_{\phi_1}(Y|x)}[logq_{\phi_2}(Z|y,x)]$ can be substituted by using logistic loss on output of both encoders $\hat{q}_{\phi_2^2}(X,Y_2)$, $\hat{q}_{\phi_2^1}(X,Y_1)$.

For the co-teaching loss, the essential idea is that we feed data to the two encoders $\hat{q}_{\phi_1^1}(X)$ and $\hat{q}_{\phi_1^2}(X)$ and the networks interact with to identify potential clean data in this mini-batch and use such data for training. Eventually, both encoders update themselves by cross-entropy loss through backpropagation over the data selected by the opposing network.

4. Experiments

In the experimental section, we tested the accuracy of the classification of our improved model and compared it to causalNL[8] and other mainstream label noise learning algorithms [12-17] on synthetic datasets.

4.1. Experimental Setup

Datasets To confirm the accuracy of our new model, we tested it on three artificially altered versions of the datasets, i.e., SVHN [18], CIFAR10 and CIFAR100[19]. SVHN involves 73,257 training images and 26,032 test images in a total of 10 classes. Both of CIFAR10 and CIFAR100 include 50,000 training images and 10,000 test images. CIFAR10 has 10 classes, with 6000 images per class and CIFAR100 has 100 classes, with 600 images per class. All of these data sets include clean data. By manually tuning, we put the instance-dependent label noise into the training datasets. To increase the authenticity and confidence of the results, we repeated the experiment five times for each synthetic noisy dataset.

Network structure and optimization to be fair, all of the attainment of our experiments is through PyTorch. For all synthetic data sets, the latent representation Z's dimension is set to 25.

Baselines and measurements We compare the modified approach (ICV) to the original model (VAE) and other mainstream approaches: (i) Decoupling, where two networks are trained on samples whose predictions differ. (ii) CE, trains standard deep networks with cross entropy loss on noisy data sets. (iii) MentorNet, Co-teaching, handles noise labels primarily by training instances with slight loss values. (iv) Reweight, Forward and T-Revision. These methods employ a matrix T, which is class-dependent transition to remedy the loss function. After 150 epochs were tested in clean test sets, we obtain relatively reliable results by calculating the means and standard deviations of the last 10 epochs. In this paper, we cite the test results in for comparison.

4.2. Classification accuracy Evaluation

Results of synthetic noise datasets: Table 1, 2, 3 reveal the classification accuracy on the data sets of CIFAR10, CIFAR100 and SVHN, at all kinds of instance-dependent label-noise (IDN), respectively. Figures 3 and 4 visually show the comparison of the proposed model (ICV) with casualNL [8] in different datasets (CIFAR10 and SVHN), at a noise level of 45%. Comprehensive experiments show that ICV is strong under different noise conditions, it has excellent stability, robustness, and has a faster convergence rate. It shows that introducing CGAN into the decoder of causalNL[8] will improve the performance of the whole model's accuracy after relinquishing the setting of the standard normal distribution or a uniform distribution.

For CIFAR10 and CIFAR100 according to the Table 1 and Table 2, ICV appear to outperform all baselines obviously, including casualNL [8], at all different noise rate. It means that CGAN does play a role in improving accuracy. The addition of CGAN does not reduce accuracy but actually plays a role in improving accuracy. Figure 1 shows that in the CIFAR10 datasets, ICV has roughly the same convergence rate as casualNL [8], but the proposed model (ICV) has higher accuracy and stability after convergence obviously.

For SVHN, Table 3 demonstrates that the performance of ICV in terms of accuracy is not so strong, which may be because CGAN is not suitable for the data, which are in the discrete forms or CGAN itself has problems with training instability, gradient disappearance, etc. But Figure 2 shows that ICV has a certain improvement in convergence speed compared with casualNL [8] .

Table 1: Means and standard deviations percentage of classification accuracy on CIFAR10 with different label noise levels.

	IDN20%	**IDN30%**	**IDN40%**	**IDN45%**	**IDN50%**
Decoupling	36.53±σ=0.49	30.93±σ=0.88	27.85±σ=0.91	23.81±σ=1.31	19.59±σ=2.12
Co-teaching	37.96±σ=0.53	33.43±σ=0.74	28.04±σ=1.43	25.60±σ=0.93	23.97±σ=1.91
CE	30.42±σ=0.44	24.15±σ=0.78	21.45±σ=0.70	15.23±σ=1.32	14.42±σ=2.21
MentorNet	38.91±σ=0.54	34.23±σ=0.73	31.89±σ=1.19	27.53±σ=1.23	24.15±σ=2.31
Reweight	36.73±σ=0.72	31.91±σ=0.91	28.39±σ=1.46	24.12±σ=1.41	20.23±σ=1.23
Forward	36.38±σ=0.92	33.17±σ=0.73	26.75±σ=0.93	21.93±σ=1.29	19.27±σ=2.11
Mixup	32.92±σ=0.76	29.76±σ=0.87	25.92±σ=1.26	23.13±σ=2.15	21.31±σ=1.32
T-Revision	37.24±σ=0.85	36.54±σ=0.79	27.23±σ=1.13	25.53±σ=1.94	20.23±σ=1.23
CausalNL[8]	41.47±σ=0.32	40.98±σ=0.62	34.02±σ=0.95	33.34±σ=1.13	32.13±σ=2.23
ICC	**42.01±σ=0.45**	**41.35±σ=0.64**	**36.17±σ=1.01**	**33.77±σ=1.09**	**33.12±σ=1.67**

Table 2: Means and standard deviations percentage of classification accuracy on CIFAR100 with different label noise levels.

	IDN20%	**IDN30%**	**IDN40%**	**IDN45%**	**IDN50%**
Decoupling	90.02±σ=0.25	91.59±σ=0.25	88.27±σ=0.42	84.57±σ=0.89	65.14±σ=2.79
Co-teaching	93.93±σ=0.31	92.06±σ=0.31	91.93±σ=0.81	89.33±σ=0.71	67.62±σ=1.99
CE	91.51±σ=0.45	91.21±σ=0.43	87.87±σ=1.12	67.15±σ=1.65	51.01±σ=3.62
MentorNet	**94.08±σ=0.12**	92.73±σ=0.37	90.41±σ=0.49	87.45±σ=0.75	61.23±σ=2.82
Reweight	92.44±σ=0.34	92.32±σ=0.51	91.31±σ=0.67	85.93±σ=0.84	64.13±σ=3.75
Forward	91.89±σ=0.31	91.59±σ=0.23	89.33±σ=0.53	80.15±σ=1.91	62.53±σ=3.35
Mixup	89.73±σ=0.37	90.02±σ=0.35	85.47±σ=0.55	82.41±σ=0.62	68.95±σ=2.58
T-Revision	93.14±σ=0.53	93.51±σ=0.74	92.65±σ=0.76	88.54±σ=1.58	64.51±σ=3.42
CausalNL[8]	94.06±σ=0.23	**93.86±σ=0.37**	**93.82±σ=0.45**	**93.19±σ=0.81**	**85.41±σ=2.95**
ICC	93.58±σ=0.29	93.58±σ=0.29	92.07±σ=0.42	92.79±σ=0.45	77.09±σ=1.65

Table 3: Means and standard deviations percentage of classification accuracy on SVHN with different label noise levels.

	IDN20%	**IDN30%**	**IDN40%**	**IDN45%**	**IDN50%**
Decoupling	90.02±σ=0.25	91.59±σ=0.25	88.27±σ=0.42	84.57±σ=0.89	65.14±σ=2.79
Co-teaching	93.93±σ=0.31	92.06±σ=0.31	91.93±σ=0.81	89.33±σ=0.71	67.62±σ=1.99
CE	91.51±σ=0.45	91.21±σ=0.43	87.87±σ=1.12	67.15±σ=1.65	51.01±σ=3.62
MentorNet	**94.08±σ=0.12**	92.73±σ=0.37	90.41±σ=0.49	87.45±σ=0.75	61.23±σ=2.82
Reweight	92.44±σ=0.34	92.32±σ=0.51	91.31±σ=0.67	85.93±σ=0.84	64.13±σ=3.75
Forward	91.89±σ=0.31	91.59±σ=0.23	89.33±σ=0.53	80.15±σ=1.91	62.53±σ=3.35
Mixup	89.73±σ=0.37	90.02±σ=0.35	85.47±σ=0.55	82.41±σ=0.62	68.95±σ=2.58
T-Revision	93.14±σ=0.53	93.51±σ=0.74	92.65±σ=0.76	88.54±σ=1.58	64.51±σ=3.42
CausalNL[8]	94.06±σ=0.23	**93.86±σ=0.37**	**93.82±σ=0.45**	**93.19±σ=0.81**	**85.41±σ=2.95**
ICC	93.58±σ=0.29	93.58±σ=0.29	92.07±σ=0.42	92.79±σ=0.45	77.09±σ=1.65

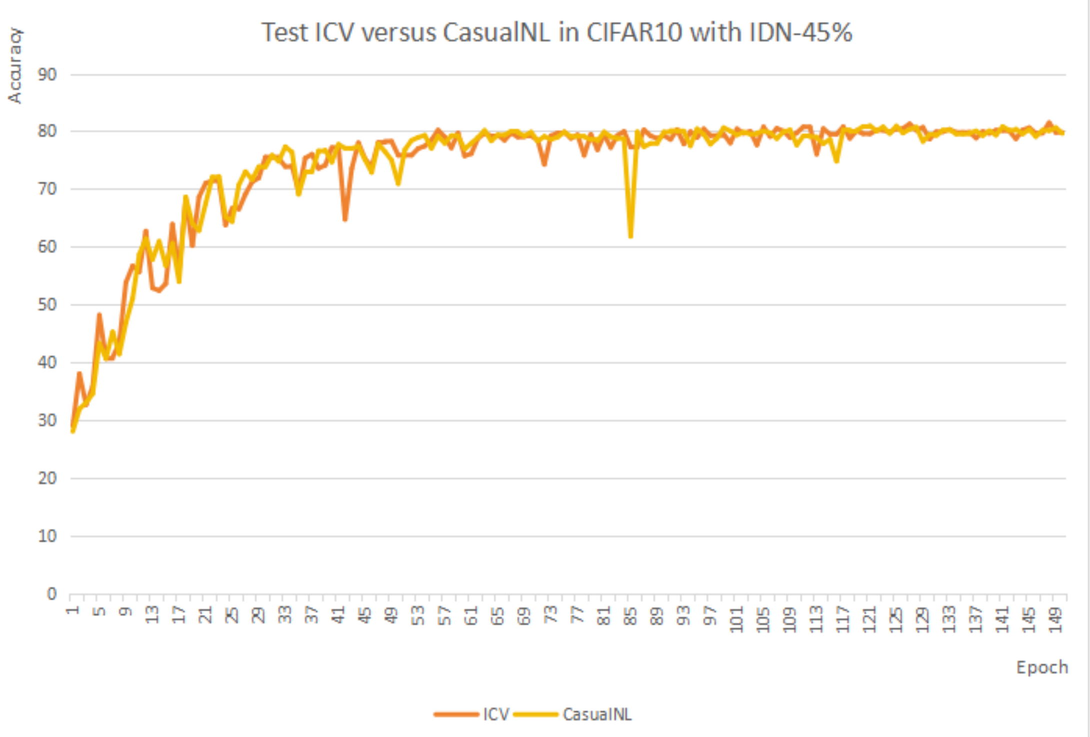

Figure 3. Test ICV versus CasualNL in CIFAR10 with IDN-45%.

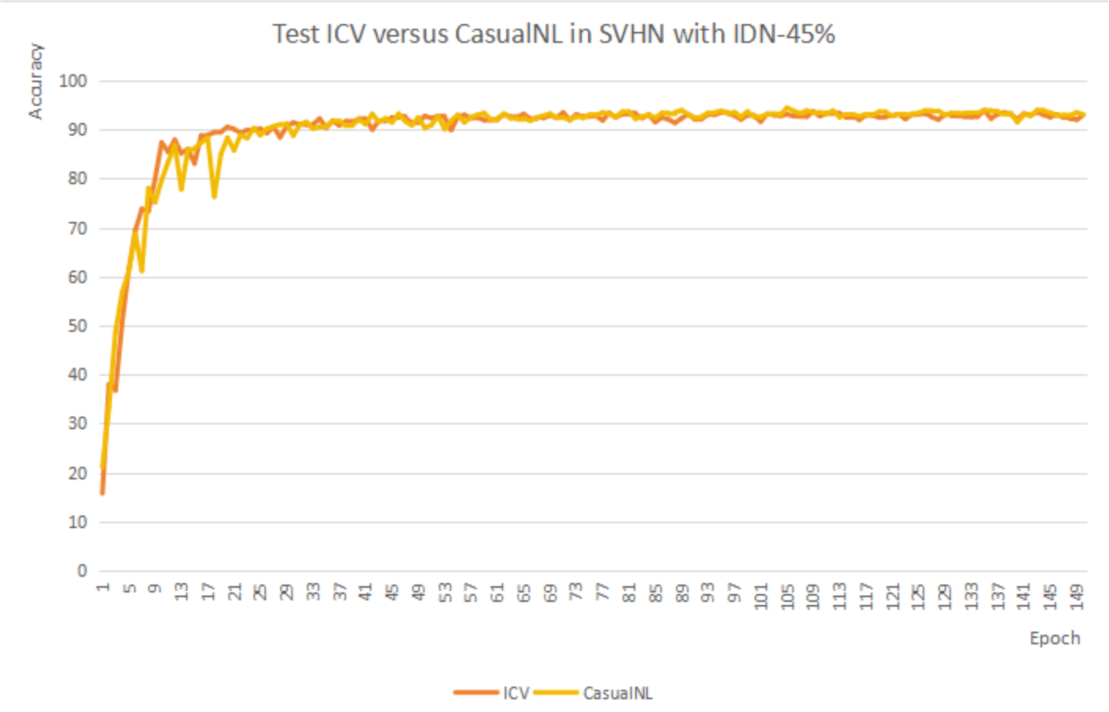

Figure 4. Test ICC versus CasualNL in SVHN with IDN-45%.

5. Conclusion

In this paper, we have investigated how to use CGAN during the process of generating image x in a VAE model. The previous assumptions are made on the predicted clean label y and latent representation z, the assumptions are a bit odd by limiting both y and z followed specific distributions. Inspired by a generative model CGAN, we propose a new approach called ICC. Our model makes use of the causalNL[8] graph to contribute to the identifiability of the transition matrix while free up the limitation on process of generating image x. The classification accuracy of ICC outperforms all the state-of-the art methods. In our future work, we will try to implement different regression models into the process of generating predicted clean label Y using image X, for a superior result of the model performance.

Acknowledgement

Zixing Gou, Yifan Sun, and Zhebin Jin contributed equally to this work and should be considered co-first authors. Hanqiu Hu, and Weiyi Xia contributed equally to this work and should be considered co-second authors.

References

[1] Dana Angluin and Philip Laird. Learning from noisy examples. Machine Learning, 2(4):343–370, 1988

[2] Yang Liu. The importance of understanding instance-level noisy labels. ICML, 2021

[3] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In ICLR, 2021

[4] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In NeurIPS, pages 6835–6846, 2019

[5] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with Instance-dependent label noise. In CVPR, 2021

[6] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In International Conference on Machine Learning, pages 233–242. PMLR, 2017

[7] Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James T Kwok. Searching to exploit memorization effect in learning with noisy labels. In ICML, 2020

[8] Yu Yao, Tongliang Liu . Instance-dependent Label-noise Learning under a Structural CausalNL[1] Model

[9] Giorgio Patrini, Alessandro Rozza, Aditya Menon, Richard Nock, Lizhen Qu .Making Deep Neural Networks Robust to Label Noise: a Loss Correction Approach. Oral paper at CVPR 2017

[10] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, Geoffre Hinton .Regularizing Neural Networks by Penalizing Confident Output Distributions. Submitted to ICLR 2017

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville,.Yoshua Bengio.Generative Adversarial Networks

[12] Yang Liu. The importance of understanding instance-level noisy labels. ICML, 2021

[13] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In CVPR, pages 1944–1952, 2017

[14] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In ICML, pages 2309–2318, 2018.

[15] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu,Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training ofdeep neural networks with extremely noisy labels. In NeurIPS, pages 8527–8537, 2018.

[16] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In ICLR, 2018

[17] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with Instance-dependent label noise. In CVPR, 2021

[18] Yuval Netzer, TaoWang, Adam Coates, Alessandro Bissacco, BoWu, and AndrewY.Ng. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.